Scheduling Sleeping Nodes in High Density Cluster-based Sensor Networks

By: Jing Deng, Yunghsiang S. Han, Wendi B. Heinzelman, Pramod K. Varshney

Abstract:

In order to conserve battery power in very dense sensor networks, some sensor nodes may be put into the sleep state while other sensor nodes remain active for the sensing and communication tasks. In this paper, we study the node sleep scheduling problem in the context of clustered sensor networks. We propose and analyze the Linear Distance-based Scheduling (LDS) technique for sleeping in each cluster. The LDS scheme selects a sensor node to sleep with higher probability when it is farther away from the cluster head. We analyze the energy consumption, the sensing coverage property, and the network lifetime of the proposed LDS scheme. The performance of the LDS scheme is compared with that of the conventional Randomized Scheduling (RS) scheme. It is shown that the LDS scheme yields more energy savings while maintaining a similar sensing coverage as the RS scheme for sensor clusters. Therefore, the LDS scheme results in a longer network lifetime than the RS scheme.

Keywords: energy efficiency, sensor networks, cluster-based, sleep scheduling

Article:

1. Introduction

Recent technological advances have enabled the emergence of tiny, battery-powered sensors with limited on-board signal processing and wireless communication capabilities. Sensor networks may be deployed for a wide variety of applications [1]. A typical sensor network could contain thousands of small sensors, with the sensor density as high as 20 nodes/m^3. If these sensors are managed by the base station directly, communication overhead, management delay, and management complexity could make such a network less responsive and less energy efficient. Clustering has been proposed by researchers to group a number of sensors, usually within a geographic neighborhood, to form a cluster. Using a clustering approach, sensors can be managed locally by a cluster head, a node elected to manage the cluster and be responsible for communication between the cluster and the base station.

Clustering provides a convenient framework for resource management. It can support many important network features within a cluster, such as channel access for cluster members and power control, as well as between clusters, such as routing and code separation to avoid inter-cluster interference. Moreover, clustering distributes the management responsibility from the base station to the cluster heads. As pointed out by Varshney et al. [5,15] and Heinzelman et al. [8], such distributed management provides a convenient framework for data fusion, local decision making and local control, and energy savings. A fixed or adaptive approach may be used for cluster maintenance. In a fixed maintenance scheme, cluster membership does not change over time. In an adaptive clustering scheme, however, nodes may change their associations with different clusters.

The sleeping technique has been used to conserve energy of battery powered sensors. Rotating active and inactive sensors in the cluster, some of which provide redundant data, is one way that sensors can be

intelligently managed to extend network lifetime. Some researchers even suggest putting redundant sensor nodes into the network and allowing the extra sensors to sleep to extend network lifetime [16]. This approach is practical due to the low cost of individual sensors.

When a sensor node is put into the sleep state, it completely shuts itself down, leaving only one extremely low power timer on to wake itself up at a later time.[1] This leads to the following *Sleep Scheduling Problem*:

– How does the cluster head select which sensor nodes to put to sleep, without compromising the sensing coverage capabilities of the cluster?

In this paper, we study this problem and propose a sleeping scheme. Our analysis shows that our proposed sleeping scheme, called Linear Distance-based Scheduling (LDS), out-performs the conventional Randomized Scheduling (RS) scheme [17]. Our goal is to reduce energy consumption and extend the sensor network lifetime while maintaining adequate sensing coverage capabilities.

2. Related work

There has been some published work related to the cluster formation and cluster head selection problem [4,7–10]. In our work, we study the sleeping node selection problem by assuming that one of these clustering techniques is in use and the clusters and cluster heads are already in place.

Several schemes have been proposed in the literature to determine which nodes should be allowed to sleep. In [16], network nodes are allowed to go to sleep according to routing information and information from the application layer. This paper proposed the Basic Energy Conserving Algorithm (BECA) and the Adaptive Fidelity Energy-Conserving Algorithm (AFECA). In the BECA scheme, nodes switch among sleeping, idling, and active states to save energy. A node alternates between the sleep state and the idling state if no traffic is present. An idling node goes into the active state when it receives traffic from its application or from its neighbors. The AFECA scheme was designed to work with an on-demand routing protocol. In the AFECA scheme, the intervals between consecutive times that a sleeping node wakes up and listens to the channel are a multiple of the route discovery interval, at the end of which Route REQuest (RREQ) packets are transmitted.

Span was proposed in [2] to maximize the amount of time network nodes spend in the sleep state while maintaining the same traffic latency and network capacity. In Span, only a few nodes that are selected as Coordinators do not sleep. All other nodes go into the sleep state according to a sleep/wake cycle specified by the Coordinators. Only the Coordinators participate in packet routing. Significant energy saving was reported with the help of Span.

In [14], a node-scheduling scheme was proposed to reduce the overall system energy consumption by turning off some redundant nodes in sensor networks. The coverage-based off-duty eligibility rule and the backoff-based node-scheduling scheme guarantee that the original sensing coverage area is maintained even after nodes are turned off. According to these rules, sensor nodes can turn themselves off when they notice that their neighbors can cover all of their sensing coverage area. In order to avoid neighboring nodes turning off simultaneously, a back-off based approach was designed.

In the S-MAC scheme [17], energy consumption is reduced by allowing randomly-selected idle sensors to go into the sleep mode. The traffic that is sent to these sleeping nodes is temporarily stored at the neighboring active nodes. The sleeping sensors wake up periodically to retrieve the stored packets from their neighboring nodes.

In the Energy Dependent Participation (EDP) scheme [11], ad hoc network nodes decide whether to participate in ad hoc routing based on their residual energy. When the residual energy is high, a network node participates in routing with higher probability. This probability is lower when the residual energy is low. A balanced energy consumption is achieved and the extension of network lifetime was reported in the paper.

In [9], a role-based hierarchical self organization algorithm for wireless sensor networks was proposed. Three different roles for the sensors are categorized: sensing collaborator, sensing coordinator, and backbone nodes. Each sensor works as a sensing collaborator, which senses its neighborhood region for the targeted activity. Some sensor nodes may need to take the role of a sensing coordinator, which coordinates sensing activity and aggregates the sensing results. The backbone nodes, which are selected with the help of the Connected Dominated Set (CDS), are in charge of the transmission and relaying of sensing results to the base station. Reference [9] concentrated on sensing coverage but not energy consumption.

Some of the schemes discussed above, e.g., [9] and [2], require some knowledge of the entire network before a sensor node can decide to go to sleep. Other schemes such as [14,16], and [11] make decisions according to a specific system metric such as routing fidelity, sensing coverage, or residual energy. Schemes in [16] and [11] are not suitable for cluster-based sensor networks in which the goal is to improve energy saving while maintaining the same sensing coverage. The schemes proposed in [17] and [14] did not consider the variable transmission range of sensor nodes. In the following section, we propose a sleep scheduling scheme that exploits the variable transmission range of sensor nodes to save energy while maintaining the same sensing coverage in cluster-based sensor networks.

