Energy-Efficient Coverage and Connectivity Maintenance for Wireless Sensor Networks

Jehn-Ruey Jiang and Tzu-Ming Sung
Department of Computer Science and Information Engineering
National Central University, Jhongli City, Taoyuan, 32001, Taiwan
jrjiang@csie.ncu.edu.tw

Abstract—**In this paper, we propose a density control algorithm for wireless sensor networks to keep as few as possible sensors in the active state to achieve a connected coverage of a specific area of interest. Inactive sensors can turn off sensing modules to save energy. Unlike other algorithms, the proposed one does not rely on position information or ranging information of sensors. It just requires each active sensor to periodically send two beacons of different transmission ranges. Sensors can then decide to stay active or inactive according to received beacons. The proposed algorithm is fault-tolerant in the sense that one or more inactive sensors can switch to the active state to take over the surveillance responsibility when any active sensor runs out of energy or fails. Under the assumption of sufficiently high density of sensors and the assumption of $R_C \geq 2R_S$, the algorithm can approximate the optimal connected coverage, where R_C and R_S are the radio communication radius and the sensing radius of sensors, respectively. We also perform simulation experiments to demonstrate the algorithm's performance.**

Index Terms—**wireless sensor network, density control, coverage, connectivity, power saving**

I. INTRODUCTION

Advances in the wireless communication and the microelectronic technologies have been expediting the development of *wireless sensor networks (WSNs)* [1]. A WSN consists of a large number of micro sensors with short-range radio and limited data processing capability. Each sensor can sense physical phenomena, such as temperature, jolt, sound, light or magnetic field, and can transmit sensed data to the sink sensor through a multiple-hop communication link (see Fig. 1). WSNs are self-organizing in the sense that they can be formed without human intervention, adapt to sensor failure and degradation, and react to task changes. They have wide applications like battlefield surveillance, environment monitoring, animal tracking and chemical detection, and so on [12].

As most sensors are supported by batteries and manually recharging batteries of deployed sensors is difficult, solutions to increase the network lifetime are important. Deploying a high number of sensors to increase the redundancy in the system can help increase the network lifetime. At a high-density network, *density control* is applied to keep part of the sensors active to guarantee the complete coverage of the monitored area;

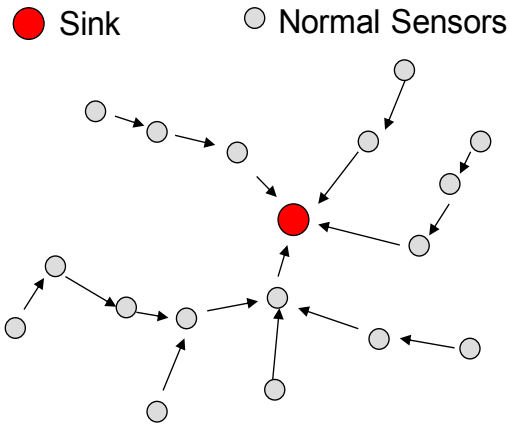

Figure 1. An example of a wireless sensor network (WSN)

other sensors are kept inactive (i.e., in sleep mode or in power saving mode) to save energy to prolong the network lifetime. Density control must satisfy the following two requirements: (1) coverage: the monitored area must be completely covered by sensing areas of active sensors, and (2) connectivity: the active sensors must keep connected so that the sensed data can be relayed to the data sink.

A lot of researches [2] have been conducted in the past addressing sensor deployment and/or density control to achieve connected coverage of the monitored area of a wireless sensor network. Most of the researches are location-based [2, 4, 7, 17-18]. They assume that each sensor knows the positions of itself and its neighboring sensors or that all sensors' positions are known by the sink. Some also assume the relation of the sensor radio communication radius R_C and the sensor sensing radius R_S. For example, the paper [7] assumes $R_C = R_S$. It pursuits almost optimal complete coverage by strip-based deployment under the assumption of $R_C = R_S$. The paper [2] addresses optimal deployment patterns to achieve coverage and connectivity for all ratios of R_C/R_S. The paper [17] proves that the coverage of an area implies the connectivity of the sensors covering the area if $R_C \geq 2R_S$. By this fact, we can pay attention only to achieving coverage in sensor deployment and density control. It is a well known result that the regular hexagon-based deployment (see Fig. 2) can reach complete coverage with the optimal (least) number of sensors [10]. Papers [4] and [18] propose approximation algorithms to find a subset of sensors that ensure both coverage and

connectivity within an O(log n) factor of the optimal deployment, where n is the number of sensors.

A location-based density control algorithm, called Optimal Geographical Density Control (OGDC) [17], tries to maintain connected coverage with a minimal number of sensors under the assumption that sensor locations and boundary information are known in advance. However, location-free algorithms are desirable when sensor location information is not available. There are two location-free density control algorithms proposed in [3] and [13]. They are called *range-based* algorithms since they rely on ranging information which is derived by ranging techniques, such as RSSI (received signal strength indication) [14], for estimating the distances of pairs of sensors. It is well known that ranging schemes are error-prone since they are vulnerable to environmental interference and multi-path fading, etc.

