

## Research Article

# Stair Scheduling for Data Collection in Wireless Sensor Networks

Jinbiao Chen,<sup>1</sup> Yongcai Wang,<sup>1</sup> Yuexuan Wang,<sup>1</sup> and Changjian Hu<sup>2</sup>

<sup>1</sup> Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, Beijing 100084, China

<sup>2</sup> NEC Laboratories, Beijing, China

Correspondence should be addressed to Yongcai Wang; wangyc@tsinghua.edu.cn

Received 19 August 2012; Accepted 15 January 2013

Academic Editor: Liusheng Huang

Copyright © 2013 Jinbiao Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Spatially organized clusters are basic structure for large-scale wireless sensor networks. A cluster is generally composed by a large amount of energy-limited low-tier nodes (LNs), which are managed by a powerful cluster head (CH). The low-tier nodes that are close to the cluster head generally become bottlenecks in data collection applications. Energy efficient scheduling is important for the low-tier sensors to be longevous while guaranteeing reliable communication. In this paper, based on three aspects of performance considerations including network longevity, multihop communication reliability, and sensing system cost minimization, we propose a stair duty-cycle scheduling method for the low-tier sensors. It is designed to make the LNs in the same cluster sleep cooperatively for most of the time and wake up in assigned sequence for multihop communication. Stair scheduling cannot only improve the energy efficiency of the network but also guarantee high communication reliability and low transmission delay. Efficiency of the proposed stair scheduling is verified by analysis and intensive simulations. The results show that the performances of stair scheduling are much better than that of random scheduling algorithms.

## 1. Introduction

Large-scale sensor networks are attracting great research interests, because they are promising in various applications such as precision agriculture and environment monitoring [1]. In order to cover the broad area of interest where information should be monitored, such a sensor network often contains thousands or tens of thousands of small and energy limited sensors. If these sensors are directly managed by the base station, the network will suffer large communication overhead, energy inefficiency problems, and unreliability multihop communication problems. Clustering scheme was proposed by researchers to organize the sensors into two-tiered structure [2]. In the higher tier, some energy-rich sensors are deployed as backbones to organize the energy-limited sensors within their geographic neighborhood to form clusters. In the lower tier, the energy-limited sensors capture, encode, and transmit relevant information of the designated area to the cluster head (CH). Since the CHs are rich in resources, the system performances are mainly determined by the lower tier nodes (LNs.)

*Sleeping and scheduling* technique is the main solution to conserve energy of the LNs. As reported in INSIGHT [3], an LN works in “sleep” state in terms of radio OFF, sensor OFF, and with HPL management can save energy up to two magnitudes than the basic “listen” state. This suggests putting LNs into sleep state for most of the time and only wake them up in periodical short slot for data capturing and transmitting. But because the LNs do not serve in sleep state, this leads to an LN scheduling problem for both energy saving and QOS preserving.

Many single-hop LN scheduling schemes were proposed. In single-hop cluster, every LN transmits data directly to the CH. The LNs need not forward messages from other nodes, so they can turn to the sleep state independently to save energy [3, 4]. In [4], high density LNs are deployed in each cluster, and a linear-distance-based scheduling has been used to define the sleep schedule of the LNs. In [5], small portion of LNs are scheduled to be activated among redundant deployed SNs for coverage preserving. In [6], LNs are scheduled based on analytical hierarchy process which considers residual energy, sensing coverage overlapping, and

so forth. However, single-hop cluster is limited in scale. Many expensive CHs will be needed to cover a broad land.

Multi-hop cluster in which LNs transmit data along a multi-hop route towards the CH is proposed by the researchers of [7, 8], which has better scalability and is more cost-efficient. But the LN scheduling problem becomes difficult in a multi-hop scenario. The LNs can no longer sleep independently, because if a relay node turns to sleep, the multi-hop route which it serves will be shut down. To support multi-hop communication, joint LN scheduling becomes necessary. In [9], distributed data gathering scheduling in multihop sensor networks was proposed by using greedy algorithm to extract a rooted spanning tree. Their work focuses on routing. It did not address jointly scheduling problem for reliable communication. In [8], hops-based sleep scheduling algorithm was proposed to assign different active probabilities to different hop LNs to balance the energy consumption. In [10], localized probabilistic routing algorithm was proposed for optimizing network lifetime. However, in their studies, the communication was assumed to happen round by round. How the LNs are jointly scheduled for multi-hop communication was not explicitly discussed.