3. The sleep scheduling scheme

In this paper, we consider the sleep scheduling problem under several assumptions. The assumptions we make about the sensor network are:

- A large number of sensor nodes are deployed over a sensing field, such that at least some sensor nodes can be put into the sleep state without degrading the sensing coverage of the network;
- Sensor nodes form static clusters in the sensor network. Each sensor node belongs to the same cluster throughout its lifetime;
- The transmission range of each sensor node is variable, so that it can use the minimal transmission power that is necessary for communication with its cluster head. For convenience, we further assume the radio transceiver is capable of changing its transmission power in continuous steps;
- The distance between each node and the cluster head is known to the cluster head or to the node itself. Other nodes do not need to know this distance;[2]
- Nodes are randomly distributed as a two-dimensional Poisson point process with density ρ ρ. That is, the probability of finding n nodes in a region of area A is equal to $(\rho A)^n \exp(-\rho A)/n!$. Furthermore, these n nodes in the area are uniformly distributed .[3]

In order to maintain the sensing capability of the cluster, only a fraction of the sensor nodes can be put into the sleep mode. We explore the following problem on which fraction of the sensor nodes is allowed to sleep.

Sleep scheduling problem: In sensor networks, a sleeping technique can be used to save energy and extend network lifetime. How should a cluster head select nodes in the cluster to sleep so as to minimize energy consumption of the entire cluster while maintaining the required sensing coverage capability?

In this paper, we propose and analyze the following Linear Distance-based Scheduling (LDS) scheme: select nodes to sleep according to their relative distance to the cluster head. A sensor node that is farther away from the cluster head will be put into the sleep state with higher probability. Intuitively, energy savings can be improved by allowing these far-away nodes to sleep compared with allowing closer-to-center nodes to sleep. To the best of our knowledge, the only sleeping scheme proposed before, that is suitable for single-hop cluster-based sensor networks with variable transmission power, is the conventional Randomized Scheduling (RS) scheme, where randomly selected nodes in the cluster are put to sleep. Hence, the proposed scheme is only compared to the RS scheme. We argue that the LDS scheme is more energy-efficient than the RS scheme. We compare the performance of the RS scheme and that of the LDS scheme in the next section.

4. Performance analysis

In this section, we define the appropriate distance-based sleeping probability for the LDS scheme, and we analyze the performance, in terms of energy consumption, sensing coverage, and network lifetime, of the RS and LDS schemes.

4.1. Energy consumption

We focus on energy consumption at the cluster level. Assume that the maximum transmission range of the cluster head is R and there are n sensor nodes in the cluster. Furthermore, assume that the cluster head can reach all the sensor nodes in the cluster in one hop, and each sensor can communicate with the cluster head in one hop. Thus, the network topology of the cluster is star-shaped. The cluster covers a circular geographic area of πR^2 with the cluster head at the center. We assume that each node can change its transmission power (affecting the transmission range) such that it only uses the minimum transmission energy that is necessary for communication with the cluster head.

Assume the cluster head plans to allow, on an average, $n \cdot \beta_s$ nodes to sleep in each cycle. Note that the cluster head selects a set of nodes to put into the sleep state in each cycle, avoiding draining the energy of the active nodes. In the RS scheme, the cluster head selects each sensor node to sleep with probability $p = \beta_s < 1$.[4] In the LDS scheme, the probability of putting a node to sleep is not purely random. It is related to the distance, x, between the sensor node and the cluster head. This distance can be estimated by the sensor nodes by measuring the received signal strength from the cluster head's transmissions.

We only consider the energy consumption of the sensor nodes and not that of the cluster head.[5] Let λ be the average packet transmission rate per second of each sensor node sending data to the cluster head during its non-sleeping period that includes all data transmission periods and idle periods.[6] The energy consumption of an active node includes the required energy to transmit sensing results to the cluster head and the energy consumed when the node is idle. That is, the average energy consumption per second of the active nodes is

$$E_{\text{active}}(x) = \lambda \cdot k_1 \cdot [\max(x_{\min}, x)]^{\gamma} + k_2, \qquad (1)$$

where x is the distance between the sensor node and the cluster head, k_1 is the constant corresponding to energy consumption due to transmission of each packet, k_2 is the idle/receive energy consumption per second, $x_{\min}$ is the minimum transmission range corresponding to the minimum allowable transmission energy [6], and $\gamma \geq 2$ is the path loss exponent. The max function indicates that, even if the distance between a sensor node and the cluster head is smaller than $x_{\min}$, the sensor needs to spend the energy that is corresponding to $x_{\min}$ for its transmission.

4.1.1. Randomized scheduling (RS) scheme

In the RS scheme, sensors elect to sleep with probability $p = \beta_s < 1$. The implementation of this scheme is extremely simple. A sensor only needs to examine the output of a biased random number generator to determine whether or not it should go to sleep.

In this case, the expected energy consumption of each node during a unit time (second) is:

$$E_{\text{RS}} = \int_0^R (1 - \beta_s) \cdot f(x) \cdot E_{\text{active}}(x) \cdot dx, \qquad (2)$$

where $f(x)$ is the Probability Density Function (PDF) of the distance, x, between a sensor and the cluster head. Assuming that the sensor nodes are uniformly distributed in the circular coverage area of the cluster, $f(x)$ can be expressed as

$$f(x) = \frac{\partial F(X)}{\partial x} = \frac{\partial \left[\Pr(X \leq x) \right]}{\partial x} = \frac{\partial}{\partial x} \left[\frac{\pi x^2}{\pi R^2} \right] = \frac{2x}{R^2},$$

where $0 \leq x \leq R$. Therefore, E_{RS} is

$$E_{\text{RS}} = \frac{1 - \beta_s}{R^2} \left[\frac{\lambda k_1 \gamma}{\gamma + 2} (x_{\min})^{\gamma + 2} + \frac{2\lambda k_1}{\gamma + 2} R^{\gamma + 2} + k_2 R^2 \right].$$
$$(3)$$

We assume that the number of sensors is distributed according to a two-dimensional Poisson point process with expected density ρ, so we need to average E_{RS} over all possible numbers of nodes in a cluster:

$$E_{\text{RS}}^{(\text{overall})} = \sum_{n=0}^{\infty} n E_{\text{RS}} \frac{(\rho \pi R^2)^n}{n!} e^{-\rho \pi R^2}$$

$$= E_{\text{RS}} \sum_{n=0}^{\infty} n \frac{(\rho \pi R^2)^n}{n!} e^{-\rho \pi R^2}$$

$$= E_{\text{RS}} \cdot \rho \pi R^2. \qquad (4)$$

Therefore, the overall expected energy consumption of the RS scheme can be calculated as E_{RS} multiplied by the expected number of nodes in a cluster.