By the discussions above, we may well resort to location-free and range-free algorithms. The density control algorithms proposed in [6][14][16] maintain connected coverage without location information and ranging information. PEAS in [14] uses a probing mechanism for a sleeping sensor to periodically wake up to broadcast a probe message to decide whether to change the state. If there is a reply from a working (active) sensor, then the probing sensor goes back to sleep; otherwise, it becomes a working sensor. Two probabilistic algorithms [6][16] relies on stochastic process for density control. Although location-free and range-free density control algorithms may not achieve 100% coverage, they are very useful for sensor networks where sensors have no location information or need no location information (e.g., the Mars sensor network [5]). However, none of the papers mentioned in this paragraph address how to approach the optimal deployment.

In this paper, we propose a location-free and range-free density control algorithm for wireless sensor networks to keep as few as possible sensors active to approach the optimal connected-coverage deployment of a specific area of interest. Active sensors should keep their sensing modules on, while inactive sensors can turn off their sensing modules to save energy for prolonging the network lifetime. Unlike other algorithms, the proposed one does not rely on position information or ranging information of sensors. It just requires each active sensor to periodically send two beacons of different transmission ranges. Sensors can then decide to stay in the active state or inactive state. When any active sensor runs out of energy or fails, one or more inactive sensors can switch to the active state to take over the surveillance responsibility. The proposed algorithm is thus fault-tolerant. Under the assumption of sufficiently high density of sensors and the assumption of $R_C \geq 2R_S$, the algorithm can approximate the optimal connected coverage, where R_C and R_S are the radio communication radius and the sensing radius of sensors, respectively. Furthermore, we perform simulation experiments to investigate the impact of the sensor density and the ratio α on algorithm performance, where α ($\frac{1}{\sqrt{3}} < \alpha < 1$) is the ratio of the transmission ranges of the two beacons.

The rest of this paper is organized as follows. Section 2 discusses some related work. In Section 3 we describe our algorithm in details. Section 4 introduces a simulation study and performance analyses. Finally, concluding remarks are drawn in Section 5.

2. Related Work

In this section, we introduce the location-free density control algorithms. We begin by introducing three location-free and range-free algorithms. In PEAS [14], all sensors are inactive (in sleep mode) initially. They wake up periodically and asynchronously to broadcast a probe message. On receiving a probe message, an active sensor should reply to it. A probing sensor can go back to sleep mode if it receives a reply; otherwise, it becomes active henceforth. PEAS has a mechanism to adjust the probing rate and it also ensures the connectivity of active sensors by adjusting the transmission of the probe message to be less than $\frac{1}{1+\sqrt{5}}$ times of the transmission range. The papers [6][16] propose density control algorithms using stochastic processes. The paper [6] proposes two algorithms. The first one is a random sleep algorithm, which makes a sensor enter the sleep mode randomly and independently. The second one is a coordinated sleep algorithm, which makes sensors coordinate with each other to decide when to enter the sleep mode and for how long to stay in the mode. The paper in [16] proposes an algorithm using exponentially distributed random variables for a sensor to set the time periods of active and inactive modes. Given the distribution of sensor positions rather than exact sensor positions, the algorithm can set the random variables properly (by adjusting their means) to achieve the expected network coverage specified.

The papers [3] and [13] propose location-free, range-based algorithms utilizing ranging techniques, such as the received signal strength indication (RSSI) scheme [14], to measure the distance of neighboring sensors for density control. In the algorithm proposed in [13], all sensors are inactive initially. A random backoff mechanism is used to select sensors to become active sensors, called starting sensors. Starting sensors should broadcast working messages, and the RSSI scheme is used to measure the distance between two neighboring starting sensors, which in turn is used to determine the intersection statuses of starting sensors. By the statuses, sensors that are redundant in covering the monitored area can turn themselves off, while sensors that are needed to cover uncovered regions can turn themselves on. Each sensor running the algorithm in [3] keeps a neighbor list and a "co-worker" list. The former keeps the ID and an estimated distance of each neighboring sensor, and the latter keeps the ID of known active (working) sensors. Initially, all sensors are in the role-deciding state; a stochastic procedure is used to make some sensors become starting sensors, which should send out a co-worker request message attached with a co-worker list. According to the co-worker messages received, the distances of message senders and the receiver can be estimated by the RSSI scheme. A sensor may then decide

to reply to the message or not according to the relationship of the neighbor list, the co-worker list, and the distance information. And on the basis of reply messages, sensors can decide to keep active or inactive.

3. The Problem and the Algorithm

A. Problem Formulation

We assume a large number of sensors are randomly distributed in a specific area G of interest. We also assume the sensor density is sufficiently high, and sensors are asynchronous and position-less. Each sensor owns the same sensing radius R_S and an adjustable communication radius R_C, where $R_C \geq 2R_S$. Sensors have two possible states: the active state and the inactive state. Active sensors should turn on their sensing modules to monitor area G, while inactive sensors can turn off their sensing modules to save energy. The first problem to solve is how to achieve connected-coverage of G by using the least number of active sensors.

To save more energy, sensors switches the communication modules into the power-saving mode, in which sensors turn on and turn off the radio alternatively. An active sensor should broadcast a beacon periodically so that other sensors, either active or inactive, can be aware of its status. Furthermore, a signal can be issued just after the beacon so that an active sensor can notify its neighboring active sensors that it has pending data to be forwarded. The difficulty for each sensor to hear the beacon and/or the notification is that sensors turn on/off the radio asynchronously. Therefore, the second problem to solve is how to provide a mechanism for sensors to be aware of active sensors' statuses by asynchronously beaconing.