We propose a stair scheduling method in this paper. Its basic idea is to schedule the LNs in one cluster to sleep and work cooperatively to conserve energy as well as to support multi-hop communication. With this purpose, in stair scheduling, we act as follows.

- (1) We propose “Stair Scheduling” to control the working slots of the LNs based on the level-based routing tree [8]. The cluster is divided into levels, and a child node in the  $i$ th level always transmits data to a parent in the  $(i-1)$ th level. The child always activates and sleeps one slot earlier than its parent. Each slot contains  $m$  slices, and an LN randomly selects one slice to transmit data in its active transmission slot for collision avoidance.
- (2) hop-by-hop time synchronization is proposed to avoid time drifting and make the “Stair Scheduling” work in a fully distributed manner.
- (3) Average function hop-by-hop data aggregation is proposed to further enhance the energy saving as well as to balance the energy consumption.

We further analyze the energy and communication reliability models of the joint scheduled cluster with respect to the cluster size and LN parameters. The results are applied to optimize the design of the network to choose a suitable number of CHs for system cost minimization. Simulation results verify the energy efficiency and communication reliability of stair scheduling and further show that it is more efficient and applicable than traditional single-hop random scheduling (SRS) and multi-hop random scheduling (MRS).

The rest of this paper is organized as follows. In Section 2, we describe the system model and formulate the problem. In Section 3, we develop the stair scheduling algorithm. Section 4 presents the energy and reliability models of the joint scheduled cluster and the network optimization results using stair scheduling. Section 5 summarizes the simulation

results and the paper is concluded in Section 6, with remarks and future work discussions.

## 2. System Model and Problem Formulation

**2.1. Assumption and Network Model.** The basic assumptions and network model are outlined as follows.

- (i) We consider two-tiered sensor network, which contains low-tier energy-limited nodes (LNs) and high-tier cluster heads (CHs). The LNs are not redundant. They are deployed in an economic way for exact land coverage. Each node is responsible for the monitoring of its own vicinity. They capture and transmit data periodically. The sampling frequency is denoted as  $U$ .
- (ii) The CHs do not sense data but receive and aggregate raw data from LNs and report the result to the base station. The CH has enough energy and is never considered to be a bottleneck.
- (iii) The cluster area is in circular shape with radius  $R$  and centered by the CH. LNs are evenly distributed in the cluster area. All the LNs have the same sensing range  $r_s$ , maximum transmission range  $r_c$ , and the initial energy  $E$ , and we assume  $r_c = 2r_s$ .
- (iv) Each LN generates  $l$ -bit data in a period and transmits the data towards the CH using a level-base routing tree [8, 11]. Symmetry link is assumed in data forwarding that if A can hear B, then B can hear A.

**2.2. Problem Formulation.** We consider large-scale sensor monitoring system, such as habitat or agriculture monitoring. In these systems, application requires that each LN reports its vicinity's information  $U$  times per hour. This prohibits an LN from sleeping for an arbitrary long time. Every LN must be active periodically. Since only CH reports results to the base station, the sampling frequency of LNs should be the same with the reporting frequency of the CH. Therefore,  $U$  is denoted as the *sampling frequency* of the cluster, and the *sampling period* is  $T = 1/U$ .