4.1.2. Linear distance-based scheduling (LDS) scheme

In the LDS scheme, nodes elect to sleep according to their distances to the cluster head. A sensor node that is farther away from the cluster head will go into the sleep state with higher probability. Intuitively, energy savings can be improved by allowing these far-away nodes to sleep for longer periods compared with allowing closer-to-center nodes to sleep.

In the LDS scheme, the probability with which a sensor node elects to sleep is linear in x. Let this probability be $p(x)$.[7] To keep our analysis tractable, we assume $p(x) = C f(x)$, where C is a constant and $f(x)$ is the PDF defined earlier. More precisely, we assume

$$p(x) = \begin{cases} C f(x) & \text{for } 0 \leq x \leq x_s \\ 1 & \text{for } x_s < x \leq R, \end{cases} \qquad (5)$$

where x_s will be determined later. $p(x)$ needs to satisfy

$$\int_0^R p(x) f(x) dx = \int_0^{x_s} C(f(x))^2 dx + \int_{x_s}^R f(x) dx = \beta_s.$$

The above condition allows $n \cdot \beta_s$ nodes to be asleep, on an average. Thus,

$$C = \frac{3R^4}{4x_s^3} \left(\beta_s - \frac{R^2 - x_s^2}{R^2} \right). \qquad (6)$$

C should be kept greater than or equal to 0 and we have

$$\beta_s - 1 + \frac{x_s^2}{R^2} \geq 0 \quad \text{or} \quad R\sqrt{1 - \beta_s} \leq x_s. \qquad (7)$$

Since $p(x) = C f(x)$ needs to be kept smaller than 1 when $x \leq x_s$, we have

$$\frac{3R^4}{4x_s^3} \left(\beta_s - \frac{R^2 - x_s^2}{R^2} \right) \frac{2x_s}{R^2} = \frac{3R^2}{2x_s^2} (\beta_s - 1) + \frac{3}{2} \leq 1.$$

Therefore,

$$x_s \leq R\sqrt{3(1 - \beta_s)}. \qquad (8)$$

The above result indicates that when the fraction of nodes going to sleep, β_s, is large, it is necessary for the LDS scheme to put some nodes to sleep with probability 1.[8] Combining Inequalities (7) and (8) and considering that $x_s \leq R$, the feasible range of x_s, is

$$R\sqrt{1 - \beta_s} \leq x_s \leq R \cdot \min(1, \sqrt{3(1 - \beta_s)}).$$

According to this form of $p(x)$, the energy consumption of the LDS scheme is

$$E_{\text{LDS}}(x_s) = \int_0^{x_s} [1 - C f(x)] \cdot f(x) \cdot E_{\text{active}}(x) dx. \qquad (9)$$

Further derivations of $E_{LDS}(x_s)$ are straightforward. We present the results here: When $x_{\min} < x_s$, $E_{LDS}(x_s)$ is

$$\frac{k_2}{R^2}(x_s)^2 - \frac{4C\lambda k_1}{(\gamma + 3)R^4}(x_s)^{\gamma+3} - \frac{4\gamma C\lambda k_1}{3(\gamma + 3)R^4}(x_{\min})^{\gamma+3}$$

$$+\frac{2\lambda k_1}{(\gamma+2)R^2}\cdot(x_s)^{\gamma+2}+\frac{\gamma\lambda k_1}{(\gamma+2)R^2}\cdot(x_{\min})^{\gamma+2}$$

$$-\frac{4Ck_2}{3R^4}\cdot(x_s)^3.$$

Otherwise, $E_{\text{LDS}}(x_s)$ becomes

$$E_{\text{LDS}}(x_s)=[\lambda k_1(x_{\min})^{\gamma}+k_2]\cdot\left[\frac{(x_s)^2}{R^2}-\frac{4C(x_s)^3}{3R^4}\right],$$

where C is given by (6). Similar to the derivation in (4), the overall expected energy consumption of the LDS scheme can be calculated as

$$E_{\text{LDS}}^{(\text{overall})}=E_{\text{LDS}}\cdot\rho\pi R^2.$$

The selection of x_s affects the energy saving of the LDS scheme. We select x_s in its feasible range from $R\sqrt{1-\beta_s}$, to $R\cdot\min(1,\sqrt{3(1-\beta_s)})$ as

$$R\sqrt{1-\beta_s}+\alpha\cdot\left[R\cdot\min(1,\sqrt{3(1-\beta_s)})-R\sqrt{1-\beta_s}\right].$$

Note that when $1\leq(1,\sqrt{3(1-\beta_s)})$.), i.e., $\beta_s<2/3$, x_s may be selected as R, which is the maximum feasible value and we choose $\alpha=1$. When x_s is chosen as R, $p(x)$ in (5) becomes

$$p(x)=Cf(x)<1 \text{ for all } 0\leq x\leq R$$

and

$$C=\frac{3R\beta_s}{4}. \qquad (10)$$

In this case,

$$E_{\text{LDS}}(x_s=R)=\frac{2\lambda k_1}{(\gamma+2)}\cdot R^{\gamma}+\frac{\gamma\lambda k_1}{(\gamma+2)R^2}\cdot(x_{\min})^{\gamma+2}$$

$$-\frac{3\beta_s\lambda k_1}{(\gamma+3)}R^{\gamma}-\frac{\gamma\beta_s\lambda k_1}{(\gamma+3)R^3}(x_{\min})^{\gamma+3}$$

$$+(1-\beta_s)k_2. \qquad (11)$$

Comparing (11) and (2), the energy saving of the LDS scheme over the RS scheme can be obtained as:

$$\frac{E_{\text{RS}}^{(\text{overall})}-E_{\text{LDS}}^{(\text{overall})}(x_s=R)}{E_{\text{RS}}^{(\text{overall})}}$$

$$=\frac{E_{\text{RS}}-E_{\text{LDS}}(x_s=R)}{E_{\text{RS}}},$$

which can be expressed as

$$\frac{\beta_s\lambda k_1 R^{\gamma}\cdot\frac{\gamma}{(\gamma+2)(\gamma+3)}-\frac{\beta_s\gamma\lambda k_1(x_{\min})^{\gamma+2}}{R^3}\cdot\left[\frac{R}{\gamma+2}-\frac{x_{\min}}{\gamma+3}\right]}{\frac{1-\beta_s}{R^2}\left[\frac{\lambda k_1\gamma}{\gamma+2}(x_{\min})^{\gamma+2}+\frac{2\lambda k_1}{\gamma+2}R^{\gamma+2}+k_2R^2\right]}.$$