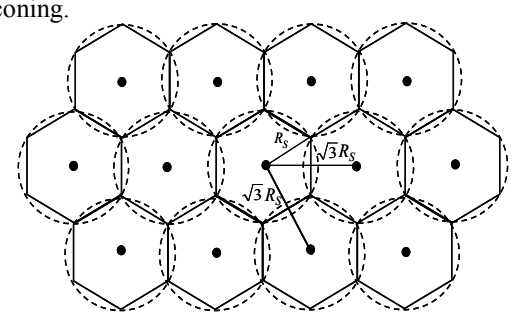

Figure 2. The optimal deployment for the connected coverage of a monitored area (The bold dots denote the sensors, and R_S denotes the sensing radius.)

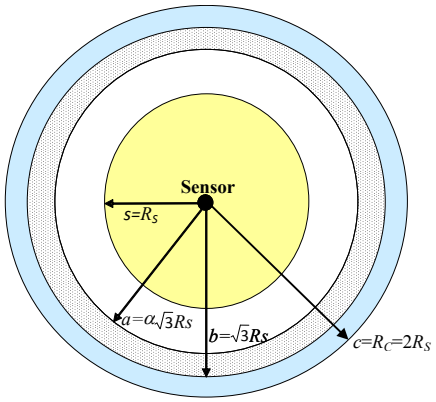

Figure 3. The relation among R_S, R_C and the transmission ranges of a-beacons and b-beacons

B. The Algorithm

Algorithm Overview

Initially, all sensors are inactive. Some sensors become active spontaneously after a random backoff time and start to broadcast beacons; others keep inactive if they are aware of any active sensors by hearing beacons. An active sensor periodically broadcasts two types of beacons: *near-beacon* (or called *a-beacon*) and *far-beacon* (or called *b-beacon*). The two beacons have different transmission ranges: $\sqrt{3}R_S$ for the b-beacon and $\alpha\sqrt{3}\,R_S$ for the a-beacon, where R_S is the sensing radius and $1 > \alpha > \frac{1}{\sqrt{3}} \approx 0.5773$ (see Fig. 3). Note that we set the transmission range of b-beacons to be $\sqrt{3}R_S$, the distance between two neighboring sensors in a regular hexagonal deployment of sensors of the sensing range (or radius) R_S. Also note that when α is about $\frac{1}{\sqrt{3}}$, the transmission range of a-beacons approaches the sensing range R_S. On the other hand, the range approaches $\sqrt{3}\,R_S$ (i.e., the transmission range of b-beacons) when α is about 1.

Active sensors broadcast beacons periodically. When an inactive sensor j can receive b-beacons from an active sensor i, it means that the distance between i and j is less than $\sqrt{3}R_S$. When an inactive sensor j cannot receive a-beacons from an active sensor i, it means that the distance between i and j is larger than $\alpha\sqrt{3}R_S$. Consequently, if an inactive sensor j can receive b-beacons but no a-beacons from an active sensor i, then the distance of i and j is between $\alpha\sqrt{3}Rs$ to $\sqrt{3}Rs$. For example, if the central sensor in Fig. 3 is an active sensor i, then a sensor in the dotted area can receive b-beacons but no a-beacons from sensor i; the sensor can thus infer that the distance of itself and sensor i is between $\alpha\sqrt{3}Rs$ to $\sqrt{3}Rs$.

To make the topology of sensors approach the optimal hexagonal deployment, an inactive sensor can become active only when it can receive b-beacons but no a-beacons from two different active sensors; moreover, when active sensors can hear one another's a-beacons, they are very close and some of them should become inactive. The information of a sensor's "active time" is attached to every beacon so that the sensor with a larger active time will influence the one with a smaller active time, but not vice versa. This is the critical concept of our proposed algorithm to approach the optimal hexagonal deployment.

To save energy, active sensors turn on their radios periodically to broadcast beacons, and inactive sensors also turn on their radios periodically to receive possible beacons. Due to asynchronism of sensors, we need a mechanism to enable sensors to receive beacons from neighboring sensors properly. The proposed algorithm divides the time axis into fixed-length time intervals and embeds special structures in intervals to realize the mechanism. The algorithm also incorporates the data transmission signaling scheme to facilitate active sensors requesting other active sensors to forward sensed data. This can further reduce the consumption of energy and help prolong the network lifetime. Below, we first describe the asynchronous beaconing mechanism.

