Coverage-aware Sleep Scheduling for High Density Cluster-based Sensor Networks

Fangyang Shen
Department of CSSE
Auburn University
Auburn, AL, 36830

Min-Te Sun
Department of CSSE
Auburn University
Auburn, AL, 36830

Chunlei Liu
Department of Math&CS
Valdosta State University
Valdosta, GA, 31698

Abstract

In cluster-based sensor networks, part of the sensor nodes can be switched into sleep state in order to conserve energy if their neighbors can provide the same or almost the same sensing coverage. However, as the number of nodes in sleep state increases, coverage for the cluster is degraded. It is crucial to maintain high coverage of clusters in order to preserve performance. In this work, we propose a coverage-aware sleep scheduling (CS) algorithm to improve the coverage of each cluster. Compared with two previous schemes: the randomized scheduling (RS) scheme and the distance-based scheduling (DS) scheme, the CS algorithm maintains higher coverage, while guaranteeing the same lifetime for the cluster. The CS algorithm thus improves the overall performance of the cluster-based sensor networks.

I. Introduction

Wireless sensor networks are network systems composed of small and inexpensive devices, deployed in a region to provide monitoring or communication capabilities for commercial or military applications. Typical applications include asset tracking and habitat monitoring [1], among many others.

The lifetime of a sensor network is measured by the time at which the network can provide a reasonable detection ratio or before most of the nodes have exhausted their battery power. Due to the vulnerable nature of individual sensors, wireless sensor networks are made up of many sensors deployed at a high density in order to increase the lifetime of the network systems. Since each sensor is battery powered, energy conservation is important to prolong network lifetime. In most of the sensor network systems (e.g., IEEE 802.15.4 ZigBee [2]), each sensor node operates in one of two states: active state and sleep state. In active state, a node is either actively transmitting or receiving data, or resting in an idle state. In sleep state, a node does not take part in most of the activities, and therefore it consumes much less energy. In a specific scenario of high density networks with energy constrained sensors, it is possible to selectively turn off some nodes rather than have all nodes active all the time. Sensor nodes can switch to sleep state if they are in a situation where their neighbors can provide the same or similar sensing coverage. Density control, which controls the density of active sensors at a desired level for a given area and controls sensor deployment, can ensure that a sufficient number of the sensor nodes remain active to maintain a high coverage level for the area where the sensor network is deployed. It is thus possible to achieve a balance between high coverage and a longer lifetime.

Many sensor systems are formed by a number of clusters [3]. In each cluster, a cluster head is elected to schedule the activities in the cluster, aggregate the sensing data, and communicate with neighboring clusters. In general, the sensing range for each node is smaller than its communication range. The goal for this paper is to develop a sleep scheduling algorithm to maximize the coverage of a cluster-based sensor network while at the same time putting a fixed percentage of the sensor nodes into sleep state in order to maintain a reasonable lifetime for the network. Specifically, we propose the coverage-aware sleep scheduling (CS) algorithm. The CS algorithm is based on the consideration of the overlap of sensor nodes' sensing areas. Nodes whose sensing coverage largely overlaps those of their neighbors are assigned a higher probability of being in sleep state in each cycle, while sensor nodes that have less overlap are assigned a lower probability of being in sleep state. Simulation results show that the CS algorithm maintains a higher coverage than schemes based on either randomized scheduling (RS) or distance-based scheduling (DS) algorithms [4], while at the same time maintaining almost the same lifetime of the cluster. The CS algorithm thus achieves a much better performance in terms of coverage and lifetime for cluster-based sensor networks. Simulation results show that the CS algorithm is very adaptable and can be applied to sensor networks with heterogeneous nodes (i.e., nodes with different sensing ranges and different initial energy), which was not considered in any previous studies.

The rest of this paper is organized as follows: Section 2 reviews the literature for this topic; Section 3 introduces the coverage-aware sleep algorithm in cluster-based sensor networks; Section 4 presents the simulation results; and Section 5 summarizes the findings of this study and discusses possible directions for future research.