An approximation can be applied. Therefore,

$$\frac{E_{\text{RS}}^{(\text{overall})}-E_{\text{LDS}}^{(\text{overall})}(x_s=R)}{E_{\text{RS}}^{(\text{overall})}}$$
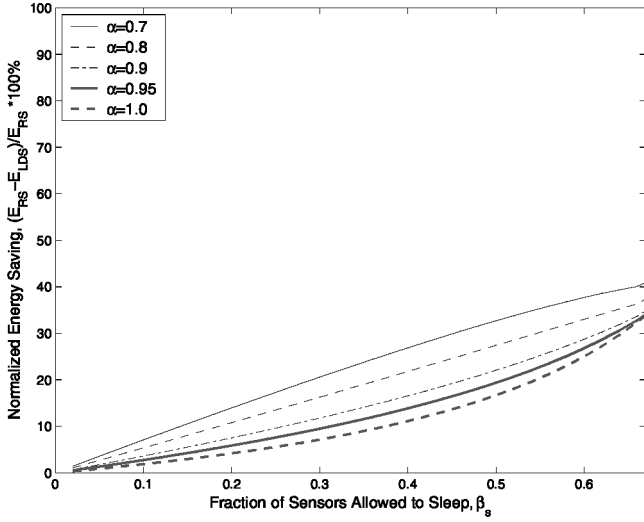
Figure 1. Normalized energy saving of the LDS scheme over the RS scheme ($\gamma = 2$).

$$\approx \begin{cases} 0 & \text{when } x_{\min} \approx R \\ \dfrac{\beta_s}{1 - \beta_s} \cdot \dfrac{\frac{\gamma}{(\gamma+2)(\gamma+3)}}{\frac{2}{\gamma+2} + \frac{k_2}{\lambda k_1} R^{-\gamma}} & \text{when } x_{\min} \ll R \end{cases} \quad . \quad (12)$$

Thus, when $x_{\min}$ is close to R, the energy consumption of the LDS scheme will be the same as that of the RS scheme. The reason for this is that every sensor node has the same transmission power consumption. When $x_{\min}$ is much smaller than R and we select x_s as R in (5), the energy consumption difference does not depend on the value of $x_{\min}$.

An interesting result conveyed by (12) is the energy savings of the LDS scheme over the RS scheme when the transmission energy cost is much larger than the non-transmission energy cost ($\lambda k_1 R^{\gamma} \gg k_2$) and when $x_{\min}$ is much smaller than R. When these two conditions are satisfied, we have

$$\frac{E_{RS}^{(overall)} - E_{LDS}^{(overall)}}{E_{RS}^{(overall)}} \approx \frac{\beta_s}{1 - \beta_s} \cdot \frac{\gamma}{2(\gamma + 3)}.$$

Therefore, the energy saving improvement of the LDS scheme over the RS scheme is about 20%, 25%, and 30% when path loss exponent y is 2, 3, and 4, respectively, and when $\beta_s = 0.5$.

In figure 1, we show the normalized energy saving of the LDS scheme over the RS scheme when $\gamma = 2$. In the sensor network that we studied, we assume that there are $n = 500$ sensors in each cluster, $k_1 = 10{-}6$ J/(frame $\cdot$ m^2), k$_2$ = 0.1 J/sec, $x_{\min} = 5$ m, and $\lambda = 100$ frame/sec. The maximum transmission range of the cluster head is $R = 100$m. Note that these are the set of system parameters we used to calculate the numerical results. Other similar parameters are possible. The number of sensors in each cluster might seem too large. However, with the low cost of sensor nodes and the need to extend network lifetime, putting more sensors than what is necessary is a viable option as suggested in [16]. From figure 1, we can conclude that the LDS scheme consumes less energy than the RS scheme.
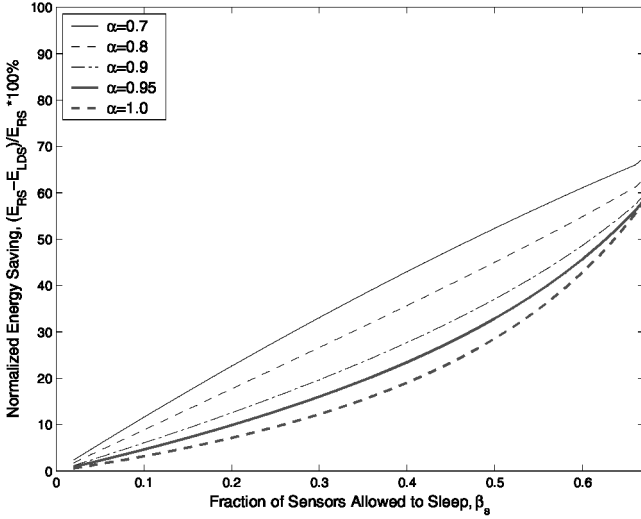
Figure 2. Normalized energy saving of the LDS scheme over the RS scheme ($\gamma = 4$).
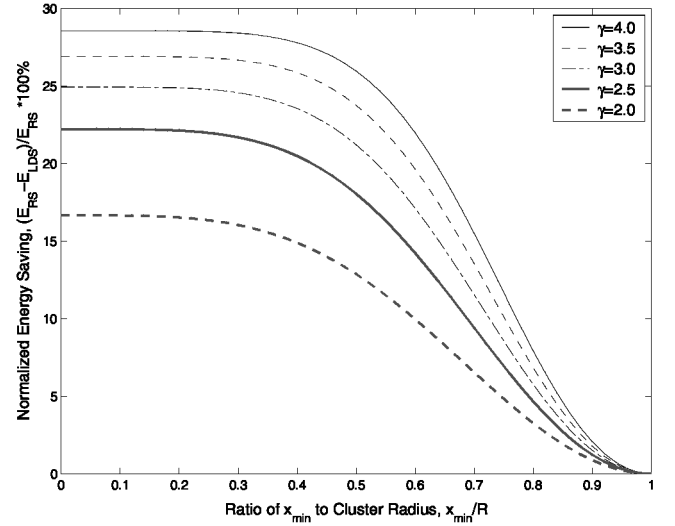
Figure 3. Normalized energy saving of the LDS scheme over the RS scheme for different γ.

We can observe from this figure that the LDS scheme results in much lower energy consumption compared to the RS scheme when a is small, i.e., when we choose x_s close to its minimum allowable value. This can be explained by the fact that when α is smaller, more nodes close to the border of the cluster are put to sleep with probability 1, as shown in (5). In fact, when $\alpha = 0$, all the sleeping nodes are located in the border ring of the circular cluster region. As we increase the value of α from 0 to 1, sensor nodes closer to the cluster head may be selected to sleep with higher probability. This energy saving comes with a possible sensing coverage loss around the border area of the cluster. Therefore, α is an important trade-off parameter between energy-saving and sensing coverage. It is also clear from the figure that the energy consumption of the LDS scheme is much lower compared to the RS scheme when the number of sleeping nodes allowed is larger.

Figure 2 compares the energy saving of the LDS scheme over the RS scheme when $\gamma = 4$ (the unit of k_1 is now J/(frame $\cdot$ m^4)). We can see that the performance gain of the LDS scheme over the RS scheme is even higher than the corresponding $\gamma = 2$ scenario. In this case, the LDS scheme (with $\alpha =1.0$) consumes 30% less energy than the RS scheme does when the fraction of sleeping nodes is about 0.5.