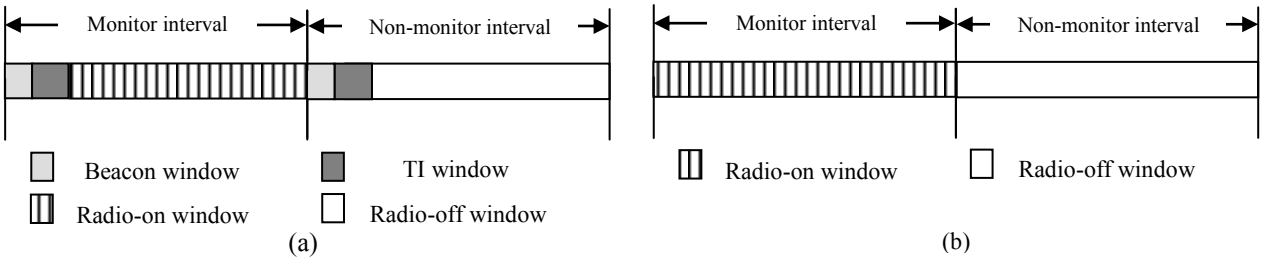

Figure 4. The monitor interval and the non-monitor interval for (a) active sensors and (b) inactive sensors

The Asynchronous Beaconing Scheme

Inspired by the asynchronous power saving protocol proposed in [8], we divide the time axis into beacon intervals which have the same length l (e.g., 100 ms). A group of n consecutive beacon intervals is called a *round*. There are two types of beacon intervals: *monitor intervals* and *non-monitor intervals*. Each sensor should randomly select only one monitor interval from a round of n beacon intervals; that is, there is one monitor interval and n-1 non-monitor intervals in a round of n beacon intervals. As shown in Fig. 4 (a), beacon intervals for active sensors have the following specific structures:

- Monitor interval: This interval starts with the beacon window following the TI (Traffic Indication) window. After the TI window, it is the radio-on window, where sensors have to keep the radio on to monitor all possible radio signals.
- Non-monitor interval: This interval also starts with the beacon window following the TI window. After the TI window, it is the radio-off window. If a sensor has no data to transmit or receive, it can turn off the radio in the radio-off window to save energy.

An active sensor should use a random backoff mechanism to try to broadcast an a-beacon and a b-beacon in succession in every beacon window. After the beacon window, the active sensor still needs to keep the radio on in the TI window. After the TI window, the sensor enters the radio-on window and keeps the radio on in the monitor interval; the sensor enters the radio-off window and turns off the radio to save energy. On the other hand, an inactive sensor just enters the radio-off window and turns off the radio during a non-monitor interval; it enters the radio-on window and turns on the radio during a monitor interval. In this way, inactive sensor can turn off the radio for most of the time (in non-monitor intervals) to save much energy.

According to inference similar to that in [8], we can easily check that every sensor, either active or inactive, can hear at least one a-beacon and one b-beacon from its neighboring active sensor within a round of n beacon intervals. Hearing beacons plays an important role in the proposed algorithm. On hearing beacons, sensors are aware of the statuses of nearby active sensors; they then can decide to keep active or inactive accordingly. Moreover, by the beacons heard, active sensors can figure out and record the offsets of their beacon interval starting times for delivering data properly later.

When an active sensor i has sensed data to send, it can send the intended recipient, say sensor j, a TI (traffic indication) message in j's TI window. On receiving a TI message, sensor j should acknowledge the message and then keep the radio on even if it is in the non-monitor interval. On receiving the acknowledgment, sensor i can then send the data to sensor j. In this way, active sensors can turn off the radio in radio-off windows to save energy. But when there are data to send/receive, sensors can just turn on the radio at the right time.

The beacon interval length l and the round size n (i.e., the number of beacon intervals in a round) are two adjustable parameters. The two parameters affect energy consumption and *neighbor sensibility*, the tendency for a sensor to detect a neighboring active sensor. Larger l's and larger n's lead to less energy consumption but worse neighbor sensibility. Actually, l indicates the minimum delay for a sensor to detect an active sensor by hearing beacons of the active sensor. Note that l is a system-wide parameter; that is, all sensors should have the same l value. On the other hand, every sensor can set its own value of parameter n. When a sensor decides to have higher neighbor sensibility, it can reduce its own n value so that it can hear beacons of an active sensor more frequently, leading to better neighbor sensibility.

Near Beacons and Far Beacons

In this section, we describe how to achieve connected coverage by as few active sensors as possible with the help of beacons heard. Initially, all sensors are inactive. Some sensors become active spontaneously after a random backoff time and start to broadcast beacons; others kept inactive if they are aware of other active sensors by hearing beacons. In beacon window of every beacon interval, an active sensor broadcasts a near beacon (a-beacon) and a far beacon (b-beacon) in a row with transmission ranges $\alpha\sqrt{3}Rs$ and $\sqrt{3}Rs$, respectively.

A spontaneously active sensor i performs a *descendant solicitation* procedure by immediately broadcasting a *solicitation message* after the beacon broadcast in every beacon interval. The procedure will continue until one or more replies to the solicitation message are received. It is noted that a solicitation message is embedded with the sender's "active time" and ID. On receiving a solicitation message from sensor i, an inactive sensor will reply to the message if it can hear i's b-beacon but no i's a-beacon. Even an active sensor j will reply to the message and becomes inactive if j can hear i's b-beacon but no i's a-beacon, and i's active time is larger than j's. On receiving a first reply from sensor j, sensor i sends sensor j a *descendant confirmation message* and stops the descendant solicitation procedure. After receiving the descendant confirmation message, sensor j inherits the

active time of sensor *i* and becomes active. Below, we describe how the active time inheritance is realized.