II. Literature Review

The lifetime of a network is the time span from its initial deployment to the instant when it is deemed nonfunctional [5]. It can be defined as the instant when the first sensor dies, a certain percentage of sensors die, or there is an overall loss of coverage. There has been considerable research on network lifetimes of wireless sensor networks [5]–[7]. Chen

and Zhao [5] proposed a general formula for the lifetime of wireless sensor networks by identifying two key parameters at the physical layer that affected the network lifetime: namely the channel state and the residual energy of sensors. Rai and Mahapatra [6] designed a mathematical model to describe the lifetime of a sensor network when the data generation events within the network were spatially and temporally independent. Based on that model, they introduced an efficient routing strategy to achieve the optimal lifetime. Ritter and Voigt [7] presented a method for experimental lifetime measurement of sensor networks that made it possible to validate lifetime models within a reasonable amount of time. All of these models and methods can accurately measure the lifetime of wireless sensor networks, but they are inadequate to quantify the coverage of wireless sensor networks.

The coverage problem in wireless sensor networks has been the subject of increasing attention in recent years [8], [9]. Meguerdichian et al. [8] proposed an optimal polynomial time algorithm that used graph theory and computational geometry for solving the best and worst case coverage. Ahmed et al. [9] proposed a distributed probabilistic coverage algorithm to evaluate the degree of confidence in detection probability provided by a randomly deployed sensor network, using a uniform circular disc for sensing coverage in the binary detection model. Although most of these methods attempt to address and solve the coverage problems in different scenarios, few consider the impact on network lifetime. Furthermore, the above approaches can not be applied to the scenarios with density control in wireless sensor networks.

There have been some reports of studies that focus on large scale wireless sensor networks [10]–[12]. Zhang and Hou [10] investigated the relationship between coverage and connectivity in large sensor networks. Liu and Towsley [11] studied the issues affecting the coverage and detectability of a 2-dimensional finite-width strip sensor network probability algorithm, which was used to find a path between two random locations without being detected. Then the paper went on to characterize the asymptotic behaviors of the coverage and detectability of large scale sensor networks. Ye et al. [12] studied the issue of sink mobility in large-scale sensor networks and proposed a two-tier data dissemination approach that provided scalable and efficient data delivery to multiple mobile sinks. These studies integrate large wireless sensor networks, but none of the techniques address the combination of coverage and lifetime problems in wireless sensor networks.

Deng et al. [4] studied the node sleep scheduling problem in the context of cluster-based sensor networks. In this paper, the traditional sleep scheduling scheme, randomized sleep scheduling(RS) scheme, is described and used to schedule the node sleeping problem. In the RS scheme, all the nodes are assigned with a random probability to sleep in each cycle. Then the author proposed the linear distance based scheduling (DS) scheme to put the sensor nodes to sleep in each cycle. The DS scheme selects sensor nodes to sleep with higher probability if they are farther away from the cluster head. In contrast, the overall performance of the DS scheme is better than that of RS. Although DS offers a longer lifetime than RS, its coverage is not high. In general, high coverage is vital for wireless sensor clusters. Even though the lifetime is also significant, coverage of a cluster cannot be sacrificed solely in order to achieve a longer lifetime.

TABLE I
TERMS

R: the maximum transmission range of a cluster head;
P: the set of nodes in the cluster;
β_s: the percentage of nodes to sleep in each cycle;
β_d: the percentage of sensor nodes that run out of energy;
T(β_d): the time when β_d percent of sensors run out of energy;
p(k): the probability of a sensor node k to put to sleep;
N: the number of test points in the cluster;
D(i, j): the distance between two points i and j;
r: the transmission range of each sensor node;
$I_k(i)$**:** the indicator function between an active node k and a point i;
O(i): the overlap of all nodes at point i;
Z(k): the coverage degree of node k;
C(k): the summation of the overlap degree of all points in Z(k);
θ**:** the parameter to adjust the value of p(k) of node k;
$E_{active}(x)$**:** the average energy consumption per second of each active node;
λ**:** the average packet transmission rate per second for each sensor node sending data to cluster head;
k_1**:** the constant on energy consumption due to transmission of each packet;
k_2**:** the idle/receive energy consumption per second;
x_{min}**:** the minimum transmission range for the minimum allowable transmission energy;
γ**:** the path loss exponent;

