Random Coverage with Guaranteed Connectivity: Joint Scheduling for Wireless Sensor Networks

Chong Liu, *Student Member, IEEE,* Kui Wu, *Member, IEEE,* Yang Xiao, *Senior Member, IEEE,* and Bo Sun, *Member, IEEE*

***Abstract*— Sensor scheduling plays a critical role for energy efficiency of wireless sensor networks. Traditional methods for sensor scheduling use either sensing coverage or network connectivity, but rarely both. In this paper, we deal with a challenging task: without accurate location information, how to schedule sensor nodes to save energy and meet both constraints of sensing coverage and network connectivity? Our approach utilizes an integrated method that provides statistical sensing coverage and guaranteed network connectivity. We use random scheduling for sensing coverage and then turn on extra sensor nodes, if necessary, for network connectivity. Our method is totally distributed, is able to dynamically adjust sensing coverage with guaranteed network connectivity, and is resilient to time asynchrony. We present analytical results to disclose the relationship among node density, scheduling parameters, coverage quality, detection probability, and detection delay. Analytical and simulation results demonstrate the effectiveness of our joint scheduling method.**

***Index Terms*— Wireless Sensor Networks, Scheduling Algorithms, Performance Analysis**

I. Introduction

Wireless Sensor Networks (WSNs) consist of a large number of sensor nodes, each of which integrates sensors, processors, memory, and wireless transceivers within the size of several cube millimeters [20]. Through short-range wireless communication, sensor nodes usually form a multi-hop wireless network to coordinate their behavior, collect and process the measurement data in a distributed fashion [2]. Due to their extremely small dimension, sensor nodes have very limited energy supply. In addition, it is usually hard to recharge the battery after deployment, either because the number of sensor nodes is too large, or because the deployment area is hostile. Once deployed, a sensor network is expected to keep working for several weeks or months. Therefore, energy efficiency becomes the essential requirement in WSNs.

Sensor scheduling plays a critical role for energy efficiency in WSNs. In this paper, we deal with a challenging task: without accurate location information, how to schedule sensor nodes to save energy and meet both constraints of sensing coverage and network connectivity? This problem should be solved in a lot of applications with WSNs, such as the detection of chemical attacks or the detection of forest fire. We use the following example to stress the importance of solving this problem.

Example: Imagine that a wireless sensor network is deployed to detect forest fire. The network should be able to detect the outbreaks of wild fire at any location within the monitored region with a high probability and report the outbreaks to the data collection center (also known as the sink node) with a small delay. In this example, sensing coverage, network connectivity, and energy efficiency are equally important: a large sensing coverage is to meet the users' requirement that an event can be detected with a high probability; network connectivity is to meet the users' requirement that the detected event can be delivered to the sink node; energy efficiency is to meet the users' requirement that the network should keep its operation as long as possible after the deployment. A good scheduling scheme should achieve energy efficiency under the constraints of sensing coverage and network connectivity.

The difficulty in the above joint scheduling problem is that a sensor's sensing range is totally independent of its radio transmission range. It is generally hard to combine and solve the two problems together. This is the reason that although scheduling based on sensing coverage [1], [9], [11], [12], [16], [21], [24], [25] and scheduling based on network connectivity [4], [6], [22] have been studied extensively, very few work has been devoted to solving the joint problem. Since it is usually costly to obtain and maintain the location information of each sensor node, the joint scheduling problem is even more difficult if the location of each sensor node is unknown.

In this paper, we are motivated to provide a solution to the joint scheduling problem under the constraints of both sensing coverage and network connectivity without the availability of per-node location information. Specifically, we aim at designing a scheduling scheme that has the following features at any given time:

1) The sensing coverage is above a given requirement.
2) All the active sensor nodes are connected.
3) Each active sensor node knows at least one shortest or nearly shortest route to the sink node.

The contributions of this paper are as follows. First, we propose a randomized scheduling algorithm and present the analytical results to illustrate the relationship among achievable coverage quality, event detection probability, event detection

This work was partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Foundation for Innovation (CFI).

Chong Liu is with Department of Computer Science, University of Victoria, Victoria, British Columbia, Canada V8W 3P6, E-mail: chongliu@cs.uvic.ca.

Kui Wu is with Department of Computer Science, University of Victoria, Victoria, British Columbia, Canada V8W 3P6, E-mail: wkui@cs.uvic.ca. Kui Wu is the corresponding author.

Yang Xiao is with Department of Computer Science, The University of Memphis, TN 38152, USA, E-mail: yangxiao@ieee.org.

Bo Sun is with Department of Computer Science, Lamar University, TX 77710, USA, E-mail: bsun@cs.lamar.edu.

Manuscript received June 29, 2005; revised December 8, 2005.

delay, energy saving, and node density. We also demonstrate that the randomized algorithm is resilient to time asynchrony if the network is sufficiently dense. Such a feature is indispensable for a practical scheduling algorithm since precise time synchronization is very hard for large sensor networks [5] and requires extra communication and energy consumptions. Second, we propose a rule to turn on extra sensors, with which the network connectivity is guaranteed. The rule permits each sensor node to decide whether and when it should turn on in addition to the time slot scheduled by the randomized scheduling algorithm. Third, with our joint scheduling method, each active node knows at least one path to the sink. In this sense, the routing problem is simultaneously solved with the maintenance of network connectivity and no extra routing protocols are needed. Finally, we carry out performance evaluation and demonstrate that the joint scheduling method can achieve user-specified coverage quality with guaranteed network connectivity.

The rest of paper is organized as follows. In the next section, we introduce the network model. In Section III, we introduce a randomized scheduling scheme that provides statistical sensing coverage. In Section IV, we propose a joint scheduling scheme that is based on the randomized scheduling algorithm and turns on extra sensors, if necessary, for maintaining network connectivity. In Section V, we analyze the relationship among coverage quality, detection delay, detection probability, energy saving, and node density. Section VI presents the performance evaluation of our joint scheduling scheme. We introduce related work in Section VII and finally conclude the paper in Section VIII.

II. Network Model

There are many ways to organize the communication architecture of a sensor network. A sensor network could be in a hierarchical structure where each sensor communicates with a local cluster head and the cluster head communicates directly with the sink node. Alternatively, it could be in a flat communication structure as well, where each sensor has essentially the same role and relies on other sensors to relay its messages to the sink node via multi-hop radio communication. In this paper, we assume the flat communication architecture, where the joint problem of sensing coverage and network connectivity arises.

We consider *stationary* sensor networks in a two-dimensional field and assume that sensor nodes are randomly and independently deployed in a field. Compared to other sensor deployment strategies such as deployment in grids or in pre-define positions, random deployment is much easier and cheaper [18]. Also, scheduling for a regular network topology such as grids is simple and may not deserve further investigation.

We assume that a sensor node's radio transmission range is fixed and totally independent of its sensing range because of different hardware components involved. Unlike other work [14], [26] that puts certain constraints on the radio range and the sensing range, our work makes no assumption on the relationship between them.

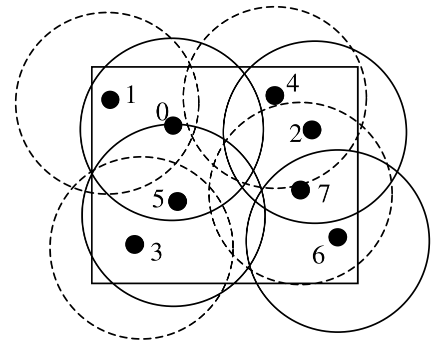


Fig. 1. An Example of the Randomized Coverage-Based Algorithm

We do not assume accurate global time synchronization, which is an extremely hard task for large-scale sensor networks. Instead, our scheduling algorithm permits slight time asynchrony without performance degradation.

III. Randomized Scheduling for Coverage

We have designed a randomized scheduling algorithm for sensing coverage which has several prominent features [10], [11]. The algorithm does not assume the availability of any location or directional information. It is a purely distributed algorithm, thus scalable for large networks. It is also resilient to clock asynchrony and requires only a roughly synchronized clock, which significantly decreases the energy and communication overhead introduced by maintaining network-wide time synchronization. In the following we briefly summarize the basic idea of the randomized scheduling algorithm.