Formally, the *active time* T of a spontaneously active sensor *i* is a 3-tuple vector (Timer, Root, Level), where Timer, Level is initially set to 0, and Root is set to be the ID of the spontaneously active sensor *i*. When sensor *i* spontaneously becomes active, its timer starts ticking. The active time is embedded in both an a-beacon and a b-beacon. When sensor *j* inherits the active time of sensor *i*, it sets its own active time by copying *i*'s Timer and Root and by decreasing Level by 1 (the root is of level 0 and descendants are of levels -1, -2, etc.). For example, if sensor *j* inherits active time (123, *i*, 0) from sensor *i*, *j*'s active time will be (123, *i*, -1). After inheriting the active time from a spontaneously active sensor *i*, sensor *j* continues ticking the timer that is newly inherited and *j*'s active time can be further inherited by other sensors. We assume the timers of the giver (e.g., sensor *i*) and the inheritor (e.g., sensor *j*) can be synchronized very closely by considering message propagation and process delays, and any possible timer drift. Note that during active time inheritance, the Root component is kept intact so that the ID of the spontaneously active sensor can be preserved. If there are many active times to inherit, a sensor only inherits the oldest (or largest) active time. An active time T1 is regarded to be older (or larger) than active time T2 if T2 precedes T1 in the lexicographic order. For example, (456, *i*, -1) is older than (123, *k*, 0), and (456, *i*, -1) is older than (456, *i*, -2) by definition. Below we say that sensor *i* is older than sensor *j* if *i* has larger active time than *j*. Note that as we have mentioned, an older sensor influences a younger sensor, but not vice versa.

Whit the help of the active time, we design the algorithm in Fig. 5 for sensors to decide to stay active or inactive. Each sensor *j* maintains two sets: A and $\overline{\text{A}}$B. Note that A and $\overline{\text{A}}$B is calculated on the basis of a round of *n* beacon intervals.

Sensor *i*'s ID will be kept in set A of sensor *j* if the following conditions all hold:
(1) *j* can hear *i*'s a-beacon;
(2) *i* is older than *j*;
(3) *i* is the oldest among those whose a-beacons are heard by *j*.

For example, if sensor *j* with active time (234, *x*, -5) hears four a-beacons of active times (123, *y*, -2), (123, *y*, -3), (456, *z*, -1) and (456, *z*, 0), respectively, then only the ID of the last beacon's sender appears in set A. Note that only one ID will appear in set A.

The case of one sensor *i* in set A of sensor *j* implies that *i* and *j* are too close to conform to the optimal hexagonal deployment. Since sensor *i* is older than *j*, and *i* has the oldest active time, it keeps active and sensor *j* should enter (retain) the inactive state.

On the other hand, sensor *i*'s ID will be kept in set $\overline{\text{A}}$B of sensor *j* if the following conditions all hold:
(1) *j* can hear *i*'s b-beacon but no *i*'s a-beacon;
(2) *i* is larger than *j*;
(3) (*i* is the oldest among those whose b-beacons are heard by *j*) or (*i*'s active time has the same Root as the oldest one's).

```
VAR S; //variable for indicating sensor state
VAR T; //variable for storing active time
IF |A|=1 THEN S = inactive; Reset T;
ELSE IF |ĀB|≥2 THEN S = active; Adjust T;
ELSE Do-Nothing;
//Reset T is to reset active time T to be (0, 0, 0)
//Adjust T is to inherit active time from the oldest sensor
```

Figure 5. The algorithm for a sensor to decide to keep active or inactive

For example, if *j* with active time (234, *x*, -5) hears four b-beacons of active times (123, *y*, -2), (123, *y*, -3), (456, *z*, -1) and (456, *z*, 0), respectively, then *j* puts into set $\overline{\text{A}}$B the IDs of the last two beacon senders (we assume *j* does not hear a-beacons of the two senders). Note that the IDs of two or more sensors may appear in set $\overline{\text{A}}$B of sensor *j*; the sensors are the one (say *i*) with the oldest active time in the neighborhood of *j*, and the ones that have active times of the same Root as *i*.

Below, we describe the details of the algorithm in Fig. 5. When |A|=1, sensor *j* enters (or retains) the inactive state (S = inactive;) and resets the active time (Reset T;) to be (0, 0, 0). In such a case, there is one older sensor within $\alpha\sqrt{3}R_S$ distance from *j*, so sensor *j* should become inactive to make the network topology approach the optimal hexagonal deployment. Otherwise, when |$\overline{\text{A}}$B|≥2, sensor *j* enters (or retains) the active state (S = active;) and inherits the largest active time ever heard (Adjust T;). In such a case, there are at least two older sensors of the same Root within a distance between $\alpha\sqrt{3}R_S$ and $\sqrt{3}R_S$ from *j*. So, sensor *j* should become active to make the network topology approach the optimal hexagonal deployment. Note that before sensor *j* enters the active state from the inactive state, it must keep the radio on and wait for a random backoff time. Before the backoff time expires, if *j* receives any older sensor's a-beacon, sensor *j* will stay inactive; otherwise *j* will enter the active state and start sending a-beacons and b-beacons. In this way, no two nearby sensors within $\alpha\sqrt{3}R_S$ distance will go active simultaneously.

Since the algorithm in Fig. 5 takes only beacons with the oldest active time into consideration, the topology will be the one decided by the oldest spontaneously active sensor, which is called the *topology initiator*, and its first descendant. In the topology, any two active sensors are separated by distance between $\alpha\sqrt{3}R_S$ and $\sqrt{3}R_S$. When the density of sensors is sufficiently high, we can set α to be around 1 and the resulted topology will conform to the optimal hexagonal deployment.