III. COVERAGE-AWARE SLEEP SCHEDULING ALGORITHM

This section introduces the CS algorithm. The related terms are defined in Table I. The CS algorithm is designed to solve the following problem: how can a cluster head schedule nodes in the cluster to sleep so that the cluster can still achieve high coverage and maintain the longest possible lifetime? The problem is formulated by utilizing the following scenario. In a static cluster, the maximum transmission range of the cluster head is **R** and the set of sensor nodes in the cluster is **P**. Let β_s represent the percentage of sensor nodes in sleep state in each cycle. The lifetime of a cluster T(β_d) is defined as the time when β_d percentage of sensor nodes run out of energy. In order to maximize the sensing coverage and maintain a reasonable lifetime for the whole cluster, how does the cluster head select exactly β_s percentage of sensor nodes to sleep in each cycle?

The underlying concept governing the CS algorithm is that the greater the overlap of a sensor node k's sensing coverage with its neighbors, the higher the probability $p(k)$ that a sensor node can be placed in sleep state without affecting the overall network performance. In each cycle, the probability that each node should be in a sleep state can be calculated based on its overlap with its neighbors, then β_d percentage of nodes is set to sleep state and is determined based on these probabilities.

Here, we apply the CS algorithm with the following assumptions. First, the cluster maintains a static structure. Each

sensor node will stay in the same cluster for the duration of its lifetime. Second, the cluster head is located in the center of the cluster. Third, a cluster head knows the location and sensing range of each node in its cluster. Last, a two-dimensional Poisson Process is used to control the distribution of the nodes in a cluster.

The cluster spans a circle area of πR^2 and each sensor node covers a small area of πr^2 if their sensing range is r. To estimate the coverage of the cluster, we use N test points in the cluster circle area. In general, the higher density of points in a cluster, the more accuracy to estimate the coverage of a cluster. If the distance of two point i and j is denoted as $D(i, j)$, the indicator function $I_k(i)$ represents whether an active node k covers a point i, which is defined in Equation 1.

$$I_k(i) = \begin{cases} 1, & \text{if } D(i,k) \leq r \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Then the overlap degree of a test point i, $O(i)$, is the number of sensor nodes that cover point i, which can be obtained from Equation 2.

$$O(i) = \sum_{k \in \mathbf{P}} I_k(i) \tag{2}$$

Note that some test points may be outside of a cluster (i.e, away from the cluster head for more than the maximum transmission range of the cluster head) but are covered by sensor nodes in the cluster. Therefore, before we estimate the degree of overlap for a sensing range by these test points, we need to accurately estimate the actual points within the sensing range of each sensor which are actually inside the cluster.

A cluster head indexes each test point inside its cluster with a unique integer ranging from 1 to N. The sensing range of sensor node k covers a number of test points. Let us denote those situated inside the cluster (i.e., indexed by the cluster head of node k) as $Z(k)$; the coverage degree of node k, $C(k)$, is defined as the summation of the overlap degree of all points in $Z(k)$. Figure 1 illustrates the test points in $Z(k)$ by small circles, which have been indexed by the cluster head of node k.

For each sensor node k inside a cluster, there exists a corresponding C(k). For all sensor nodes in a cluster with cluster head h, let min(C(k)) be $\min_{D(k,h) \leq R} C(k)$. And max(C(k)) be $\max_{D(k,h) \leq R} C(k)$. The probability of node k to be in sleep state, $p(k)$, can be computed from C(k) using Equation 3.

$$p(k) = \frac{\theta \cdot C(k)}{MAX(C(k)) - MIN(C(k))} \quad 1 \leq k \leq P; \tag{3}$$

In Equation 3, the value of θ is a constant, which will be set appropriately so that the number of nodes to sleep reaches the expected level of β_s percent. In other words, θ is set so that Equation 4 is satisfied.