Assume that the sensor nodes constitute a set S. Given a number k, each sensor node randomly joins one of the k disjoint subsets of set S. Once the k subsets are determined, they work alternatively. At any given time, there is only one subset working, and all the sensor nodes belonging to this subset will turn on. The intuition is that when the network is sufficiently dense, each subset alone will cover most part of the field. This randomized algorithm was stimulated by the work [15] and has been proposed independently at the same time by [1] and [10].

Figure 1 shows an example. Assume that we have eight sensors (with IDs $0, 1, ..., 7$) randomly deployed in a rectangular area. Assume that the eight sensors will be assigned into two disjoint subsets, S_0 and S_1. Each sensor randomly selects a number (i.e., 0 or 1) and then joins the corresponding subset. Assume that sensors $0, 2, 5, 6$ select number 0 and thus join subset S_0, and sensors $1, 3, 4, 7$ select number 1 and thus join subset S_1. Then subsets S_0 and S_1 work alternatively, that is, when sensors $0, 2, 5, 6$, whose sensing ranges are denoted as the solid circles, are active, sensors $1, 3, 4, 7$, whose sensing ranges are denoted as the dashed circles, fall asleep, and vice versa.

IV. Joint Scheduling: Random Coverage with Guaranteed Connectivity

A. Motivation

With the proposed randomized coverage-based scheduling scheme, the coverage quality can be guaranteed statistically by setting an appropriate subset number k. Yet, there is no guarantee on the network connectivity after scheduling. To operate successfully, a sensor network must be connected so that sensor nodes can report the detected events to the sink node. Therefore, in addition to sensing coverage, the sensor network must remain connected, i.e., the active nodes should not be partitioned in any schedule of node duty cycles. We are hence motivated to enhance the above randomized scheduling algorithm such that both coverage quality and network connectivity can be met at any given time.

B. Extra-On Rule

Given that the total number of subsets is k, after the randomized coverage-based scheduling scheme, there are k sub-networks formed, each of which corresponds to a specific subset and consists of all the nodes assigned to that subset. However, there is no guarantee on the connectivity of each sub-network. The following extra-on rule ensures that each sub-network is connected, given that the original network before scheduling is connected. Besides, it also guarantees that the path from any sensor node to the sink node has the global minimum hop count.

Assume that each sensor node knows its minimal hop count to the sink node. A sensor node A is called the upstream node of another sensor node B, if node A and node B are neighboring nodes and the minimal hop count of node A to the sink node is one less than that of node B. Node B is also called node A's downstream node.

Extra-on rule: If a sensor node A has a downstream node B, which is active in time slot i, and if none of node B's upstream nodes is active in that time slot, then node A should also work in time slot i. In other words, besides working in the duty cycles assigned by the randomized coverage-based scheduling, node A is required to work in extra time slots, e.g., time slot i in this case.

This rule requires each sensor node to maintain its minimum hop count to the sink node and the list of its upstream nodes. We stress that the minimum hop count is used to label the relative location information among sensor nodes. Since we focus on static sensor networks only and the failure of certain nodes does not influence such relative relationship, our joint scheduling based on the extra-on rule works correctly in face of network failure without requiring periodical update of the minimum hop count values. The method of collecting the hop count information and its energy cost will be addressed in detail in Section IV-D.

It is natural to realize that the extra-on rule may cause problems of synchronization and large overheads due to the dependency among nodes. These potential problems, however, have been carefully avoided in our protocol design. The details will be provided in the following sections.

C. The Correctness of the Extra-on Rule

Proposition 1: Given that the original network is connected. Applying the extra-on rule to each sub-network obtained with randomized coverage-based scheduling ensures that each sensor node has a shortest path to the sink node in the sub-network.

Proof: Let d_i denote the shortest hop distance from sensor node i to the sink node ($d_i \geq 1$). Let sensor node set S_l^i denote the sensor nodes that are on one of the shortest paths from sensor node i to the sink node and have a path to the sink with length l ($1 \leq l \leq d_i - 1$) in the original network. Since the original network is connected, none of the S_l^i ($1 \leq l \leq d_i - 1$) is empty. Also let p_l^i denote one sensor node in the set S_l^i.

If $d_i = 1$, sensor node i has a shortest path to the sink with one hop. If $d_i > 1$, with the extra-on rule, in any time slot, there exists at least one node p with hop distance $d_i - 1$ belonging to $S_{d_i-1}^i$ that is active and connected with sensor node i. Recursively, in any time slot, for any sensor node i, there is a path $\{p_{d_i-1}^i, p_{d_i-2}^i, \cdots, p_1^i\}$ connecting sensor node i and the sink node. Since the length of this path is equal to d_i, this is a shortest path. □

Based on Proposition 1, after using the extra-on rule, each sensor has a path to the sink node. We can therefore get the following proposition:

Proposition 2: Given that the original network is connected. Applying the extra-on rule to each sub-network obtained with randomized coverage-based scheduling ensures the connectivity of each sub-network.

D. The Joint Scheduling Method in Detail

The joint scheduling method ensures the coverage quality and network connectivity simultaneously and has the following steps.

Step 1: Select a subset randomly

Initially, each sensor node generates a random number i between 0 to $k-1$ (0 and $k-1$ inclusive) and assigns itself to subset i. This is exactly the same as in the randomized coverage-based scheduling scheme.

Step 2: Propagate minimum hop count

This step starts from the sink node at the time when it broadcasts a HOP advertisement message to its immediate neighboring sensor nodes. Each HOP advertisement message contains the minimum hop count to the sink, the nodeID and its subset decision. In the packet broadcast from the sink, the minimum hop count is set to 0. Initially, the minimum hop count to the sink is set to infinity at each sensor node.

Each node, after receiving a HOP advertisement message, will put the message in its buffer. It will defer the transmission of the HOP message after a backoff time and only re-broadcasts the HOP message that has the minimum hop count. Before the re-broadcast of the HOP message, the hop count value in the HOP message is increased by 1. With this method, HOP message broadcasts with a non-minimal hop count will be suppressed if the HOP message with the actual minimal hop count arrives before the backoff time expires. The number of broadcasts from each sensor node depends on the length of backoff time. Increasing the backoff time will significantly decrease the number of broadcasts. Although a large backoff time value will increase the total time required for the completion of this step, we argue that it is an effective solution because this step is a one-time task for static sensor networks and the energy is the most precious resource for sensor nodes.

If no packets are lost, our method can guarantee that at the end of this step, each sensor node will obtain the minimum hop count to the sink node. In practice, packets may be lost due to collisions or poor channel quality. Nevertheless, packet losses will not impact the successful operation of our joint scheduling scheme, i.e., the network will still be connected even if some nodes may have only a nearly shortest path to the sink node. Our simulation results in Section VI-B demonstrate that the number of nodes without knowing the actual minimum hop count at the end of this step is negligible even in the presence of packet losses.

Step 3: Exchange information with local neighbors

Each sensor node locally broadcasts its minimum hop count, its nodeID, its subset decision, the nodeIDs of its upstream nodes and their subset decisions. The upstream nodes are the nodes from which the current node receives its minimum hop count. Each sensor node records and maintains all the information it receives from its immediate neighbors.

Step 4: Enforce the extra-on rule

Based on the extra-on rule and the information from Step 3, each sensor node decides the extra time slots it has to remain active to ensure network connectivity and updates its working schedule accordingly. Then the updated working schedule is broadcasted locally to neighboring sensor nodes.

It is easy to see that the update of a sensor node's working schedule can impact the working schedule of its upstream nodes and the neighboring nodes with the same minimum hop count to the sink. To minimize the number of broadcasts of working schedule updates, it is desirable that a sensor node updates its working schedule after it receives all of the latest working schedules from its downstream nodes. This is exactly the reverse process of Step 2. Therefore, a backoff-based broadcast scheme similar to that in Step 2 can be applied here.

As an example, assume that the network consists of one sink node and four sensor nodes, A, B, C, and D as shown in Figure 2. D is three hops away, B and C are two hops away, and A is one hop away from the sink node. Assume that at the end of Step 1, A, B, C and D are assigned to time slots 1, 2, 3 and 4, respectively. Assume that D broadcasts its updated working schedule first. B and C are its upstream nodes and in Step 3 they know that node D does not have an upstream node working in time slot 4. After they receive D's working schedule update, there are several possibilities.