4. Simulation Results

We develop a simulator in C language for demonstrating the performance of the proposed algorithm. The simulation experiments assumes 3000, 4000,…, or 7000 sensors randomly deployed in a 100 m × 100 m monitored area. The sensing radius (R_S) of each sensor is 10 meters, and the communicating radius (R_C) of each sensor is 20 meters ($R_C \geq 2R_S$). The default beacon interval length *l* is 100 ms, round size *n* is 10, and α is set to 0.7, 0.75, 0.8 or 0.85. Our simulator also considers the

details of MAC protocol and follows the IEEE 802.11 MAC standard [11].

We conduct simulation experiments for estimating the *coverage ratio* C and the *optimality ratio* R, where C = (the total area covered by active sensors / the entire monitored area) and R = (the number of active sensors / the number of sensors in the optimal hexagonal deployment). The first experiment is for 5000 sensors randomly placed in a 100 m × 100 m monitored area with α=0.8 (see Fig. 6). As shown in Fig. 7, our algorithm keeps only 51 sensors active to cover the area. According to [9], the optimal hexagon-based deployment needs 42 sensors to completely cover the 100 m × 100 m area. Our algorithm thus has an optimality ratio 1.21 (51/42). However, due to the *boundary effect* [10] that some uncovered regions exist near the boundary of the monitored area, the 51 sensors achieve a coverage ratio of only 93.31%. To ignore the boundary effect, we have a better coverage ratio of 97.35%. Fig. 8 shows the coverage ratio with/without the boundary effect for 3000, 4000,…, or 7000 sensors randomly deployed in a 100 m × 100 m area with α=0.8.

Below we discuss the effect of α on the coverage ratio and the optimality ratio. As shown earlier, the transmission ranges of a-beacons and b-beacons are $\alpha\sqrt{3}R_S$ and $\sqrt{3}R_S$, respectively, where $1 > \alpha > \frac{1}{\sqrt{3}}$. Only sensors hearing b-beacons and not hearing a-beacons of older active sensors can be candidates to be active sensors. Therefore, when α is smaller, there will be more candidates to be active. However, the number of active sensors will increase and the network topology deviates a lot from the optimal hexagonal deployment for smaller α values.

On the other hand, when α is larger, the sensor candidates to be active will be fewer. If the sensor density is too low, there may be no sensors to be candidates to be active, which may lead to larger uncovered area. So, we must properly adjust α according to the sensor density. When the sensor density is sufficiently high, we can set α to be about 1. In such a case, both the optimality ratio and the coverage ratio will approximate 1. Fig. 9 shows the performance for 3000, 4000,…, or 7000 sensors randomly deployed in a 100 m × 100 m area with α=0.7, 0.75, 0.8 and 0.85. By Fig. 9, we can see that α=0.8 seems to be a good setting, since it renders relatively high coverage ratios and low optimality ratios for most of the cases.

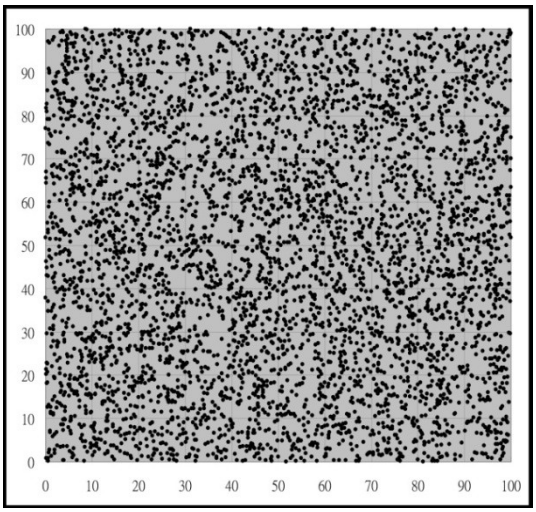

Figure 6. 5000 sensors randomly deployed in a 100 m × 100 m area

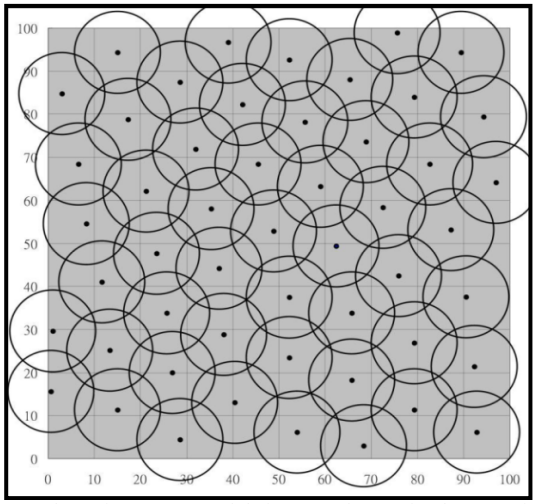

Figure 7. The active sensors and their covered areas for the scenario shown in Fig. 6