$$\sum_{k \in \mathbf{P}} p(k) = \beta_s \cdot \|\mathbf{P}\| \tag{4}$$

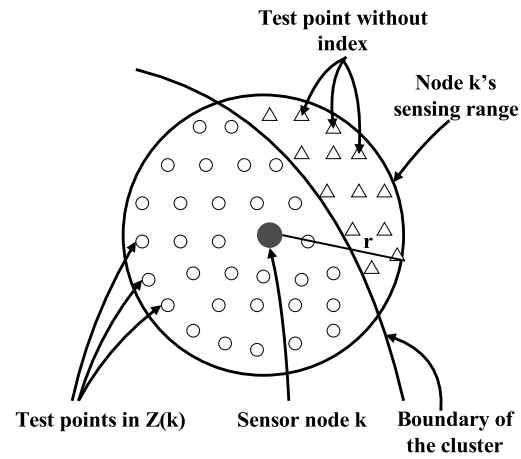

Fig. 1. Test points in the sensing range of node k

As we mentioned before, the lifetime of a cluster is defined as T(β_d), the time when β_d portion of sensors runs out of energy. To calculate the lifetime, we need to know the average energy consumption of each sensor node. According to [4], assuming the cluster head is h, the average energy consumption per second of an active node x for sensing and data communications can be obtained from equation 5.

$$E_{active}(x) = \lambda * k_1 * [max(x_{min}, D(x,h))]^{\gamma} + k_2 \tag{5}$$

It is important to note that this approach does not guarantee that a sensor node will be assigned to sleep state even if its sensing range is fully covered by its neighbors. However, the CS algorithm addresses the following two key issues: First, the more often a sensor's sensing range is covered by its neighbors, the higher probability of that sensor being assigned to sleep. Second, the greater the percentage of a sensor's sensing range that is covered by its neighbors, the higher probability that that sensor will be assigned to sleep. This strategy is not directly related to the distance between a sensor node and its cluster head. A major advantage of this approach is that it can be applied to sensor nodes with different sensing ranges, which is not addressed by either the RS or the DS scheme. The CS algorithm also maintains a higher level of coverage than either of the other two approaches.

The time complexity of the RS scheme is $O(1)$ and that of DS is ($\|\mathbf{P}\|$). On the other hand, the time complexity of the CS algorithm is in proportion to the product of N and $\|\mathbf{P}\|$. To achieve better coverage estimation accuracy, the value of N is normally set to be large, so the running time of the CS algorithm is longer than that of RS and DS. However, there is no need for a cluster head to coordinate with other clusters when conducting the sleep scheduling in its cluster using the CS algorithm as all the information needed for the computation can be obtained locally.

IV. Simulation Results

A. Simulation Settings

To determine the strength of the CS algorithm, we simulated the coverage and lifetime for cluster-based sensor networks of the CS algorithm using Matlab and compared the results with those obtained by the RS and DS algorithms. To assess the coverage of a cluster, the three algorithms were tested under multiple situations, including assigning different percentages of nodes to sleep state, maintaining nodes with different sensing ranges, and changing the number of nodes in the cluster. Simulation results in terms of lifetimes were shown for different situations, including the use of sensor nodes with different initial energies and sensor nodes with different sensing ranges. The default values used are as follows: $R = 100$ meters, minimum transmission range for each sensor = 5 meters, $\gamma = 2$, average total number of nodes is set to be 500, $k_1 = 0.000001J$, $k_2 = 0.1$ J/sec, $\lambda = 100$ frame/sec. For DS scheme, α is set to be 1.0.

B. Coverage for Different Schemes

1) Impact of β_s on Coverage: The relationships between the percentage of nodes in a cluster assigned to sleep β_s and the coverage of the cluster for the different algorithms are shown in Figure 2. The results plotted in Figure 2 are obtained by using the default values. The percentage of β_s is set as 5, 10, 15, 20, 25, 30, 35, and 40. It is evident that the coverage of a cluster achieved by all three algorithms degrades as the percentage of sensor nodes in sleep state increases, which occurs because the sensing ranges of some sleeping nodes are not adequately covered by their neighbors. Once these nodes go into sleep, their sensing range will not be covered anymore, which reduces the total coverage for the cluster. Figure 2 reveals that the CS algorithm not only achieves the best performance in terms of coverage but also has a lower tendency to suffer from decreased coverage with increasing numbers of nodes in sleep state. This is because the sensing area of the CS algorithm takes into account overlapping coverage with neighbors, and so performance suffers less in terms of coverage. Based on the evidence presented in this figure, we can conclude that CS algorithm is effective to achieve good coverage for the cluster than either of the other two algorithms.