1) Case 1: Suppose that B and C can hear each other. If B broadcasts its working schedule prior to C, C can hear B's updated working schedule and knows that D has an upstream node working in time slot 4. So C will not schedule itself to work in time slot 4. In this case, B will work in time slots 2 and 4 and C will work in time slot 3 only. Likewise, if C broadcasts its working schedule prior to B, C will work in time slots 3 and 4 and B will work in time slot 2 only.
2) Case 2: Suppose that B and C cannot hear each other. B and C will both work in time slot 4 to ensure network connectivity, no matter which node broadcasts first. Therefore, B will work in time slots 2 and 4 and C will work in time slots 3 and 4.

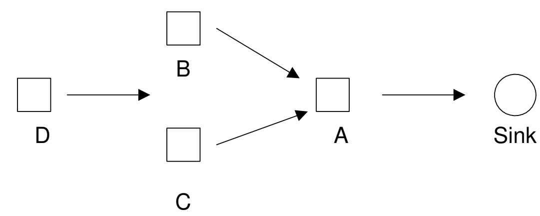


Fig. 2. An Example of the Extra-on Rule

In both cases, based on the latest working schedules received from nodes B and C, node A will know that it has to work in time slots 2, 3 and 4 to ensure network connectivity. Therefore, A will work in time slots 1 to 4.

Step 5: Work according to the new working schedule

E. System Overhead

In this section, we evaluate the system overhead of joint scheduling in terms of the average number of broadcasts from individual sensor node.

Step1: No broadcast is needed in this step.

Step2: The number of broadcasts from each sensor node depends on the length of the backoff time. We study their relationship via simulation. We deploy 1500 sensor nodes randomly in a 200 meters * 200 meters area and place the sink node at the center of the area. The radio range of each sensor node is fixed to 10 meters. We adopt the CSMA MAC layer protocol. To broadcast a HOP advertisement message, we assume that each sensor node has to capture the channel for 1 ms. Figure 3 illustrates the results with the backoff time from 1 ms to 120 ms.

From the figure, we can see that if the backoff time is large enough, each sensor node almost broadcasts only once in this step. The energy saving is at the cost of longer delay to complete this step, as shown in Figure 4. Nevertheless, since this is only a one-time task, the delay should not be a big concern. Actually, our simulation results indicate that if the backoff time is set to 120 ms, the time from the moment when the sink node broadcasts the HOP advertisement message to the moment when the last sensor node finishes the HOP advertisement message is below 2500 ms, which is acceptable. Figure 4 also indicates that using a too small backoff time does not necessarily reduce the total delay since a node may need to broadcast multiple times if HOP messages with a smaller hop value arrive later.

Step3: Each sensor node needs to broadcast only once to notify their neighbors.

Step4: Since the propagation of the extra-on decisions is actually the reverse process of Step 2, we can expect that most of the sensor nodes will broadcast only once if the backoff time is appropriately set.

Step5: No broadcast is needed in this step.

In conclusion, the overhead for most of the sensor nodes is three local broadcasts if the backoff time in Step 2 and Step

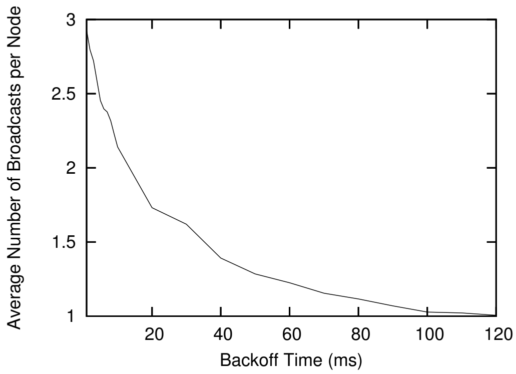


Fig. 3. Average Number of Broadcasts per Node in Step 2

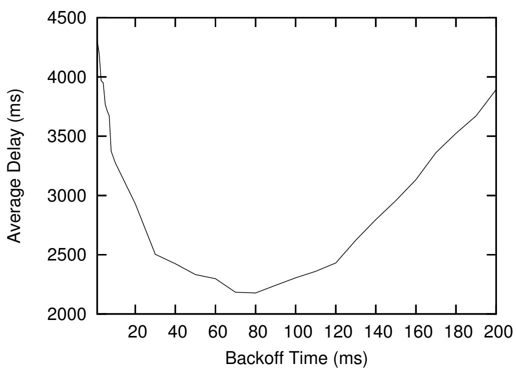


Fig. 4. Average Delay to Complete Step 2

4 are large enough. The whole system setup process can be finished within several seconds, given a sensor network system similar to the one in our simulation.

F. Advantages of the joint scheduling algorithm

First, the joint scheduling method can guarantee that the resulting coverage quality is above a given requirement, since the extra-on nodes actually increase the coverage quality provided by the randomized coverage-based scheme. Later performance evaluation demonstrates that the number of extra-on sensor nodes is not large in any time slot; hence good coverage with guaranteed connectivity is not at the cost of large energy waste.

Second, the route from each sensor node to the sink node has the minimum or nearly minimum hop count. This feature of our joint scheduling method roughly eliminates the extra energy consumption on data delivery with unnecessarily longer paths.

Third, with the joint scheduling method, the routing problem is simultaneously addressed with the connectivity problem. A sensor node not only has a route to the sink node, but also knows the upstream node in this route. Therefore, no additional routing protocols are needed.

Fourth, the overhead to set up the system is small. As demonstrated in the previous section, most sensor nodes need only three local broadcasts if the backoff times in Step 2 and Step 4 of the joint scheduling algorithm are set appropriately.

Finally, since each sensor node uses only local information to make its scheduling decision, the joint scheduling method is purely distributed and is scalable well to large and dense networks.

Note that our joint scheduling scheme exploits the redundancy in both sensing coverage and network connectivity. However, we cannot exclude the possibility that some nodes are critical nodes for connectivity, i.e., the network will be partitioned if these nodes are turned off. Therefore, to maintain network connectivity requirement, our joint scheduling method has to turn on these critical nodes all the time. Therefore, these critical nodes might die sooner than other nodes due to their heavier workload. We point out that there is no algorithm to solve this problem unless more nodes are deployed nearby these critical nodes to increase redundancy. Also, note that the joint scheduling is decoupled from the MAC layer protocol. Although the network topology is different from time slot to time slot, within a single time slot, the network topology is fixed and any MAC layer protocol can be used.

V. Performance Analysis

A. Performance Metric

There is a clear trade-off in the randomized scheduling algorithm. Generally, a larger k value means more subsets, and thus a subset can wait longer until its next turn to work. As such, the network can last longer. But a larger k value means smaller number of sensors in each subset and thus potentially worse network coverage. We need to select a proper k value so that the energy can be saved with desirable network coverage. For this, we need to clearly define the network coverage.

Definition 1: Coverage Intensity for a Specific Point. For a given point p in the field, we define the coverage intensity for this point as

$$C_p = \frac{T_c}{T_a}$$

where T_a is any given long time period and T_c is the total time during T_a when point p is covered by at least one active sensor.

It is obvious that C_p depends on both the number of sensor nodes deployed in the neighborhood of p and the scheduling scheme. Due to the randomness in the sensor deployment strategy and the scheduling scheme, C_p is a random variable. Hence, the expectation of C_p reflects the *average* time fraction when point p can be monitored. Notably, the expectation of C_p for any point inside the field is equal because sensors are independently and uniformly distributed in the field (For simplicity, we ignore the edge effect for large deployment area). Because of this reason, the expectation of C_p is a network-wide consistent metric and could be used to evaluate the coverage quality of the whole network.

Definition 2: Network Coverage Intensity. We define the network coverage intensity, C_n, as the expectation of C_p. That is, $C_n = E[C_p]$.

Since the main task of wireless sensor networks is to detect and report interesting events within the monitored field and the coverage intensity C_n reflects the probability that an event can be detected, C_n can be considered as the coverage measurement of sensor networks. The ideal value of C_n is 1, which indicates that with the probability 1 every point in the field is covered by at least one active sensor at any given time. But achieving this ideal value may require very dense deployment and is extremely expensive. Since different applications have different requirements on acceptable coverage intensity, a good scheduling scheme should set the number of simultaneous working sensor nodes merely enough to fulfill a given coverage requirement. In the sequel, we will investigate the relationship among achievable coverage, energy saving, and the minimum number of required sensor nodes.

B. Analysis on Coverage Intensity

For easy reference, all notations used in our probabilistic analysis are listed in Table I.

Theorem 1: Without considering network connectivity, $C_n = 1 - \left(1 - \frac{q}{k}\right)^n$, where $q = \frac{r}{a}$ is the probability that each sensor covers a given point.