For an LN, each sampling period can be further divided into *task slots*. In each task slot, the LN can choose to sleep, sense, process data, receive message, or transmit message. We omit sensing and data processing time. Therefore, in a task slot, the LN either sleeps, receives, nor transmits message. We suppose that each slot is equal in length and denote the slot length by  $T_m$ . Each sample period is therefore divided into  $M = T/T_m$  task slots. The joint scheduling problem becomes slot state assignment problem with joint consideration of the system performance. Formally, in a sample period, the  $i$ th LN has slots  $\mathbb{S}^i = \{S_1, S_2, \dots, S_M\}$ , and the task of scheduling is to determine the state of each slot.

If a cluster has  $N$  LNs, each sampling period contains  $M$  task slots, each slot can be assigned  $v$  states, and the solution space of the joint scheduling problem will be  $v^{MN}$ . We can see that it encounters a combination explosion problem with respect to  $v$ ,  $M$ , and  $N$ . In addition, because an LN is commonly not aware of other LNs' status, the joint scheduling problem is challenging.

But a heuristic is that, for energy saving and in order to support multi-hop communication, the LNs in one cluster should sleep for most of the time and should have some overlaps in their active time slots to exchange message. In view of this, we propose a multi-hop joint scheduling (stair scheduling) algorithm. The details will be given in the next section.

### 3. Multihop Joint Scheduling

**3.1. Overview of Stair Scheduling.** The basic idea of stair scheduling is to schedule the LNs in one cluster to sleep and work cooperatively to conserve energy as well as to support multi-hop communication. The design of joint scheduling basically contains three schemes as follows.

- (1) “Stair scheduling” is proposed to assign every LN three continuous active slots in a sample period: “R-Slot” to listen to child nodes, “T-Slot” to sense and forward data, and “Syn-Slot” to synchronize time. A child node will be always activated one slot earlier than its parent, so that its forwarding can be heard by its parent. Therefore, most energy is conserved by sleeping and reliable multi-hop communication is supported. For collision avoidance, every task slot is further designed to contain  $m$  slices. The length of each slice is equal to the atomic data transmission time, which is denoted by  $t$ . So the slot length  $T_m = mt$ . In “T-slot” anLN randomly selects one slice to transmit data for collision avoidance.
- (2) A hop-by-hop time synchronization scheme is proposed to avoid sensor time drifting and make stair scheduling work in fully distributed manner. Due to “Stair scheduling,” when a parent LN forwarding its message in its “T-Slot,” for the symmetric link, its children will hear this broadcasting in their “Syn-Slot.” The children synchronize time with the parent. When the hop-by-hop synchronization reaches the CH, the whole link is synchronized and “Stair scheduling” can work distributedly.
- (3) Average-function-based hop-by-hop data aggregation is proposed to further enhance energy saving and to balance the energy consumption.

The detailed design of the proposed stair scheduling is as follows.

**3.2. Stair Scheduling.** “Stair scheduling” is to assign active task slots to LNs based on the hop counts of the LNs. It is initialized by the CH and is continually maintained by the LNs themselves during the data collection process.

**3.2.1. Premise for Stair Scheduling.** The level-based energy-balance routing (LEB) tree [8] and time synchronization among LNs and CH should be achieved at the cluster formation phase. The level-based energy-balance routing tree [8] is different from the shortest-path tree. Sensors are divided into levels based on the average hop progress, which can be easily

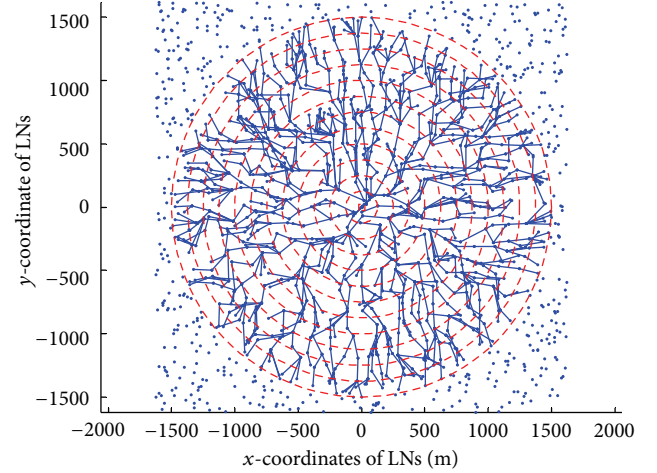


FIGURE 1: Level-based energy-balance multihop communication in a cluster.