2) Impact of Number of Nodes on Coverage: Figure 3 and 4 show the correlation between the number of nodes and the coverage of a cluster. The simulation settings of these two figures again use the default values, except that Figure 3 uses $\beta_s = 0.10$ and Figure 4 sets $\beta_s = 0.30$. The values of R are set as 300, 350, 400, and 450. From these figures we can see that the coverage of the cluster gradually increases with an increase in the number of nodes for all three algorithms. One important observation is that the coverage achieved by the DS algorithm decreases sharply as the number of nodes in the cluster decreases. This is because the DS scheme shuts down nodes which are far away from the cluster head with higher probability, and the sensing ranges for these nodes are less likely to be covered by their neighbors. The RS scheme performs better because it selects nodes to put into sleep state with a random probability between 0 and 1, which results in a uniform distribution of sleeping nodes in the cluster. On the whole, the CS algorithm achieves the best performance because the CS algorithm puts those nodes with the highest overlap with their neighbors to sleep. Hence, even as the number of nodes steadily decreases, the coverage achieved does not become as bad, as the sensing areas of the sleeping nodes are covered by their active neighbors. This also means that the CS algorithm adapts better to changing numbers of nodes in a cluster and can achieve a better coverage using fewer nodes than either of the other two algorithms.

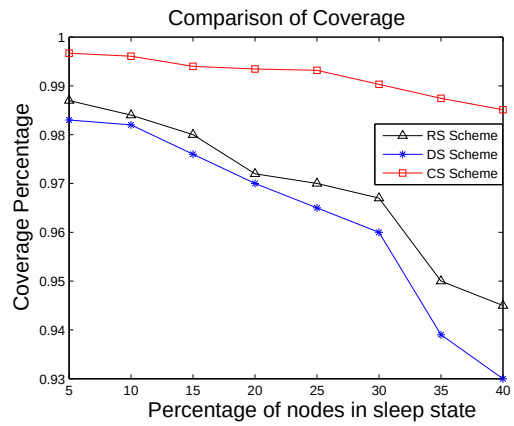

Fig. 2. Coverage for RS, DS, and CS Algorithms

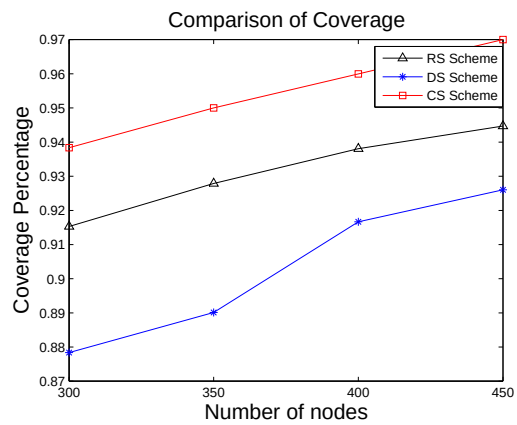

Fig. 3. Coverage for RS, DS, and CS Algorithms, $\beta_s = 0.10$

3) Impact of Sensing Range on Coverage: In order to evaluate the performance of different protocols for different sensing ranges, simulations were conducted using different sensing ranges for the sensor nodes, ranging from 6 to 14 meters. In each of the experiments, all the sensors had the same sensing range, for example, 6 meters or 8 meters. Other settings used were the default values. The results are presented in Figures 5 and 6. Figure 5 shows the coverage of the three algorithms for $\beta_s = 0.10$ and Figure 6 shows the coverage