Proof: Suppose that a given point inside the monitored field is covered by s sensor nodes, which construct a set denoted as set S. The randomized algorithm will assign each sensor node in S to one of the k disjoint subsets randomly. Let's consider the question of how many subsets do not include any sensor in S. For the first subset, denoted as subset 0, it must miss all the s sensor nodes to let the above event happen. Since each sensor node hits the first subset independently with same probability of $\frac{1}{k}$,

$$\Pr\{S_0 \text{ } is \text{ } empty\} = \left(1 - \frac{1}{k}\right)^s$$

and thus

$$\Pr\{S_0 \text{ } is \text{ } not \text{ } empty\} = 1 - \left(1 - \frac{1}{k}\right)^s$$

This probability is the same for all subsets by symmetry.

We define a random variable X_j. $X_j = 0$ if S_j is empty and $X_j = 1$ otherwise. Let $X = \sum_{j=0}^{k-1} X_j$ denote the total number of nonempty $S_j, (0 \le j \le k-1)$. Then

$$E[X] = \sum_{j=0}^{k-1} E[X_j] = k \times \left[1 - \left(1 - \frac{1}{k}\right)^s\right]$$

According to the definition of C_p, the coverage intensity for point p, which is covered by s sensor nodes, is

$$C_p = \frac{E[X] \times T}{k \times T} = 1 - \left(1 - \frac{1}{k}\right)^s$$

Here s is a binomial random variable, and

$$\Pr\{s = j\} = \binom{n}{j} \times q^j \times (1-q)^{n-j}$$

where $q = \frac{r}{a}$ is the probability that each sensor covers a given point.

Therefore, the network coverage intensity C_n, which is the expectation of C_p, can be calculated as

$$\begin{aligned} C_n &= E[C_p] \\ &= E\left[1 - \left(1 - \frac{1}{k}\right)^s\right] \\ &= 1 - \sum_{j=0}^{n} \left(1 - \frac{1}{k}\right)^j \times \binom{n}{j} \times q^j \times (1-q)^{n-j} \\ &= 1 - \sum_{j=0}^{n} \binom{n}{j} \times \left(q - \frac{q}{k}\right)^j \times (1-q)^{n-j} \\ &= 1 - \left(1 - \frac{q}{k}\right)^n \square \end{aligned}$$

Corollary 1: For a given k, the lower bound on the number of sensor nodes required in the whole network to provide a network coverage intensity of at least t is

$$\left\lceil \frac{\ln(1-t)}{\ln(1-\frac{q}{k})} \right\rceil$$

where $q = \frac{r}{a}$.

proof: Based on Theorem 1, if we predefine the value of k and we require that the network coverage intensity is no less than a threshold value t, we can compute the lower bound on the number of sensor nodes required to fulfill the task, by solving the inequality

$$1 - \left(1 - \frac{q}{k}\right)^n \ge t.$$

It is easy to see that

$$n \ge \left\lceil \frac{\ln(1-t)}{\ln(1-\frac{q}{k})} \right\rceil . \square$$

Theorem 1 and Corollary 1 illustrate clearly the relationship among the coverage, energy saving, and the minimum number of sensor nodes. For example, if we set $t = 0.9, q = \frac{\pi}{400}$, and $k = 3$, we can get $n \ge 878$ with Corollary 1, which means at least 878 sensor nodes are required in the whole network to achieve a network coverage intensity of no less than 0.9. This example also implies that, if we use 3-disjoint subsets and the ratio of each sensor's sensing range over the network size is $\frac{\pi}{400}$, the average network density is around 6 to meet a network coverage intensity of 0.9. Note that the value 6 is estimated by calculating $\frac{878 \times \pi}{400} - 1$. From this example, it is really exciting that the randomized algorithm can achieve a reasonably high coverage requirement with a moderate network density.

We performed simulation to validate our analytical results. Figure 5 illustrates the relationship between the number of deployed sensor nodes and network coverage intensity with the randomized scheduling scheme. Both analytical results and simulation results are presented in the figure. From the figure, the analytical results and simulation results match pretty well, indicating the correctness of our mathematical analysis. Given a fixed k, network coverage intensity increases with the increase of the number of deployed sensor nodes, and given a fixed number of deployed sensor nodes, network coverage intensity increases with the decrease of k. It is consistent with the intuition since decreasing k or increasing the number of sensor nodes will increase the density of working nodes and hence improves the coverage intensity. This figure also

TABLE I
NOTATIONS

Symbol	Description
n	the total number of deployed sensor nodes
T	the time duration of each time slot
a	the size of the whole field
r	the size of sensing area of each sensor
k	the number of disjoint subsets
s	the number of sensors that cover a specific point inside the field
S	the set of sensors that cover a specific point inside the field
s_i	the number of sensors that belong to subset i and cover a specific point inside the field
S_i	the set of sensors that belong to subset i and cover a specific point inside the field
C_p	Coverage intensity for a specific point
C_n	Network coverage intensity

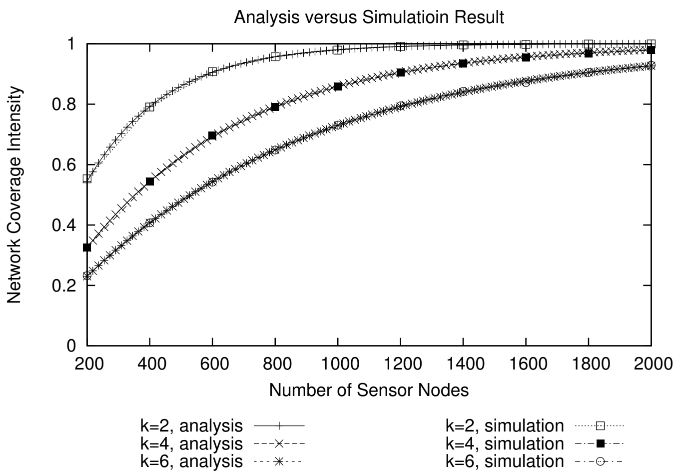


Fig. 5. Analytical and Simulation Results

provides a reference map between network coverage intensity and the number of sensor nodes needed.

Based on Theorem 1, we can easily get the following corollary:

Corollary 2: For a given n, the upper bound on the number of disjoint subsets to provide a network coverage intensity of at least t is

$$\frac{q}{1-e^{\frac{\ln(1-t)}{n}}} = \frac{q}{1-\sqrt[n]{1-t}}$$

where $q = \frac{r}{a}$.

Corollary 2 is very useful in dynamically adjusting the coverage of a sensor network after it is deployed. When the total number of sensor nodes is fixed, the network coverage intensity can be adjusted by changing the number of disjoint subsets k. This feature is extremely useful for practical sensor networks requiring adjustable measurement quality for energy saving. Using Corollary 2, we can easily map the coverage requirement to an appropriate k value so that a perfect balance between energy conservation and coverage can be achieved. The coverage can be adjusted by a simple message flooding to inform all sensor nodes about the new k value.

C. Analysis on Detection Delay and Detection Probability

Besides coverage intensity, users sometimes may be interested in the detection probability and the average detection delay. The detection probability is defined as the probability that the occurrence of an event can be detected by one or more sensor nodes. The average detection delay is defined as the expectation of the time elapsed from the occurrence of an event to the time when the event is detected by some sensor nodes.

For the event lasting for a duration larger than $(k-1)\times T$, if there is at least one sensor node covering the occurrence location of the event, it will be detected with probability 1. Here k denotes the number of total subsets and T denotes the duration of each time slot. In this case, a short average detection delay is desirable. In the following, we investigate the average detection delay for events lasting longer than $(k-1)\times T$, as well as the detection probability for other shorter events.

1) Average Detection Delay: With random deployment, it is possible that there exist some blind points in the field, which cannot be covered by any sensor nodes. For these blind points, the detection delay is infinite. In the following analysis, we do not consider blind points since it is meaningless for such evaluation if a point cannot be covered at any time.