Note that the higher energy saving for small α values is due to the fact that the corresponding x_s values are smaller than R. Therefore, the sensor nodes that are located in the border area of the circular cluster region are always put to sleep. As noted previously, this may lead to loss of sensing coverage. However, figures 1 and 2 show that the energy saving of the LDS scheme is 17–28% higher than that of the RS scheme for $\alpha = 1.0$, in which case the sensor nodes in the border area are kept active with non-zero probability. We study the performance in terms of sensing coverage in Section 4.2.

In figure 3, we compare the energy savings of the LDS and the RS schemes for different x_{min} and path loss exponents, γ. The fraction of sensor nodes that are allowed to sleep is fixed at 0s = 0.5, and the parameter a is set to 1.0 for the LDS scheme. Therefore, x_s is equal to the cluster radius, R. It can be observed clearly from figure 3 that the performance gain of the LDS scheme over the RS scheme in terms of energy saving decreases with an increase in x_{min}. This is due to the increase in area that requires the same transmission power as x_{min}. When $x_{min}/R = 1$, i.e., x_{min} is equal to R, there is no difference between the LDS and the RS schemes in terms of energy saving. However, for a smaller x_{min}, the performance gain of the LDS scheme over the RS scheme remains roughly at the same level, 17 to 28%, depending on the value of the path loss exponent, γ. The larger γ is, the better the performance gain will be. This can be explained by the higher energy saving in the LDS scheme when γ is larger and sensor nodes farther away from the cluster head are selected to sleep with higher probability.

4.2. Sensing coverage
By putting the border nodes to sleep with a higher probability, the sensing coverage area of the cluster may be decreased.[9] One important performance study of our proposed sleep scheduling scheme is to calculate the

change in the sensing coverage area. This can be done analytically and by simulations, by assuming that the sensor's coverage area is a circle. We further assume that the sensing range of a sensor node (R_s) and the radio communication range (R) satisfy:

$$R_s < R. \qquad (13)$$

Since we employ a probabilistic sleep scheduling scheme, it is possible that all the nodes that can cover a location go to sleep. This location is then not covered by any active node. Assuming a location A that is at a distance x away from the center of the cluster, all nodes that may cover this location must reside in the circle with radius R_s, centered at A (see figure 4).

Note that we assume that the node density in a cluster is p and the sensors are distributed according to a two-dimensional Poisson distribution. Therefore, the probability that all nodes in the circle centered at location A with radius Rs (the shaded circle shown in figure 4) are sleeping, when there are altogether ns sensor nodes in the shaded circle, is:

$$p_0(x \mid n_s) = \prod_{Y \in S_A} P(Y)$$

$$= \exp \left\{ \int_{x-R_s}^{x+R_s} \log(p(y)) \cdot \rho_s \cdot y\phi \, dy \right\},$$

where S_A is the circular area centered at A with radius R_s, $P(Y)$ is the sleeping probability of node Y that resides in S_A, φ is the angle shown in figure 4, $p_s = \dfrac{n_s}{\pi R_3^2}$ is the node density in the circle, and $\gamma \, \varphi \, dy$ is the area of the arc ring. The expression for φ is

$$\phi = 2\cos^{-1}\left(\frac{x^2 + y^2 - R_s^2}{2yx}\right).$$

We define rc as the ratio of the area of regions that are covered by at least one sensor to the area of the entire circular cluster region. When there are n_s sensor nodes residing in the shaded circle, the value of rc can be estimated[10] as

$$rc(n_s) = 1 - \frac{1}{\pi R^2} \int_0^R \int_0^{2\pi} p_0(r \mid n_s) r \, dr d\theta$$

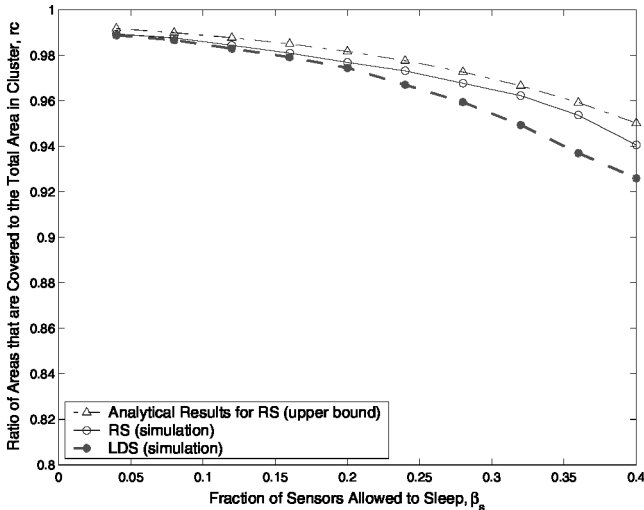$$= 1 - \frac{2}{R^2} \int_0^R p_0(r \mid n_s) r dr.$$

Figure 5. Comparison of the sensing coverage of the LDS and RS schemes.

Then, according to the Poisson distribution, the average ratio of areas that are covered by at least one active sensor to the area of the entire circular cluster region can be estimated as

$$rc^{(\text{overall})} = \sum_{n_s=0}^{\infty} (\rho\pi R^2)^{n_s} \exp(-\rho\pi R^2)/n_s! \cdot rc(n_s). \qquad (14)$$

In the LDS scheme, the probability of going to sleep for each sensor is dependent on the location of that sensor. Currently, no closed form of $rc^{(overall)}$ has been obtained for the LDS scheme due to the complicated form of (14); however, a closed form can be obtained in the RS scheme where the probability of going to sleep for each sensor is not dependent on the location of the sensor.

In the RS scheme, every sensor node goes to sleep with probability β_s, thus

$$p_0(x \mid n_s) = \beta_s^{n_s},$$

when there are n_s sensors in the shaded circle. We notice that $p_0(x \mid n_s)$ is not related to the distance, x, from the location of interest to the cluster head.[11] Therefore,

$$rc(n_s) = 1 - p_0\,(n_s) = 1 - \beta_s^{n_s},$$

when the RS scheme is employed. The average ratio of areas that are covered is now

$$rc^{(overall)} = \sum_{n_s=0}^{\infty} \left(\rho \pi R_s^2\right)^{n_s} \exp\left(-\rho \pi R_s^2\right)/n_s! \cdot rc(n_s)$$

$$= \sum_{n_s=0}^{\infty} \left(\rho \pi R_s^2\right)^{n_s} \exp\left(-\rho \pi R_s^2\right)/n_s! \cdot \left(1 - \beta_s^{n_s}\right)$$

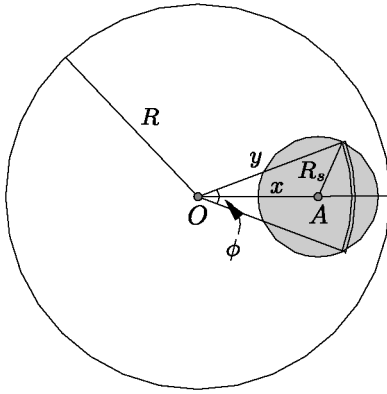$$= 1 - \exp\left\{-\rho \pi R_s^2(1 - \beta_s)\right\}. \qquad (15)$$

Figure 4. Calculation of location coverage.