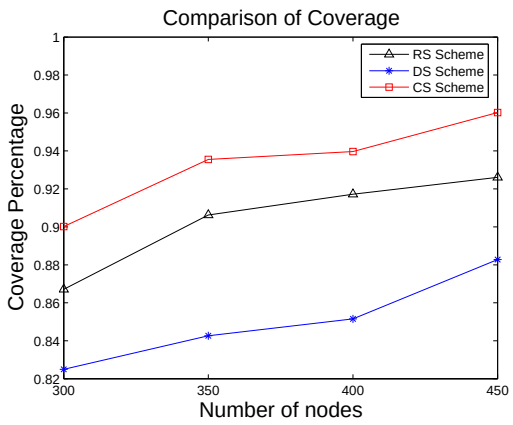

Fig. 4. Coverage for RS, DS, and CS Algorithms, $\beta_s = 0.30$

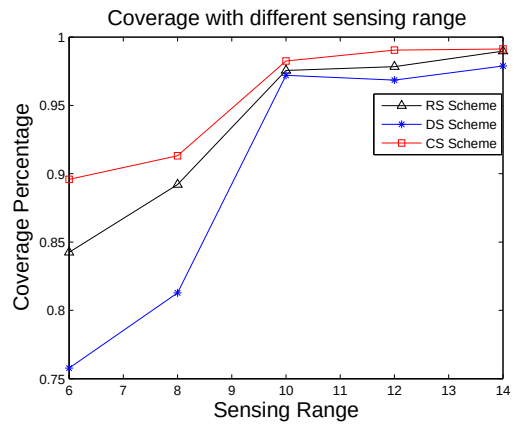

Fig. 6. Impact of sensing range, $\beta_s = 0.30$

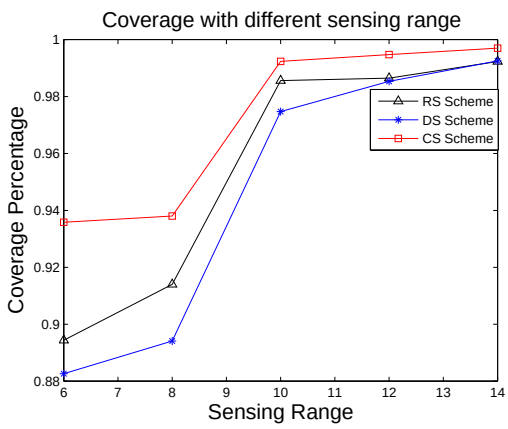

Fig. 5. Impact of sensing range, $\beta_s = 0.10$

of the three algorithms for $\beta_s = 0.30$. Figure 5 reveals that the performance of all three algorithms is poor when the sensing ranges for all the sensors are below 10 meters. This is because shutting down some portions of the sensor nodes will reduce the coverage when the sensing range for all the sensors is small. When sensing ranges are greater than 10 meters, the three protocols achieve almost the same performance, because nodes have relatively large overlaps, so turning off a small number of nodes will not significantly degrade the performance of sensor networks. Figure 6 reveals a similar situation, with the only difference being that the total coverage for all the protocols is reduced proportionally more as a higher percentage of nodes go to sleep. Based on the data presented in Figures 5 and 6, we can conclude that a small sensing range is not good for any of the sleep scheduling protocols and a large sensing range also contributes little to the cluster's overall coverage, so the optimum balance can be achieved when the sensing range is around 10 meters.

4) Impact of Random Sensing Range on Coverage: To further test the ability of the CS algorithm to achieve a high coverage for the cluster, we conducted experiments on sensor nodes with randomly assigned sensing ranges of 6 meters to 14 meters. This difference from the experiment described in the previous section is that here all the sensors are assigned a random sensing range, while previously all the sensors used the same sensing range in each experiment. The simulation settings were once again the default settings, and the performance of the three algorithms was compared by setting β_s to values in the range of $5\% \sim 40\%$. The simulation results are presented in Figure 7. As the figure shows, the coverage of all the protocols decreases with an increase in the number of sleeping nodes. When less than 25% of nodes are in sleep state, the RS and CS algorithms exhibit almost same performance, but when the portion of nodes in sleep state exceeds 25%, the CS algorithm performs slightly better than the RS algorithm. From the same graph, we can see that when β_s is larger than 15%, the coverage of the DS algorithm becomes rapidly worse, dropping sharply with a gradually increasing percentage of nodes going to sleep. These results indicate that both the RS scheme and the CS algorithm are suitable for sensor networks with heterogeneous sensor nodes, with the CS algorithm achieving a slightly higher coverage. The DS scheme performs poorly in situations where sensors have different sensing ranges. Overall, the CS algorithm is best suited for networks made up of heterogeneous sensors with different sensing ranges.