Theorem 2: Assume that an event arrives at any time slot with equal probability and lasts for a duration longer than $(k-1)\times T$. With the randomized coverage-based scheduling algorithm, the average detection delay for an event occurring at a point covered by s sensor nodes, is equal to $\frac{T}{2}\left[\left(\frac{k-1}{k}\right)^s + 2\sum_{i=2}^{k-1}\left(\frac{k-i}{k}\right)^s\right]$

Proof: Without losing generality, we assume that the event arrives at time slot 0, followed by time slots $1, 2, ..., k-1$. Time slot $i (0 \le i \le k-1)$ is associated with the working shift of subset i. Therefore, time slot 0 to time slot $k-1$ consist of a scheduling cycle. Let H_i be the event that none of the s covering sensors belongs to subset i and $\overline{H_i}$ be the event that at least one of the s covering sensors belongs to subset i. Therefore, the average detection delay, $delay_s$, can be calculated as

$$delay_s = \sum_{i=1}^{k-1} \int_0^T \frac{1}{T} * Pr\left(H_0 \bigcap H_1 \bigcap \ldots \bigcap \overline{H_i}\right) *$$
$$(i * T - t)\, dt$$
$$= \sum_{i=1}^{k-1} \int_0^T \frac{1}{T} * \left(1 - \frac{1}{k}\right)^s * \left(1 - \frac{1}{k-1}\right)^s * \ldots *$$
$$\left[1 - \left(1 - \frac{1}{k-i}\right)^s\right] * (i * T - t)\, dt$$
$$= \sum_{i=1}^{k-1} \frac{(2i-1) * T}{2} \left[\left(\frac{k-i}{k}\right)^s - \left(\frac{k-i-1}{k}\right)^s\right]$$
$$= \frac{T}{2}\left[\left(\frac{k-1}{k}\right)^s + 2\sum_{i=2}^{k-1}\left(\frac{k-i}{k}\right)^s\right] \square$$

From the above expression, it is easy to see that the average detection delay for a specific point covered by s sensor nodes is influenced by three factors.

1) T**:** the time duration of each time slot, i.e., the working duration of each subset in one round of scheduling. The average detection delay increases with the increase of T. This is because a larger T will lead to a larger waiting time for the event to be detected by the active sensor nodes of the next working subset, if no node in the current subset can detect the event.
2) k**:** the number of total disjoint subsets. The average detection delay increases with the increase of k (the derivative of $delay_s$ regarding k is always positive when $k > 0$ and thus $delay_s$ monotonically increases with the increase of k). This is because increasing k will potentially increase the probability that no node in a subset can detect the event and hence prolongs the event detection delay.
3) s**:** the number of sensor nodes that cover point p. The average detection delay decreases with the increase of s. This is because a larger s value results in a less chance of generating a subset such that none of its nodes can cover p.

2) Detection Probability: For events lasting less than $(k-1) \times T$ and occurring at a point covered by s sensor nodes, we calculate its detection probability, dp_s.

Theorem 3: Let $l(< (k-1)T)$ denote the duration of an event occurring at a point covered by s sensor nodes. With the randomized coverage-based scheduling algorithm, the detection probability of the event, dp_s, is equal to $1-(1-\gamma_1)\beta_1 - \gamma_1\beta_2$, where $\gamma_1 = \frac{l}{T} - \left\lfloor \frac{l}{T} \right\rfloor, \beta_1 = (1 - \frac{\lceil l/T \rceil}{k})^s$, and $\beta_2 = (1 - \frac{\lceil l/T \rceil + 1}{k})^s$.

Proof: For an event with duration $l(< (k-1)T)$, it can span either $\left\lceil \frac{l}{T} \right\rceil$ or $\left\lceil \frac{l}{T} \right\rceil + 1$ time slots. Therefore, dp_s is equal to the probability that either case occurs, conditioned on the probability that in the corresponding case there are some nodes detecting the event.

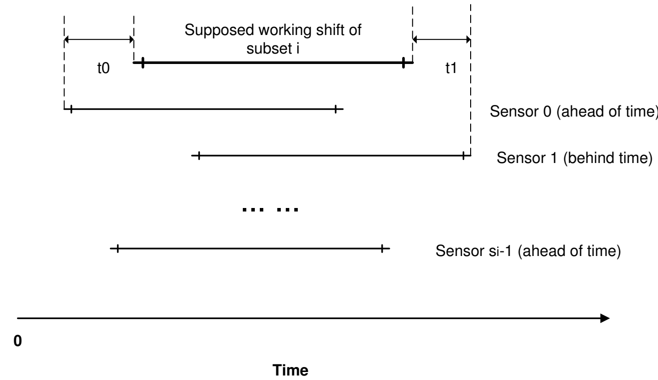


Fig. 6. A point p monitored by s_i sensor nodes in subset i

$$dp_s = (1 - (\frac{l}{T} - \left\lfloor \frac{l}{T} \right\rfloor)) \times \left[1 - (1 - \frac{\lceil l/T \rceil}{k})^s\right] +$$
$$(\frac{l}{T} - \left\lfloor \frac{l}{T} \right\rfloor) \times \left[1 - (1 - \frac{\lceil l/T \rceil + 1}{k})^s\right]$$
$$= 1 - (1-\gamma_1)\beta_1 - \gamma_1\beta_2,$$

where $\gamma_1 = \frac{l}{T} - \left\lfloor \frac{l}{T} \right\rfloor$, $\beta_1 = (1 - \frac{\lceil l/T \rceil}{k})^s$, and $\beta_2 = (1 - \frac{\lceil l/T \rceil + 1}{k})^s$. Note that γ_1 is the probability that the event spans $\left\lceil \frac{l}{T} \right\rceil + 1$ time slots. From the above expression, the detection probability for a specific point covered by s sensor nodes, dp_s, is influenced by three factors.

1) T**:** The detection probability decreases with the increase of T. It is not apparent from the formula given above, but can be easily observed with numerical results. This is because a larger T will make the event span fewer time slots and hence decrease the detection probability.
2) k**:** The detection probability decreases with the increase of k. This is because a large k value will increase the number of subsets that does not include a node to cover the point.
3) s**:** The detection probability increases with the increase of s. This is because a larger s value results in a smaller chance that a subset does not include any sensor to cover the point.

D. Analysis on the Impact of Clock Asynchrony on Coverage Quality

1) A Glance at Clock Asynchrony: Intuitively, the randomized scheduling algorithm should work well without requiring strict time synchronization. Let's check the example shown in Figure 6. A point p in the monitored region is covered by s_i sensor nodes in the subset i. Assume that among the s_i sensor nodes, some sensor nodes (e.g., sensor node 0) are ahead of the supposed starting time while some (e.g., sensor node 1) are behind the time. In this example, point p can be monitored during the whole working shift of subset i even if the sensor nodes are not synchronized very well.

There are only three possibilities that point p may not be monitored during the working shift of subset i:

1) All the s_i sensor nodes are ahead of the starting time of subset i.
2) All the s_i sensor nodes are behind the starting time of subset i.
3) Some sensor nodes in S_i are ahead of the starting time of subset i while some in S_i are behind the time, and there is a gap period when no sensor node in S_i can monitor point p during the working shift of subset i.

2) Analysis on the Impact of Clock Asynchrony: To facilitate analysis, we make the following assumptions:

1) We assume that the internal time ticking frequency of each sensor node is accurate but may not be synchronized precisely to the standard time.
2) We assume that the clock drift of each sensor node from the standard time, Δt, is a random variable following a normal distribution with parameters $(0, \sigma)$.
3) If we use T to normalize Δt, we assume $\Delta t \geq \frac{T}{2}$ is an extremely rare case and could be ignored.

For a point p in the region, we suppose that there are s_i sensor nodes assigned to subset i $(1 \leq i \leq k)$ covering p. Let Δt_j denote the deviation of the clock of the j-th sensor node from the standard clock $(0 \leq j \leq s_i - 1)$. Δt_j is a random variable following a normal distribution with parameters $(0, \sigma)$. If $\Delta t_j \leq 0$ holds for all j $(0 \leq j \leq s_i - 1)$, indicating that all the clocks of these s_i sensor nodes are ahead of time, there will be a period of unmonitored time at the end of working duration of subset i with the length of $\min\{-\Delta t_j, 0 \leq j \leq s_i - 1\}$. Likewise, if $\Delta t_j \geq 0$ holds for all j $(0 \leq j \leq s_i - 1)$, indicating that all the clocks of these s_i sensor nodes are behind time, there will be a period of unmonitored time at the beginning of working duration of subset i with the length of $\min\{\Delta t_j, 0 \leq j \leq s_i - 1\}$.