Simulations were performed in Matlab to evaluate the sensing coverage of the LDS and the RS schemes.[12] Their performances in terms of sensing coverage are shown in figure 5. This figure shows the average ratio of areas in the circular cluster region that are covered by at least one sensor to the total circular cluster region, $rc^{(overall)}$, as a function of the fraction of sensors that are allowed to sleep, β_s. The average total number of sensor nodes in the cluster is 500. The radius of the cluster circle is 100 m. As expected, the values of $rc^{(overall)}$ decrease as β_s increases. We use $\alpha = 1.0$ for the LDS scheme. That is, no sensors go to sleep with probability 1. The $rc^{(overall)}$ value of the LDS scheme is slightly smaller than that of the RS scheme. This could be due to the higher probability of allowing sensor nodes that are far away from the cluster head go to sleep. Therefore, these border areas are not covered by any active sensors with slightly higher probability. Figure 5 also shows that our analytical upper bound on the sensing coverage of the RS scheme is tight.

Note that the similar sensing coverage performance of the RS and the LDS schemes does not mean that these two schemes cover the circular cluster region with the same coverage pattern. We expect the sensing coverage of the LDS scheme in the border area to be lower than that in the central area, as sensor nodes closer to the border are put to sleep with higher probability. The RS scheme, however, should result in a more uniform coverage pattern across the entire circular cluster region. Further performance results and the related discussions can be found in [3].

4.3. Network lifetime
We define the network lifetime $T(\beta_d)$ as the time when a fraction of sensors, β_d, run out of energy. Let Ψ be the total battery energy each sensor node carries when the sensor network is initialized.

In the RS scheme, the sensor nodes farther away from the cluster head consume much more energy than the sensor nodes that are closer to the cluster head due to (1). Therefore, the outer sensor nodes will run out of battery much faster than the inner sensor nodes. The time when β_d fraction of nodes run out of battery is the time when sensor nodes with $x \geq x_d^{(RS)}$ all run out of battery, where $x_d^{(RS)}$ satisfies:

$$\beta_d = \int_{x_d^{(RS)}}^{R} f(x)dx = \frac{R^2 - \left[x_d^{(RS)}\right]^2}{R^2}$$

leading to $x_d^{(RS)} = R \cdot \sqrt{1 - \beta_d}$.

The network lifetime of the RS scheme is then

$$T_{RS}(\beta_d) = \frac{\Psi}{E_{RS}\left(x_d^{(RS)}\right)}$$
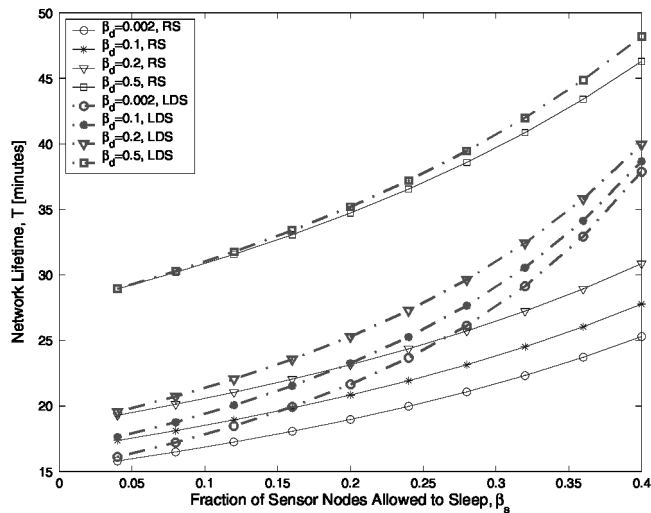
which can be expressed as

$$\frac{\Psi}{(1 - \beta_s)\{\lambda k_1 [\max(R \cdot \sqrt{1 - \beta_d}, x_{\min})]^\gamma + k_2\}}.$$

The analysis of network lifetime of the LDS scheme is given in Appendix A, where we only show the cases used to obtain our numerical results. Note that the network lifetime of the LDS scheme can also be calculated numerically in the following way: from (1) and (5), the energy consumption of all sensor nodes can be calculated based on their distance from the cluster head. We then find a β_d fraction of sensor nodes that run out of battery sooner than the rest of $1 - \beta_d$ fraction of sensor nodes. The time when the last of these 0d fraction of sensor nodes runs out of battery represents the network lifetime, $T_{LDS}(\beta_d)$.

We show the numerical results of the network lifetimes of the RS and the LDS schemes in figure 6. In the calculations, we assume $\Psi = 103J$.[13] The network lifetimes of both schemes improve as β_s increases, due to increasing energy saving in the sensor network. However, the network lifetime of the RS scheme is shorter than that of the LDS scheme. When β_s increases, the performance gain of the LDS scheme is better. For example, when $\beta_s = 0.4$, the time when 20% of sensors in the entire network run out of battery is 30% longer in the LDS scheme than that in the RS scheme.

5. Conclusions

Energy efficiency is very important in wireless sensor networks, as oftentimes sensor nodes are battery-operated. One way to reduce energy consumption is to turn off sensors that are not needed to sense data. Using this approach, the question becomes, how to schedule nodes to go to sleep? In this paper, we have studied the sleep scheduling problem for cluster- based sensor networks where sensors have variable transmit power. We proposed the Linear Distance-based Scheduling (LDS) scheme as an alternative to the conventional Randomized Scheduling (RS) scheme for selecting which nodes should sleep. Through

analysis and simulations, we showed that the LDS scheme can, indeed, save energy with negligible loss in sensing coverage area.

The drawbacks of the LDS and the RS schemes are that the average energy consumption of sensors with different distance x to the cluster head are different. In our future work, we will propose a new scheme based on the LDS scheme to balance the energy consumption for all sensors. Another approach to balance energy consumption would be to employ a dynamic cluster formation scheme.

Appendix A: Analysis of the network lifetime of LDS scheme

In the following analysis of network lifetime of the LDS scheme we assume that $\beta_s \leq 2/3$ (i.e., $x_s = R$) and only focus on the case when $\gamma = 2$. Therefore, the energy consumption of a node at x distance from the cluster head is

$$\mathcal{E}_{\text{LDS}}(x) = [1 - p(x)]E_{\text{active}}(x),$$

$$= \left(1 - \frac{3\beta_s x}{2R}\right) \cdot \left\{\lambda \cdot k_1 \cdot [\max(x_{\min}, x)]^2 + k_2\right\} \quad (16)$$

where $0 \leq x \leq R$.