obtained by flooding during the initialization phase [2, 12]. In LEB, for energy balance, a flag is assigned to each LNs, which is initially set to 0 and is changed to 1 once it acts as a relay for any other sensors. So when the node in the  $i$ th level finds its closest neighbor in the  $(i-1)$ th level and has flag = 1, it will try to select the second closest neighbor in the  $(i-1)$ th level as the relay. If it happens that all the neighbors in the  $(i-1)$ th level have flag = 1, the sensor in the  $i$ th level will randomly choose a neighbor in the  $(i-1)$ th level as the relay node.

After initialization of LEB, the CH is aware of the levels of all the LNs and all the links in its cluster. Figure 1 shows a graph illustration of the CH’s knowledge after LEB construction in a cluster. The square in the center is the CH and the surrounding points are LNs. The dashed circle is the levels.

**3.2.2. Stair Scheduling Scheme.** If the largest level is  $n$ , and the sample period of each LN is divided into  $M$  task slots, the “stair scheduling” works as follows.

(1) For an  $i$ th level of the LN, it will sleep from its 1st to  $(n-i-1)$ th slots and will activate at the  $(n-i)$ th slot. The active status will last three slots, and then the LN turns to sleep, till the end of the sample period. Among the three active slots, we have the following.

- (i) The first slot is used to receive messages from the child LN in the  $(i+1)$ th level. We call it “R-Slot.”
- (ii) The second slot is used to sense local area and forward message to a parent node in the  $(i-1)$ th level. We call it “T-Slot.”
- (iii) The third active slot is used to synchronize time between the parent node and this node. Since the link is symmetry, when the parent node relays the message towards CH, the broadcasting will be overheard by this child. The overheard message is processed to synchronize time between the parent and this node. We call this slot “Syn-Slot.”

For communication collision avoidance, each slot is designed to contain  $m$  slices. In “T-Slot,” the LN randomly selects one slice for data transmission. Carrier sense and retransmission scheme is not used in stair scheduling.

(2) A special case will appear for the level  $n$  LNs. Since they do not relay other LNs’ message, they only assign their first two slots to be active.

- (i) The first slot is a “T-Slot” to sense local area and transmit message.
- (ii) The second slot is used to synchronize time with a parent node in the  $(n - 1)$ th level, which is a “Syn-Slot.”

Figure 2 shows a graph illustration of the task slot assignments for LNs in the different levels. We can see two features of “stair scheduling” as follows.

- (1) The active task slots of LNs form a stair shape with respect to the hop count; that is, a child LN will be activated one slot earlier than its parent.
- (2) For LNs in two neighboring levels, they have two overlapped active slots. The “T-Slot” of the child overlaps with the “R-Slot” of the parent. This establishes the link for data transmission. The “Syn-Slot” of the child overlaps with the “T-Slot” of the parent. This establishes the hop-by-hop time synchronization.

**3.3. Time Synchronization.** Time synchronization plays an important role in stair scheduling. It is carried out during two phases: initialization phase and run-time hop-by-hop time synchronization.

In the initialization phase, time synchronization is carried out by Flooding Time Synchronization Protocol (FTSP) [13]. This is done simultaneously with the setting up of the LEB routing tree. Initially, all the LNs are active and their timers are not synchronized. The CH periodically broadcasts  $m$  CH messages for both time synchronization and LEB routing tree. The message generation time  $t_{0,j}$  is broadcasted in each message, where  $j$  is the message index. When an LN receives the CH message, it uses the MAC-layer time-stamping to measure the packet forwarding delay. With the recorded delays of the previous hops, the LNs in the  $i$ th hop adjust their clock to

$$t_{i,j} = t_{0,j} + \sum_{k=1}^{i-1} s_{k,j}, \quad (1)$$

where  $s_{k,j}$  is the delay at the  $k$ th hop. Since  $m$  CH messages are broadcasted periodically each LN will obtain a sequence of synchronized time  $\{t_{i,1}, \dots, t_{i,m}\}$ . Linear regression is used to achieve high accuracy of the clock synchronization [13].