Note that the sensor nodes with an ahead-of-time clock in subset $i + 1$ and the sensor nodes with a behind-time clock in subset $i - 1$ could help decrease the unmonitored time length during the working duration of subset i. Nevertheless, considering these cases will greatly increase the analysis complexity by introducing correlation between neighboring subsets, we ignore these cases when calculating the network coverage intensity. Therefore, the calculated network coverage intensity is the lower bound of its actual value.

We now calculate the expectation of unmonitored-time fraction (the time when p is not covered by any of these s_i sensor nodes) during the working shift of subset i. We denote this expectation as E_{s_i}.

When $s_i = 0$, it is obvious that $E_0 = 1$. When $s_i \geq 0$,

$$E_{s_i} = \int_0^{\infty} x f_1(x)\, dx + \int_{-\infty}^{0} -y f_2(y)\, dy$$

where $x = \min\{\Delta t_j, 0 \leq j \leq s_i - 1\}$, $y = \max\{\Delta t_j, 0 \leq j \leq s_i - 1\}$, $f_1(x)$ and $f_2(y)$ are the p.d.f. (probability density function) of x and y, respectively.

Since $\Delta t_0, \Delta t_1, \cdots, \Delta t_{s_i - 1}$ are independent random variables following a normal distribution, we can get

$$Pr\{x \geq a\} = [1 - \Phi(a)]^{s_i}$$

where $\Phi(a)$ is c.d.f. (cumulative distribution function) of normal distribution. Therefore,

$$f_1(x) = s_i \phi(x) [1 - \Phi(x)]^{s_i - 1}$$

where $\phi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}}$ and $\Phi(x) = \int_{-\infty}^{x} \phi(x)\, dx$.
By symmetry, we have

$$\int_{-\infty}^{0} -y f_2(y)\, dy = \int_0^{\infty} x f_1(x)\, dx$$

Therefore,

$$E_{s_i} = 2 \int_0^{\infty} s_i x [1 - \Phi(x)]^{s_i - 1} \phi(x)\, dx.$$

Since x follows a normal distribution with parameters $(0, \sigma)$, when $x \geq 0$, we have $\Phi(x) \geq \frac{1}{2}$, and thus $1 - \Phi(x) \leq \frac{1}{2}$.

So we get

$$E_{s_i} \leq 2 \int_0^{\infty} s_i x \left(\frac{1}{2}\right)^{s_i - 1} \phi(x)\, dx = \frac{s_i \sigma}{\sqrt{2\pi}} \left(\frac{1}{2}\right)^{s_i - 2}$$

Here, E_{s_i} is the expectation of the unmonitored time fraction during the working-shift of subset i, which includes exactly s_i sensor nodes covering the point p. Suppose that the total number of sensor nodes in the network that cover point p is s. Any subset may contain j sensor nodes to cover p, where j varies from 0 to s. Therefore, we can calculate the expectation of unmonitored time fraction for any subset (denoted as $\bar{E}_s$):

$$\begin{aligned}
\bar{E}_s &= \sum_{j=0}^{s} E_j \times Pr\{the\ subset\ contains\ j\ nodes\ to\ cover\ p\} \\
&\leq 1 \times \left(1 - \frac{1}{k}\right)^s + \sum_{j=1}^{s} \frac{j\sigma}{\sqrt{2\pi}} \left(\frac{1}{2}\right)^{j-2} \left(1 - \frac{1}{k}\right)^{s-j} \left(\frac{1}{k}\right)^j \binom{s}{j} \\
&= \left(1 - \frac{1}{k}\right)^s + \frac{2s\sigma}{\sqrt{2\pi}k} \left(1 - \frac{1}{2k}\right)^{s-1}
\end{aligned}$$

Thus, for any point covered by s sensor nodes, the expectation of the monitored time fraction of the working-shift of any subset

$$E_s = 1 - \bar{E}_s \geq 1 - \left(1 - \frac{1}{k}\right)^s - \frac{2s\sigma}{\sqrt{2\pi}k} \left(1 - \frac{1}{2k}\right)^{s-1}$$

We next calculate E, the expectation of E_s.

$$\begin{aligned}
E &= \sum_{s=0}^{n} E_s \times Pr\{the\ point\ is\ covered\ by\ s\ sensor\ nodes\} \\
&= \sum_{s=0}^{n} E_s \binom{n}{s} q^s (1 - q)^{n-s},\ where\ q = \frac{r}{a} \\
&\geq 1 - \sum_{s=0}^{n} \left(1 - \frac{1}{k}\right)^s \binom{n}{s} q^s (1 - q)^{n-s} \\
&\quad - \frac{2s\sigma}{\sqrt{2\pi}k} \sum_{s=0}^{n} \left(1 - \frac{1}{2k}\right)^{s-1} \binom{n}{s} q^s (1 - q)^{n-s} \\
&= 1 - \left(1 - \frac{q}{k}\right)^n - \frac{2nq\sigma}{\sqrt{2\pi}k(1 - q)} \left(1 - \frac{q}{2k}\right)^{n-1}
\end{aligned}$$

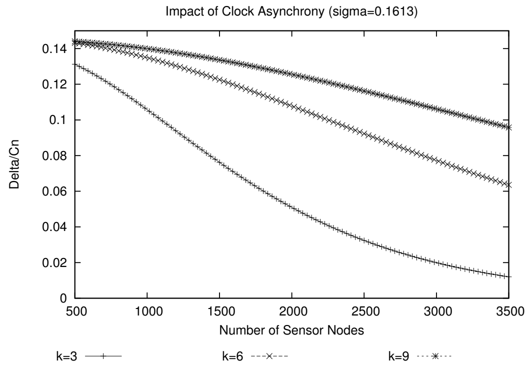


Fig. 7. Impact of Clock Asynchrony

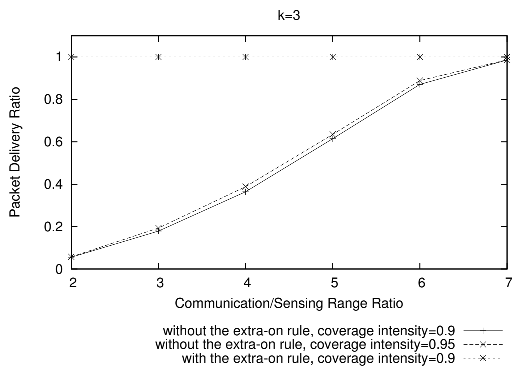


Fig. 8. The Packet Delivery Ratio

For any point p, by symmetry, each subset has the same E value, so the expectation of monitored time fraction, which is the network coverage intensity C_n, according to the definition, can be calculated as

$$C_n = \frac{k \times E}{k} = E$$

Observing the expression of C_n above, we find that the term $1 - \left(1 - \frac{q}{k}\right)^n$ is equal to the C_n in Section V-B, where all the clocks are assumed well-synchronized. Thus, the last term $\Delta = \frac{2nq\sigma}{\sqrt{2\pi}k(1-q)}\left(1 - \frac{q}{2k}\right)^{n-1}$ indicates the impact of time asynchrony on network coverage intensity. The numeric results in Figure 7 show the weight of Δ over C_n when $\sigma = 0.1613$. The small values of the weight indicate that the impact of clock asynchrony on coverage is negligible and that the randomized scheduling scheme is resilient to time asynchrony.

VI. Simulation Evaluation

A. Simulation Settings

We evaluate the performance of the joint scheduling algorithm on a simulator we implemented in Java. We use the CSMA MAC layer protocol. In our simulation, we deploy sensor nodes randomly in a 200 meters * 200 meters square region. The sink node is located at the center of the region. The total number of sensor nodes is selected to meet any given network coverage intensity, according to Corollary 1. The sensing range of each sensor node is fixed to 10 meters. We normalize the communication range with the sensing range and use the ratio of the communication range over the sensing range as the measure of the communication range. The traffic load is very light such that packet losses are mainly caused by network partition or channel errors. Under each simulation scenario, 100 runs with different random seeds are executed.

We use the following metrics to evaluate the joint scheduling algorithm:

Packet Delivery Ratio: It is defined as the ratio of total number of packets received at the sink node over the total number of transmitted packets from sensor nodes. Because the traffic load is very light, this metric is an indicator of network connectivity.

The Ratio of Nodes Having the Shortest Path: It is defined as the number of nodes that have the shortest path to the sink node over the total number of nodes. It is an indicator of path optimality.

The Ratio of Extra-on Sensor Nodes: It is defined as the ratio of the number of sensor nodes, which must remain active beyond their regular working shifts assigned by the randomized coverage-based scheduling algorithm, to the total number of deployed sensor nodes. A small ratio of extra-on sensor nodes indicates that small extra energy is required to maintain connectivity after the coverage requirement is granted.