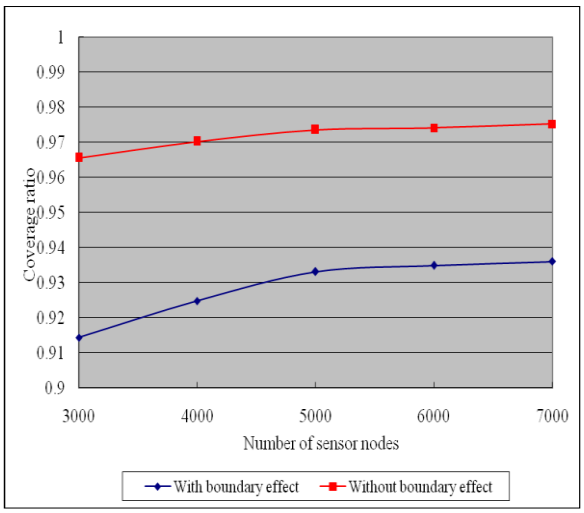

Figure 8. The coverage ratio of the proposed algorithm with/without the boundary effect

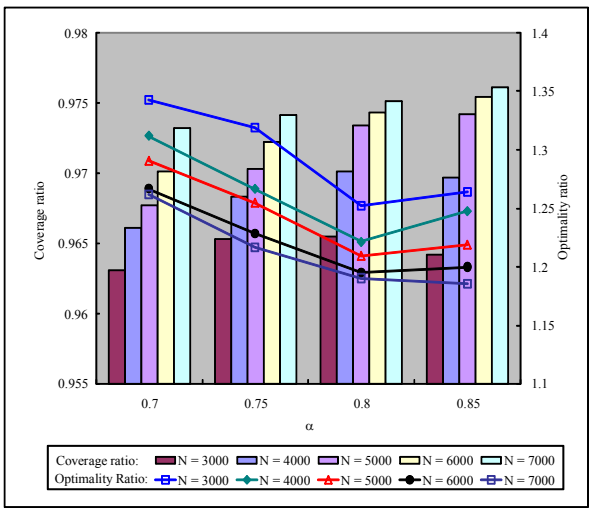

Figure 9. The impact of α on the coverage ratio and the optimality ratio

5. CONCLUDING REMARKS

Under the assumption of sufficiently high density of sensors and the assumption of $R_C \geq 2R_S$, we have proposed a location-free and range-free density control algorithm for wireless sensor networks, where R_C and R_S are sensors' radio radius and sensing radius, respectively. The algorithm tries to keep as few as possible sensors in the active state to approximate the optimal connected hexagonal deployment to cover a specific area of interest. The basic concept of the algorithm is to use two types of beacons, near beacons and far beacons, with different transmission ranges of $\alpha\sqrt{3}R_S$ and $\sqrt{3}R_S$, where α is a parameter and $1 > \alpha > \frac{1}{\sqrt{3}}$. An active sensor should turn on the sensing module and broadcast the two types of beacons periodically. However, an inactive sensor can turn off both the sensing module and the radio module to save energy for prolonging the network lifetime. Without synchronizing with active sensors, an inactive sensor just periodically turns on the radio module for a short period of time for receiving beacons to detect the statuses of neighboring active sensors.

By the asynchronous beaconing mechanism, sensors are ensured to hear beacons from neighboring active sensors within a round of beacon intervals. Sensors can thus decide to keep active or inactive based on the beacons heard or not heard. By setting the parameter α properly, the active sensors can approximate the optimal hexagonal deployment of sensors. We have performed simulation experiments for the proposed algorithm to investigate how close it can approximate the optimal hexagonal deployment.

The algorithm is fault-tolerant in the sense that when any active sensor runs out of energy or fails, one or more inactive sensors can switch to the active state to take over the surveillance responsibility. However, our algorithm makes the newly active sensors very close to the failing sensor, leading to the *load-unbalance problem*. That is, sensors near positions of the hexagonal deployment decided by the topology initiator have higher probability to be active and thus deplete energy more quickly. We can apply the concept of *epoch* to solve the problem as follows. When a sensor's timer exceeds a threshold, it can then reset its timer to start a new epoch after a random backoff time which is inversely proportional to the residual energy (or proportional to the number of times ever being active). The active time is now a 4-tuple vector: (Epoch, Timer, Root, Level). Sensors are initially in epoch 0 and every time a new epoch is started, the epoch is increased by one. A sensor starting a new epoch behaves like a spontaneously active one, and should perform the descendant solicitation procedure. And the first sensor to start a new epoch becomes the new topology initiator. In this way, sensors have nearly equal chances to be active to share the surveillance load. The epoch concept also solves the timer overflow problem caused by the limited number of bits to represent timer. For example, a 32-bit timer can count from 0 to 2^{32}-1. If the timer is increased by one per millisecond, it takes about 49.7 days for the timer to reach the maximum value and overflow. Since our algorithm relies on the comparison of timers, it may work improperly when timers overflow. By the epoch concept, a sensor will reset its timer before the timer overflows, so there will be no timer overflow problem. Note that the epoch variable is unlikely to overflow, since a 16-bit epoch variable will overflow only after several thousands of years.

Unlike other algorithms, the proposed one does not rely on position information or ranging information of sensors. It is thus suitable for more environments, such as the Mars sensor network. However, if sensors can obtain their accurate position information, then the algorithm can even be simplified. Each sensor just embeds its position information into beacons and sends them periodically. On receiving a beacon, a sensor can calculate the distance between itself and the sender by the position information embedded. Thus, only one type of beacons suffices and the algorithm can approximate the optimal deployment more closely even when the beacon transmission ranges vary significantly due to environmental influence or signal interference.