In the running phase, time synchronization is done hop-by-hop to avoid clock drifting. In “T-slot,” LN broadcasts sensing data together with its local time. This broadcasting will be heard by its child nodes due to the symmetry link. These child nodes use the MAC-layer time stamping to record the processing delay and adjust their clock to the clock of their parents. With this hop-by-hop scheme, children synchronize

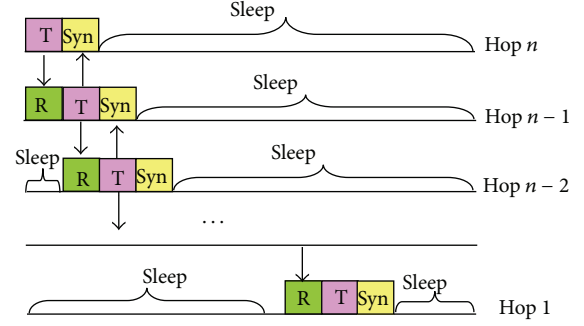


FIGURE 2: Joint active task slot scheduling with the “stair scheduling” method.

clock with their parents, and the process repeats until the CH is reached as the last parent. Hence the clocks of the whole cluster can be continuously synchronized. And based on this, all LNs can maintain the “stair scheduling” in a fully distributed manner.

**3.4. Hop-by-Hop Data Aggregation.** In addition to stair scheduling, we adopt hop-by-hop data aggregation to enhance energy saving. For data collection with reverse shortest-path tree, the ideal data aggregation will be chain-based hop-by-hop data aggregation as discussed in PEGASIS [14]. However, such chain-based data aggregation poses high requirement to the scheduling of node transmission. Any child node should transmit earlier than its parent sensor; otherwise, redundant packet will be generated and additional energy will be consumed [15]. However, the authors of [14] have not discussed the details of the transmission scheduling.

The “stair scheduling” of stair scheduling provides fine solution to such chain-based hop-by-hop data aggregation, because in stair scheduling any child sensor will transmit data one slot earlier than its parent sensor. To implement hop-by-hop data aggregation, for an LN node, in R-Slot, it listens to messages from its child nodes. The data is extracted from the messages and is stored. In T-Slot, the LN senses its vicinity. The local readings will be merged with the stored data, and the results will be forwarded towards CH. We propose *average function* in data aggregation. If the size of the received data is  $l_r$  and the size of the local sensed data is  $l_s$ , using *average function*, the size of the merged data to be forwarded will be  $l_f = \max\{l_r, l_s\}$ . The energy performance of stair scheduling using average aggregation functions will be analyzed and evaluated in Sections 4 and 5.

## 4. Performance Analysis of Stair Scheduling for Network Optimization

In this section, we analyze the multi-hop reliability and energy consumption model of stair scheduling with respect to the cluster parameters. The model will be applied to optimize the tier-structure design of the sensor network.



**4.1. Basic Analysis.** We consider large-scale sensor systems, where sensors are not redundantly deployed for the cost consideration. Each sensor is responsible for the monitoring of its own vicinity. The sampling frequency  $U$  is commonly very low, for example, two samples per hour, and so forth.

For the monitoring completeness, there must be enough LNs deployed in the field to provide full land coverage. Based on the result of 1-coverage [16] analysis that calculates the probability of any point covered by at least one LN, the density of LN deployment can be derived as

$$\lambda = -\frac{\log(1 - P_{1\text{-cover}})}{\pi r_s^2}, \quad (2)$$

where  $P_{1\text{-cover}}$  is the desired 1-coverage probability. Since CHs do not sense data, whatever the cluster size chosen, for coverage preserving, the required LN density is  $\lambda$ . When CHs are deployed, the LNs are organized into clusters, and the number of LNs in each cluster can be calculated as  $N = \lambda \pi R^2$ . The following performance analysis and simulations are all carried out in such a coverage preserving scenario.