Network Coverage Intensity: It is defined in Section V-A and is a measure of coverage quality.

B. Network Connectivity and Path Optimality

To demonstrate that the extra-on rule can assure network connectivity, we compare the packet delivery ratio with and without the extra-on rule applied to the network. The number of deployed sensor nodes is determined by the network coverage intensity and k. In this test, to preclude the packet losses due to broadcast collision or channel errors, we adopt a perfect radio channel without medium contention. In all the simulated scenarios, the networks are connected if all the deployed sensor nodes are active. After randomly turning off redundant sensor nodes without applying the extra-on rule, the networks are partitioned. This is evident from the fact that the packet delivery ratio cannot achieve 100% without using the extra-on rule as shown in Figure 8. However, with the extra-on rule, the networks can always achieve a 100% packet delivery rate, indicating that the extra-on rule provides guaranteed network connectivity.

As discussed in Section IV-D, if there exists packet losses, some nodes may not have the shortest path to the sink node. Nevertheless, from Figure 9, we can see that at the end of Step 2 of the joint scheduling scheme, even if the packet loss rate is as high as 10%, the ratio of nodes having the shortest path to the sink node is no less than 95%.

C. Ratio of Extra-on Sensor Nodes

Apparently, there are three factors influencing the ratio of extra-on sensor nodes: network coverage intensity, the number of subsets, and the ratio of the communication range over the sensing range.

To investigate the influence of network coverage intensity, we fix the number of subsets and vary the communication

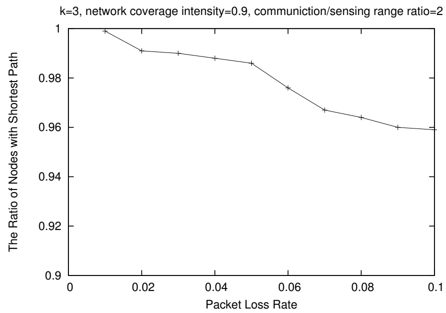


Fig. 9. The Ratio of Nodes Having the Shortest Path

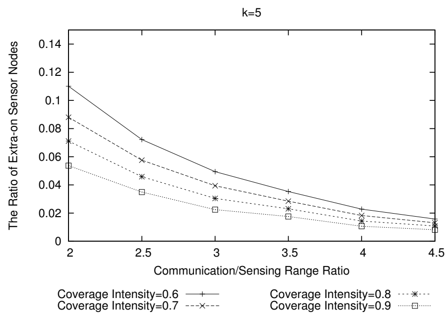


Fig. 10. Influential Factors of Average Number of Extra-On Nodes (K is fixed)

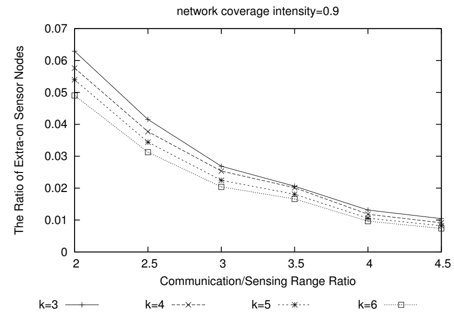


Fig. 11. Average Number of Extra-On Nodes (Coverage Intensity is fixed)

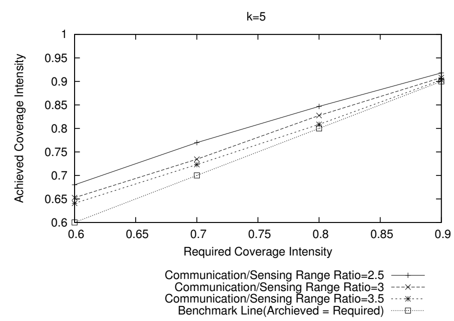


Fig. 12. Achieved Coverage Intensity Vs Required Coverage Intensity

range and the network coverage intensity. As shown in Figure 10, the ratio of extra-on sensor nodes drops with the increase of the coverage intensity and the communication range. This is because when the coverage intensity increases, more sensor nodes will remain active for coverage, hence few extra nodes are needed for network connectivity. In addition, the increase of communication range enhances the connectivity of the original networks, resulting in the decrease of the number of extra-on sensor nodes.

Similarly, to investigate the influence of the number of subsets, we fix the network coverage intensity and vary the communication range and the number of subsets. As shown in Figure 11, increasing communication range decreases the ratio of extra-on sensor nodes, due to the same reason mentioned above.

From Figure 11, we can see that the ratio of extra-on nodes decreases with the increase of k. Given fixed coverage intensity, the number of simultaneous active sensor nodes scheduled by the randomized scheduling algorithm is roughly the same and is independent of the value k. That is, the density of the network after the randomized scheduling only depends on the coverage intensity. Since network connectivity is mainly determined by network density, the number of extra-on nodes should be roughly the same in order to maintain network connectivity. Since for a given coverage intensity a larger k value means a larger total number of deployed sensor nodes, the ratio of extra-on sensors decreases with the increase of k.

In Figure 10 and Figure 11, when the coverage intensity is sufficiently high and the communication range is sufficiently large, the number of extra nodes needed to turn on for connectivity maintenance is very small, compared to the total number of active nodes for coverage. Therefore, with our joint scheduling approach, the extra energy consumption on connectivity maintenance is small and the network coverage intensity does not unnecessarily exceed a given requirement too much.

D. Network Coverage Intensity

The achieved network coverage intensity by our joint scheduling method is illustrated in Figure 12. As expected, the achieved coverage intensity is always slightly higher than the required coverage due to the fact that extra sensor nodes are needed to turn on to remain network connectivity.

VII. RELATED WORK

The work in [8] provides the first asymptotic result on the relationship between the power level and the network connectivity. By using percolation theory, it proves that in order to maintain connectivity in a network with randomly placed nodes, the average node degree of the network should be in the order of $(logn + c)$, where n is the total number of nodes and c is a constant. Another similar work could be found in [23].

In wireless sensor networks, topology management protocols are needed to maintain the network connectivity. To save energy, redundant (in terms of network connectivity) sensor nodes can be turned off to save energy without degrading the network connectivity. The existing topology management protocols for wireless sensor networks include [22], [4], [3], [13].

With GAF [22], the area is divided into cells with same size by a set of virtual grids. The size of cell is small enough so that any node can communicate directly with any other nodes within its neighboring cells. Therefore, only one node is required to be active within a virtual cell in order to maintain the network connectivity. SPAN [4] maintains a routing backbone and allows sensor nodes that do not belong to the backbone to sleep. The backbone nodes are also called "coordinators". A coordinator selection process is triggered periodically to balance the energy consumption among different nodes, since compared to a non-coordinator node, a coordinator consumes more energy to relay data for other nodes. ASCENT [3] is similar to SPAN in the sense that it chooses some sensor nodes to be active as routers while allows others to sleep to save energy. However, unlike SPAN, ASCENT selects an active router depending on not only local connectivity, but also the observed data loss rates. Therefore, ASCENT can obtain a strongly connected network with more reliable transmissions. STEM [13] puts sensor nodes into sleep state more aggressively. Sensor nodes wake up only when they have data to transfer or they receive requests from their neighbors to forward the data to the sink node. A separated paging channel is dedicated to the wakeup operation. Therefore, STEM trades latency for further energy saving.

In addition to maintaining network connectivity, maintaining a sufficient sensing coverage is also a critical requirement of sensor networks, since sensing coverage directly determines the monitoring quality provided by sensor networks in a designated region. By turning off redundant (in terms of coverage) sensor nodes, the coverage quality can be maintained and energy efficiency can be achieved at the same time. The existing coverage-preserving scheduling schemes for wireless sensor networks include [1], [9], [11], [12], [16], [21], [24], [25].

Unfortunately, most existing work addresses the connectivity problem and the coverage problem in isolation. They only solve one problem without considering the other. However, to operate successfully, a sensor network must ensure network connectivity and coverage quality at the same time. Treating coverage and connectivity in a unified framework for coverage preservation and connectivity maintenance is critical in building up practical sensor network applications.

Recently, in [14], the joint problem of coverage and connectivity is considered in a network with sensor nodes deployed strictly in grids. Each sensor node can probabilistically fail. The sufficient and necessary conditions for connectivity and coverage in this type of networks are provided.