ACKNOWLEDGMENT

This paper is partially supported by the National Science Council in Taiwan through Grant NSC 96-2221-E-008-007.

REFERENCES

[1] Ian Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, pp. 102-104, 2002.

[2] Xiaole Bai, Santosh Kumar, Ziqiu Yun, Dong Xuan, Ten H. Lai, "Deploying Wireless Sensors to Achieve Both Coverage and Connectivity," in *Proc. of the 7th International Symposium on Mobile Ad Hoc Networking and Computing*, 2006.

[3] Yang-Min Cheng and Li-Hsing Yen, "Range-Based Density Control for Wireless Sensor Networks," in *Proc. of the 4th Annual Communication Networks and Services Research Conference (CNSR 2006)*, pp. 170-177, 2006.

[4] Himanshu Gupta, Samir R. Das, Quinyi Gu, "Connect Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution," in *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 189-200, 2003.

[5] Xiaoyan Hong, M. Gerla, R. Bagrodia, P. Estabrook,T.J. Kwon, G. Pei, "The Mars Sensor Network: Efficient, Power Aware Communications," in *Proc. of IEEE MILCOM 2001*, McLean, VA, Oct. 2001.

[6] C.-F. Hsin and M. Liu, "Network Coverage Using Low Duty-cycled Sensors: Random and Coordinated Sleep Algorithms," in *Proc. of International Symposium on Information Processing in Sensor Networks*, pp. 433–442, 2004.

[7] Rajagopal Iyengar, Koushik Kar, Suman Banerjee, "Low-coordination Topologies for Redundancy in Sensor Networks," in *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 332-342, 2005.

[8] Jehn-Ruey Jiang, Yu-Chee Tseng, Chih-Shun Hsu and Ten-Hwang Lai, "Quorum-based Asynchronous Power-saving Protocols for IEEE 802.11 Ad Hoc Networks,"

ACM Journal on Mobile Networks and Applications, Vol. 10, Issue 1-2, pp. 169–181, 2005.

[9] Mohamed K. Watfa, Sesh Commuri, "Optimal 3-Dimensional Sensor deployment Strategy," in *Proc. of Consumer Communications and Networking Conference*, 2006.

[10] R. Kershner, "The Number of Circles Covering A Set," A*merican Journal of Mathematics*, pp. 665-671, 1939

[11] LAN MAN Standards Committee of the IEEE Computer Society, *IEEE Std 802.11-1999: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, IEEE, 1999.

[12] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, "Wireless Sensor Networks for Habitat Monitoring," in *Proc. of the 1st ACM International Workshop on Wireless Workshop in Wireless Sensor Networks and Applications* (WSNA 2002), August 2002.

[13] W. T. Wang, K. F. Ssu, H. C. Jiau, "Density Control without Location Information in Wireless Sensor Networks," in *Proc. of the International Conference on Wireless and Mobile Communications*, 2006.

[14] Rong-Hou Wu, Yang-Han Lee, Hsien-Wei Tseng, Yih-Guang Jan and Ming-Hsueh Chuang, "Study of Characteristics of RSSI Signal," in *Proc. of IEEE International Conference on Industrial Technology (ICIT 2008)*, 2008.

[15] Fan Ye, G. Zhong, J. Cheng, Songwu Lu and Lixia Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks," in *Proc. of International Conference on Distributed Computing Systems*, pp. 28-37, May 2003.

[16] L.-H. Yen, C. W. Yu, and Y.-M. Cheng, "Expected *K*-Coverage in Wireless Sensor Networks," *Ad Hoc Networks*, vol. 5, no. 4, pp. 636-650, 2006.

[17] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Journal on Wireless Ad Hoc and Sensor Networks*, vol. 1, pp. 89-123, Jan 2005.

[18] Zongheng Zhou, Samir Das, and Himanshu Gupta, "Connected *K*-Coverage Problem in Sensor Networks," *IEEE Computer Communications and Networks*, pp. 373-378, 2004.

Jehn-Ruey Jiang was born in Taiwan in 1965. He received his PhD degree in Computer Science in 1995 from National Tsing-Hua University, Taiwan, R.O.C. He joined Chung-Yuan Christian University as an Associate Professor in 1995. He joined Hsuan-Chuang University in 1998 and became a full Professor in 2004. He is currently with the Department of Computer Science and Information Engineering, National Central University, Taiwan. He was a recipient of Best Paper Award of the 32nd International Conference on Parallel Processing, 2003, and has served as the editors of Journal of Information Science and Engineering and International Journal of Ad Hoc and Ubiquitous Computing in 2004 and 2005, respectively. He has organized the 1st, 2nd and 3rd International Workshop on Peer-to-Peer Networked Virtual Environments in 2007, 2008 and 2009, respectively. His research interests include distributed algorithms for peer-to-peer networks, mobile ad hoc networks and wireless sensor networks. He is a member of ACM and IEEE.

Tzu-Ming Sung was born in Taiwan in 1983. He received his master degree in Computer Science and Information Engineering in 2007 from National Central University, Taiwan, R.O.C. His research interests include mobile ad hoc networks and wireless sensor networks.