C. Lifetime for Different Schemes

1) Impact of β_s on Network Lifetime: The impact of β_s on network lifetime for the three algorithms is shown in Figures 8 and 9. The simulation settings are the default values. The percentage of β_s is set as 5, 10, 15, 20, 25, 30, 35, and 40. Figure 8 sets $\beta_d = 0.2$, and Figure 9 sets $\beta_d = 0.5$. From Figure 8, we can see that overall the DS scheme achieves the best network lifetime under all the settings of β_s. Although the RS scheme and CS scheme achieve almost the same network lifetime, both have about 10 percent shorter lifetime than that of the DS algorithm. This is because the DS scheme preferentially uses nodes that are close to the cluster head, which consume less energy than nodes that are farther away from the cluster head. From Figure 9, with the increase of β_d,

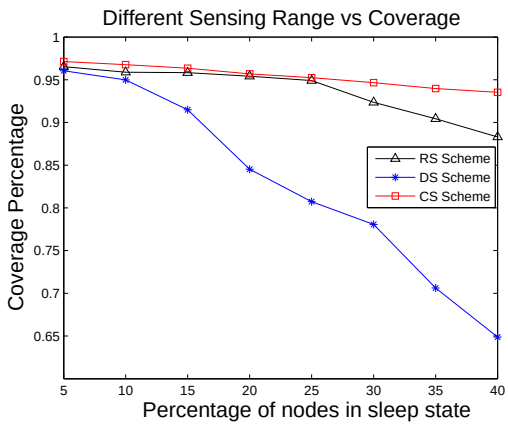

Fig. 7. Impact of random sensing range

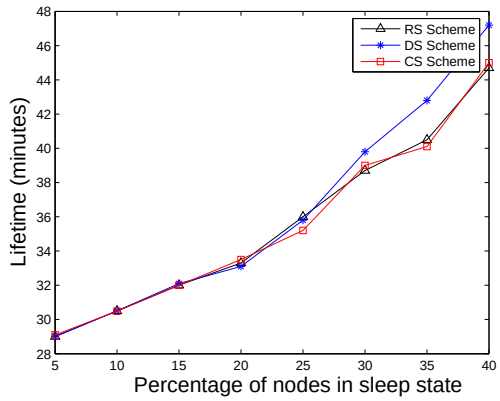

Fig. 9. Impact of β_s on lifetime, β_d=0.5

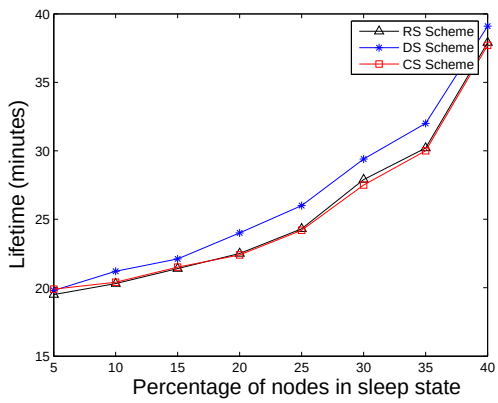

Fig. 8. Impact of β_s on lifetime, β_d=0.2

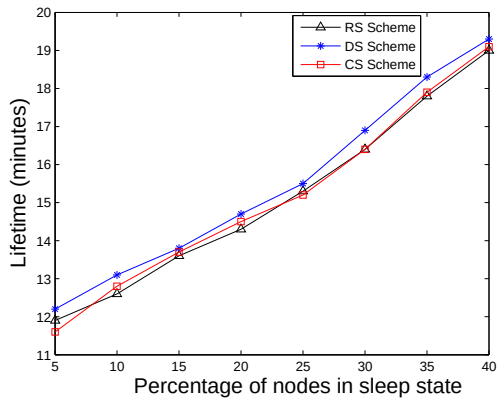

Fig. 10. Impact of different initial energy, β_d=0.2

and when there are less than 25% of nodes in sleep state, all three algorithms achieve almost the same lifetime. Above this point, the DS scheme performs slightly better than the other two, which means that the DS scheme is effective only when a high percentage of nodes are sleeping. If fewer than 25% of nodes are sleeping, and when the value of β_d is large, there is no significant difference between the three algorithms in terms of lifetime.