Note that, when $x < x_{\min}$, $\varepsilon_{\text{LDS}}(x)$ is a decreasing function. Since $x_{\min}$ is usually much smaller than R, e.g. $x_{\min} = 0.1\,R$, the fraction of nodes reside within $x < x_{\min}$ is quite small. In order to simplify the analysis, we treat $\varepsilon_{\text{LDS}}(x)$ as $\varepsilon_{\text{LDS}}(x_{\min})$ for $x < x_{\min}$ in the following derivation.

Let us assume that $x \geq x_{\min}$. The first and the second derivatives of $\varepsilon_{\text{LDS}}(x)$ are

$$\mathcal{E}'_{\text{LDS}}(x) = \frac{-9\beta_s \lambda k_1 x^2 + 4R\lambda k_1 x - 3\beta_s k_2}{2R}$$

and

$$\mathcal{E}''_{\text{LDS}}(x) = \frac{-18\beta_s \lambda k_1 + 4R\lambda k_1}{2R},$$

respectively.

If $4R^2(\lambda k_1)^2 - 27\beta_s^2 \lambda k_1 k_2 < 0$, $\varepsilon_{\text{LDS}}(x)$ is a non-increasing function on $(0, R)$. Therefore, the time when β_d fraction of nodes run out of battery is the time when sensor nodes with $x \leq x_d^{(\text{LDS})}$ all run out of battery, where $x_d^{(\text{LDS})}$ satisfies:

$$\beta_d = \int_0^{x_d^{(\text{LDS})}} f(x)dx = \frac{[x_d^{(\text{LDS})}]^2}{R^2},$$
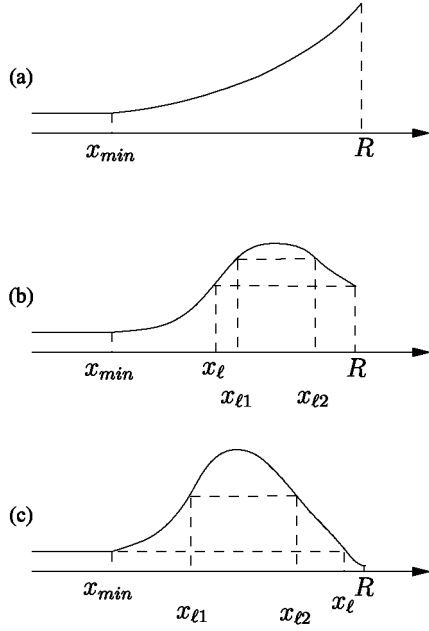
Figure 7. Different cases of $\mathcal{E}_{\mathrm{LDS}}(x)$.

leading to $x_d^{(\mathrm{LDS})} = R \cdot \sqrt{\beta_d}$. The network lifetime of the LDS scheme in this case is then

$$T_{\mathrm{LDS}}(\beta_d) = \frac{\Psi}{\mathcal{E}_{\mathrm{LDS}}\left(x_d^{(\mathrm{LDS})}\right)}$$

$$= \frac{\Psi}{\left(1 - \frac{3\beta_s\sqrt{\beta_d}}{2}\right)\left\{\lambda k_1 \left[\max\left(R \cdot \sqrt{\beta_d}, x_{\min}\right)\right]^2 + k_2\right\}}$$

Next we assume that

$$4R^2(\lambda k_1)^2 - 27\beta_s^2 \lambda k_1 k_2 \geq 0. \qquad (17)$$

In this case, the maximum of $\mathcal{E}_{\mathrm{LDS}}(x)$ is at

$$x = \alpha_1 = \frac{2R}{9\beta_s} + \frac{\sqrt{4R^2(\lambda k_1)^2 - 27\beta_s^2 \lambda k_1 k_2}}{9\beta_s \lambda k_1}.$$

and the minimum of $\mathcal{E}_{\mathrm{LDS}}(x)$ is at

$$x = \alpha_2 = \frac{2R}{9\beta_s} - \frac{\sqrt{4R^2(\lambda k_1)^2 - 27\beta_s^2 \lambda k_1 k_2}}{9\beta_s \lambda k_1}.$$

Note that (17) means that β_s must satisfy

$$\beta_s \leq \frac{2R}{3}\sqrt{\frac{\lambda k_1}{3k_2}}. \qquad (18)$$

Since $\beta_s \leq 2/3$ in our study, (18) means

$$R\sqrt{\frac{\lambda k_1}{3k_2}} \geq 1.$$

There are four cases that need to be considered. However, we only present two that are used to obtain numerical results in this paper.

Case 1. $\alpha_2 \leq x_{\min}$ and $\alpha 1 \geq R$, as shown in figure 7(a). In this case, $\varepsilon_{\mathrm{LDS}}(x)$ is a non-decreasing function on $(x_{\min}, R)$.

Therefore, the outer sensor nodes will run out of battery much faster than the inner sensor nodes. Similar to the case of RS scheme, the network lifetime $T_{\mathrm{LDS}}(\beta_d)$ is now

$$\frac{\Psi}{\left(1 - \frac{3\beta_s\sqrt{1-\beta_d}}{2}\right)\left\{\lambda k_1 [\max(R \cdot \sqrt{1 - \beta_d}, x_{\min})]^2 + k_2\right\}}.$$

Case 2. $\alpha_2 \geq x_{\min}$ and $\alpha_1 < R$. First we assume that $\varepsilon_{\text{LDS}}(R) \geq \varepsilon_{\text{LDS}}(x_{\min})$, as shown in figure 7(b). Let $x_\ell < R$ be another distance such that $\varepsilon_{\text{LDS}}(x_\ell) = \varepsilon_{\text{LDS}}(R)$. If

$$\frac{R^2 - x_\ell^2}{R^2} \leq \beta_d,$$

$T_{\text{LDS}}(\beta_d)$ is now

$$\frac{\Psi}{\left(1 - \frac{3\beta_s\sqrt{1-\beta_d}}{2}\right)\{\lambda k_1[\max(R \cdot \sqrt{1-\beta_d}, x_{\min})]^2 + k_2\}}.$$

If

$$\frac{R^2 - x_\ell^2}{R^2} > \beta_d,$$

we need to find the two distances $x_{\ell 1}$ and $x_{\ell 2}$, where $x_{\ell 1} < \alpha_1 < x_{\ell 2}$ such that

$$\beta_d = \int_{x_{\ell 1}}^{x_{\ell 2}} f(x)dx = \frac{x_{\ell 2}^2 - x_{\ell 1}^2}{R^2}$$

and

$$\varepsilon_{\text{LDS}}(x_{\ell 1}) = \varepsilon_{\text{LDS}}(x_{\ell 2}).$$

$T_{\text{LDS}}(\beta_d)$ becomes

$$\frac{\Psi}{\left(1 - \frac{3\beta_s x_{\ell 1}}{2R}\right)\{\lambda k_1[\max(x_{\ell 1}, x_{\min})]^2 + k_2\}}.$$