The joint problem in more general sensor networks where the sensor nodes are deployed at random is also investigated. In [26], it is proved that to ensure that a full coverage of a convex area also guarantees the connectivity of the active nodes inside the area, the communication range should be at least twice of the sensing range. Therefore, the joint problem is simplified to maintain a complete coverage of a convex region if the communication range is greater than twice of the sensing range. In [17], the authors enhance the work in [26] by releasing the constraint. They proved that "the communication range is twice of the sensing range" is the sufficient condition and the tight lower bound to ensure that complete coverage implies connectivity, no matter the area is a convex area or not.

In paper [19], the authors draw the same conclusion as in paper [26]. In addition, they present a Coverage Configuration Protocol (CCP) that can provide fully coverage of a convex region. In the case where the communication range is greater than twice of the sensing range, the connectivity of networks is guaranteed by the full coverage, so no mechanism for connectivity maintenance is needed. To deal with the situation where the communication range is less than twice of the sensing range, the authors propose to integrate CCP with a topology management protocol SPAN [4] to provide both coverage and connectivity. The main qualitative differences between our method and CCP+SPAN proposed in [19] are summarized in Table II.

In [7], the authors develop the notion of a connected sensor cover, defined as the sensor set that can fully cover the queried area and constitute a connected communication graph at the same time. The authors also demonstrate that the calculation of the smallest connected sensor cover is NP-hard, and they propose both centralized and distributed approximate algorithms to solve it and provide the performance bounds as well. However, unlike our approach, the method in [7] requires each individual sensor node to be aware of its precise location, in order to check its local coverage redundancy.

Since our approach to dealing with the joint scheduling problem is unique, we point out that it is impossible to provide a fair, quantitative comparison between our method and any existing solution.

VIII. CONCLUSION

Sensor scheduling plays an essential role for energy efficiency of wireless sensor networks. Traditional sensor scheduling methods usually use sensing coverage or network connectivity, but rarely both. Some research [14], [19], [26] has dealt with the joint problem of sensing coverage as well as network connectivity, but requires specific network topology such as grid or strong constraints on the relationship of sensing range and radio transmission range. In this paper, we take a different approach to solving the joint scheduling problem. We use a randomized scheduling method to provide statistical sensing coverage and then switch on extra sensors, if necessary, for network connectivity. Analytical and simulation results demonstrate the effectiveness of our joint scheduling method. More specifically, our joint scheduling method has the following good features:

1) It can achieve substantial energy saving and meet both constraints of coverage and connectivity without relying on location information or any assumptions on the relationship between the sensing range and the radio range.
2) It is totally distributed and thus easy to implement.
3) Each active sensor node knows at least one route with minimum hop count to the sink node. In this sense, our joint scheduling method solves the routing problem and eliminates the system cost incurred by routing protocols.
4) The coverage intensity could be dynamically adjusted by a simple broadcast of the k value.
5) The scheduling method is resilient to time asynchrony.

TABLE II

COMPARISON OF OUR METHOD TO CCP+SPAN

Method	Full Coverage	Dynamic Coverage Adjustment	Location Awareness
Our Method	Not Guaranteed	Allowed	Not Needed
CPP+SPAN	Guaranteed	Not Allowed	Needed

REFERENCES

[1] Z. Abrams, A. Goel, and S. Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," in *Proceedings of the Third International Symposium On Information Processing In Sensor Networks (IPSN 2004)*, Berkeley, California, April 2004.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[3] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sensor networks topologies," in *Proceedings of the IEEE INFOCOM 2002*, New York, NY, June 2002.

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 2001)*, Rome, Italy, July 2001, pp. 85–96.

[5] E. Elson and K. Romer, "Wireless sensor networks: A new regime for time synchronization," in *Proceedings of First Workshop on Hot Topics in Networks*, Princeton, New Jersey, October 2002.

[6] P. Godfrey and D. Ratajczak, "Robust topology management in wireless ad hoc networks," in *Proceedings of the Third International Symposium On Information Processing in Sensor Networks (IPSN 2004)*, Berkeley, California, April 2004.

[7] H. Gupta, S. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," in *Proceeding of the ACM MobiHoc 2003*, Annapolis, Maryland, June 2003.

[8] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity in wireless networks," in *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming.*, Boston, 1998.

[9] C. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithm," in *Proceedings of the Third International Symposium On Information Processing In Sensor Networks (IPSN 2004)*, Berkeley, California, April 2004.

[10] C. Liu, "Randomized scheduling algorithm for wireless sensor neworks," in *Project Report of Randomized Algorithm*, University of Victoria, March 2004.

[11] C. Liu, K. Wu, and V. King, "Randomized coverage-preserving scheduling schemes for wireless sensor networks," in *Proceedings of IFIP Networking Conference 2005*, Waterloo, Canada, May 2005.

[12] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad-hoc sensor neworks," in *Proceedings of IEEE Infocom 2001*, Ankorange, Alaska, April 2001.

[13] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Strivastava, "Topology management for sensor networks: Exploiting latency and density," in *Proceedings of the ACM MobiHoc 2002*, Lausanne, Switzerland, June 2002.

[14] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *Proceedings of IEEE INFOCOM 2003*, San Francisco, CA, April 2003.

[15] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications 2001*, Helsinki, Finland, June 2001.

[16] D. Tian and D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, October 2002.

[17] D. Tian and D. Georganas, "Connectivity maintenance and coverage preservation in wireless sensor networks," *Ad Hoc Networks Journal*, pp. 744–761, 2005.

[18] S. Tilak, N. Abu-Ghazaleh, and H. W., "Infrastructure tradeoffs for sensor networks," in *Proceedings of First International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, September 2002.

[19] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of ACM Sensys, 2003*, Los Angeles, CA, November 2003.

[20] B. Warneke, M. Last, B. Leibowitz, and K. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Ad Hoc Networks Journal*, vol. 34, no. 1, pp. 44–51, January 2001.

[21] K. Wu, Y. Gao, F. Li, and Y. Xiao, "Lightweight deployment-aware scheduling for wireless sensor networks," *ACM/Springer Mobile Networks and Applications (MONET), Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks*, vol. 10, no. 6, pp. 837–852, December 2005.

[22] Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin, "Topology control protocols to conserve energy in wireless ad hoc networks," in *CENS Technical Report 0006*, January 2003.

[23] F. Xue and P. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[24] T. Yan, T. He, and J. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of the First International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, November 2003.

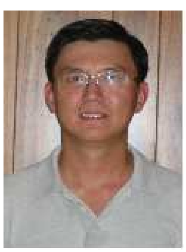
[25] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "Peas: A robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the 10th IEEE International Conference on Network Protocols*, Paris, France, November 2002.

[26] H. Zhang and J. Hou, "Maintaining coverage and connectivity in large sensor networks," in *Proceedings of International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless and Peer-to-Peer Networks (invited paper)*, Florida, U.S.A., February 2004.

Chong Liu is a Ph.D. candidate in Computer Science Department, University of Victoria. His major research interests include energy efficient node clustering, scheduling, and data retrieval in wireless sensor networks.

Kui Wu received the Ph.D degree in computing science from the University of Alberta, Canada in 2002. He then joined the Department of Computer Science, University of Victoria, Canada, where he is currently an Assistant Professor. His research interests include mobile and wireless networks, sensor networks, network performance evaluation, and network security.

Yang Xiao worked at Micro Linear as an MAC (Medium Access Control) architect involving the IEEE 802.11 standard enhancement work before he joined Dept. of Computer Science at University of Memphis in 2002. Dr. Xiao is an IEEE Senior member. He was a voting member of IEEE 802.11 Working Group from 2001 to 2004. He currently serves as Editor-in-Chief for *International Journal of Security and Networks (IJSN)* and for *International Journal of Sensor Networks (IJSNet)*. He serves as an associate editor or on editorial boards for several refereed journals. He serves as a reviewer for many funding agencies, as well as a panelist for NSF. His research interests include wireless security and wireless networks. E-mail: yangxiao@ieee.org.



Bo Sun received his Ph.D. degree in Computer Science from Texas A&M University, College Station, U.S.A., in 2004. He is now an assistant professor in the Department of Computer Science at Lamar University, U.S.A. His research interests include the security issues (intrusion detection in particular) of Wireless Ad Hoc Networks, Wireless Sensor Networks, Cellular Mobile Networks, and other communications systems.