2) Impact of Different Initial Energy on Network Lifetime: To further compare the effectiveness of the three algorithms, we tested their lifetimes for sensor nodes with different initial energies, once again using the default values for these experiments. Each node was randomly assigned an initial energy ranging from $500J$ to 1000J. The percentage of β_s is set as 5, 10, 15, 20, 25, 30, 35 and 40. The results are presented in Figures 10 and 11. Figure 10 sets $\beta_d = 0.2$ and Figure 11 uses $\beta_d = 0.5$. As the figures show, all three algorithms result in almost the same network lifetimes for all the settings of β_s, no matter whether β_d is large or small. This implies that none of the three algorithms adapt well to situations in which the nodes have different initial energies. These results also indicate that the DS algorithm is particularly ineffective in achieving better network lifetimes when nodes have different initial energies. The possible reason is as follows. There are more nodes with high initial energies which are far away from the cluster head, and the DS scheme puts a disproportionally large number of nodes with high initial energies to sleep. However, the nodes with lower initial energies remain active for most of the cycles, thus they exhaust their batteries quickly. This does not affect either the RS scheme or the CS algorithm, as neither is sensitive to nodes with different initial energies.

V. Conclusions

Network coverage is of crucial importance for the operation of cluster-based sensor systems, and it is therefore necessary to give equal weighting to network lifetime and network coverage in cluster-based sensor networks. In this paper, we proposed the use of the coverage-aware sleep scheduling (CS) algorithm to improve coverage for the whole cluster with no adverse effect on lifetime. The fundamental concept governing the design of the CS algorithm is to assign the nodes with the highest sensing coverage overlap with their neighbors to be in sleep state with the highest probability in each cycle, while scheduling the sensor nodes with less overlap to remain active with higher probability.

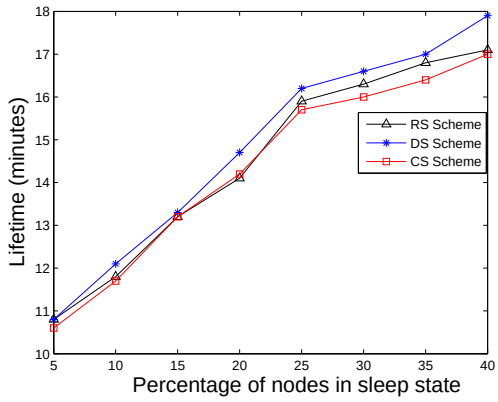

Fig. 11. Impact of different initial energy, β_d=0.5

The simulations revealed that the CS algorithm maintained higher coverage than either the RS scheme or the DS scheme, while guaranteeing the same lifetime for the whole cluster. Simulation results also showed that the CS algorithm adapted well to clusters containing sensor nodes with different sensing ranges and different initial energies.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] "IEEE 802.15. 4, ZigBee standard," http://www.zigbee.org/en/index.asp, 2004.

[3] O. Younis, S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Computing*, vol. 3, no. 4, pp. 366–379, Dec. 2004.

[4] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, "Scheduling Sleeping Nodes in High Density Cluster-based Sensor Networks," *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on "Energy Constraints and Lifetime Performance in Wireless Sensor Networks"*, vol. 10, no. 6, pp. 825–835, Dec. 2005.

[5] F. Chen and Q. Zhao, "On the Lifetime of Wireless Sensor Networks," *IEEE Communications Letters*, vol. 9, no. 11, pp. 976–978, Nov. 2005.

[6] V. Rai and R. N. Mahapatra, "Lifetime Modeling of a Sensor Network," *Proc. Conf. Design, Automation and Test in Europe*, vol. 1, pp.202–203, Mar.2005.

[7] H. Ritter, T. Voigt, "Experimental Evaluation of Lifetime Bounds for Wireless Sensor Networks," *Proc. Second European Workshop Wireless Sensor Networks*, pp. 25–32, Feb. 2005.

[8] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," *The 20th Conf. Computer Comm.*, pp. 1380–1387, 2001.

[9] N. Ahmed, S. S. Kanhere, S. Jha, "Probabilistic Coverage in Wireless Sensor Networks," *Proc. IEEE Conf. Local Computer Networks 30th Anniversary*, 2005.

[10] H. Zhang, J. Hou, "Maintaining Sensing Coverage and Connectivity in Sensor Networks," *Int'l Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, Feb. 2004.

[11] B. Liu, D. Towsley, "On the Coverage and Detectability of Wireless Sensor Networks," *Proc. of WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[12] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks," *The Annual Int'l Conf. Mobile Computing and Networking* , pp.148–159, 2002.