Second, we assume that $\varepsilon_{\text{LDS}}(R) < \varepsilon_{\text{LDS}}(x_{\min})$, as shown in figure 7(c). Let $x_\ell < R$ be another distance such that $\varepsilon_{\text{LDS}}(x_\ell) = \varepsilon_{\text{LDS}}(x_{\min})$. If

$$\frac{x_\ell^2}{R^2} \leq \beta_d,$$

$T_{\text{LDS}}(\beta_d)$ becomes

$$\frac{\Psi}{\left(1 - \frac{3\beta_s\sqrt{t a_d}}{2}\right)\{\lambda k_1[\max(R \cdot \sqrt{t a_d}, x_{\min})]^2 + k_2\}}.$$

If

$$\frac{x_\ell^2}{R^2} > \beta_d,$$

we need to find the two distances $x_{\ell 1}$ and $x_{\ell 2}$, where $x_{\ell 1} < \alpha_1 < x_{\ell 2}$ such that

$$\beta_d = \int_{x_{\ell 1}}^{x_{\ell 2}} f(x)dx = \frac{x_{\ell 2}^2 - x_{\ell 1}^2}{R^2}$$

and

$$\varepsilon_{\text{LDS}}(x_{\ell 1}) = \varepsilon_{\text{LDS}}(x_{\ell 2}).$$

The network lifetime is now

$$T_{\text{LDS}}(\beta_d) = \frac{\Psi}{\left(1 - \frac{3\beta_s x_{\ell 1}}{2R}\right)\{\lambda k_1[\max(x_{\ell 1}, x_{\min})]^2 + k_2\}}.$$

References:
1) I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40(8) (2002) 102–114.

2) B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, ACM Wireless Networks 8(5) (2002).

3) J. Deng, Y.S. Han, W.B. Heinzelman and P.K. Varshney, Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks. Computer Communications: special issue on ASWN04, 28 (2005) 1631–1642.

4) C.-R. Dow, J.-H. Lin, S.-F. Hwang and Y.-W. Wang, An efficient distributed clustering scheme for ad-hoc wireless networks, IEICE Trans. Commun. E85-B(8) (2002) 1561–1571.

5) W. Du, J. Deng, Y.S. Han and P.K. Varshney, A witness-based approach for data fusion assurance in wireless sensor networks. in: *15EEE Global Telecommunications Conference (GLOBECOM)* (San Francisco, CA, USA) (2003) 1435–1439.

6) J.L. Gao, Analysis of energy consumption for ad hoc wireless sensor networks using a bit-meter-per-Joule metric. IPN Progress Report 42– 150 (2002).

7) M. Gerla and J. Tsai, Multicluster, mobile, multimedia radio network, ACM/Baltzer Journal of Wireless Networks, 1(3) (1995) 255–265.

8) W.R. Heinzelman, A.P. Chandrakasan and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1(4) (2002) 660–670.

9) M. Kochhal, L. Schwiebert and S. Gupta, Role-based hierarchical self-organization for wireless ad hoc sensor networks, in: Proceedings of the Second ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'03) (2003) 98–107.

10) C.R. Lin and M. Gerla, Adaptive clustering for mobile wireless networks, IEEE Journal of Selected Areas in Communications 15(7) (1997) 1265–1275.

11) M.R. Pearlman, J. Deng, B. Liang and Z.J. Haas, Elective Participation in ad hoc networks based on energy consumption. in: *Proc. of 15EEE Global Telecommunications Conference (GLOBECOM) / Symposium on Ad-Hoc Wireless Networks (SAWN)*, Vol. 1. (Taipei, Taiwan) (2002) 26–31.

12) V. Raghunathan, C. Schurgers, S. Park and M.B. Srivastava, Energy-aware wireless microsensor networks, IEEE Signal Processing Magazine 19(2) (2002) 40–50.

13) F. Stajano and R. Anderson, The resurrecting duckling: security issues in ad-hoc wireless networks, in: B. Christiansen, B. Crispo, and M. Roe (eds.), *Security Protocols, 7th 15nternational Workshop, Volume 1796 of Lecture Notes in Computer Science*, (1999) 172– 194.

14) D. Tian and N.D. Georganas, A coverage-preserving node scheduling scheme for large wireless sensor networks, in: *Proceedings of the First ACM 15nternational Workshop on Wireless Sensor Networks and Applications*. (2002) 32–41.

15) P.K. Varshney, *Distributed Detection and Data Fusion*. (New York: Springer) (1997).

16) Y. Xu, J. Heidemann and D. Estrin, Adaptive energy-conserving routing for multihop ad hoc networks, Research report 527, USC/Information Sciences Institute (2000).

17) W. Ye, J. Heidemann and D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: *Proceedings of the 21 st 15nternational Annual Joint Conference of the 15EEE Computer and Communications Societies (15NFOCOM 2002)* (2002) 1567–1576.

Notes:

1. Another approach is to use a low power wake-up circuit as in the WINS project, but a drawback of this approach is that it may suffer from the so-called "sleep deprivation torture attack" [13] by malicious nodes.

2. Sometimes, the distance might be difficult to obtain, then the power of the received signal during the initialization process might be used. The accuracy of the distance estimation affects the performance of our scheme. However, these are out of the scope of this work.

3. This deployment model has been frequently used in the technical literature [6]. Note that a different nodal deployment distribution may lead to different derivations. However, we believe that it will not change our conclusions on the superiority of the scheme proposed in this work.

4. A distributed algorithm may also be employed such that sensor nodes may decide to go to sleep based on this probability without the intervention of the cluster head.

5. The energy consumption of the cluster head is expected to be similar for both the LDS and the RS schemes based on the assumption that the cluster head always uses the same transmission power to

contact the sensors in its cluster. Since we are only interested in the relative comparison of these two schemes, we do not include the energy consumption of the cluster head in our analysis.

6. The sleeping nodes do not generate any traffic to send to the cluster head. However, we want to stress that the neighborhood of the sleeping nodes are covered by other active neighboring sensors. We study the sensing coverage performance of our proposed scheme in Section 4.2.

7. When this probability is not set appropriately, the inner nodes may run out of energy before the outer nodes do. A more detailed studies can be found in [3].

8. Indeed, according to (8), when $\beta \beta,, > 2/3$, it is necessary to put a fraction of sensors near the border of the cluster to sleep with probability 1. For a static cluster formation, these sensors will be wasted; however, when using a dynamic cluster formation, which changes cluster heads and members of clusters constantly, these sensors may be used after the cluster formation is changed.

9. We concentrate on active sensors in this work. Passive sensors might be put into sleep while maintaining their sensing capability, therefore, sensing coverage will not be affected by our scheme at all [12].

10. Since two locations can be covered by a common sensor, not all the events that a location is not covered by any active node are independent. Therefore, this estimate is only an upper bound on the desired ratio.

11. Of course, this is only true when we neglect boundary effects.

12. We only simulated an abstract level of a sensor network with clusters. Nevertheless, our results shed lights on the performance improvements of our scheme. More results can be found in our later work [3].

13. These results only have relative significance, as network lifetime depends largely on iY, $k1$, $k2$, y, and other system parameters.