A simple but effective energy consumption model for sensor operations is assumed in this paper [1, 8, 17]. To transmit  $l$ -bits data over distance  $d$ , the sender will expend energy as

$$E_{TX}(l, d) = l * E_{\text{elec}} + l * \epsilon_{\text{amp}} * d^\gamma, \quad (3)$$

where  $E_{\text{elec}}$  is the unit and  $\epsilon_{\text{amp}}$  is the amplifier energy.  $\gamma$  is the path loss factor, usually  $2 < \gamma < 4$ , and here we assume  $\gamma = 2$ . The energy expended by a receiver depends only on the length of the data and can be expressed as

$$E_{RX}(l) = l * E_{\text{elec}}. \quad (4)$$

Therefore, the key issue of energy consumption model of stair scheduling counts the transmission and reception times of a sensor in a sample period. In [8], the average hop progress of the sensor network  $r_h$  was derived as

$$r_h = \sqrt{3\lambda} \int_0^{r_c} x^2 e^{(\pi/3)\lambda(x^2 - r_c^2)} dx, \quad (5)$$

where  $r_c$  is the sensor's maximum transmission range and  $\lambda$  is the density of the sensors. The result divides the circular shape cluster into  $n = R/r_c$  levels and verifies that sensors in the  $i$ th level have the most probability to reach CH by  $i$  hops. With this result, due to the uniform sensor distribution, the average number of sensor nodes in each level is

$$N_i = \pi \lambda (2i - 1) r_h^2, \quad (6)$$

where  $i = 1, \dots, n$ . We assume that cluster radius  $R$  can be divided exactly by  $r_h$ .

According to the uniform distribution, an outer level will have more sensors than an inner level. So the average number of children for a sensor in level  $i$  is

$$C_i = \frac{N_{i+1}}{N_i}. \quad (7)$$

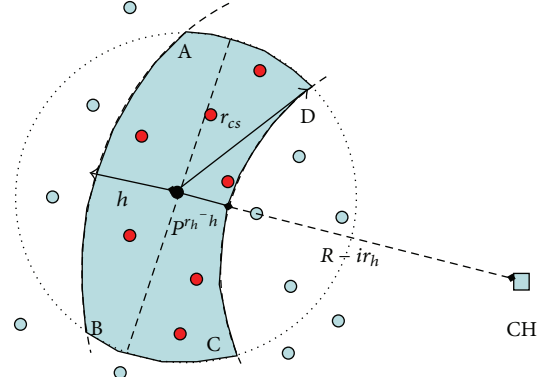


FIGURE 3: The interference region.

**4.2. Multihop Communication Reliability.** With “stair scheduling,” for an arbitrary LN, if its “T-slot” comes, it randomly selects a slice within the  $m$  slices of the “T-slot” to transmit data. If two LNs share the same parent and select the same slice to transmit data, their messages will collide, and the transmissions will fail. Although the link can be enhanced by retransmission schemes, and so forth [18], we focus more on the performance of the basic stair scheduling and leave the enhanced methods to future work.

We suppose that an LN node  $C$  in the  $i$ th level tries to transmit. For the “stair scheduling” of stair scheduling, only the LNs in the same level share the same parent of  $C$  may collide  $C$ 's transmission. Recall that the parent node found children nodes by broadcasting with radius  $r_c$  in the initialization phase. The “coparent circle” of  $C$  is therefore a circle centered at its parent node, and with radius  $r_c$ . We suppose that  $C$ 's parent is node  $P$ , which locates in the  $(i-1)$ th level. Figure 3 shows the scenario. In the figure, the region of the  $i$ th level is ring centered at CH and with inner radius  $(i-1)r_h$  and outer radius  $ir_h$ . The “interference region” is the region indicated by ABCD, which is the overlapped area of the “coparent circle” and the  $i$ th level ring.

**4.2.1. Lower Bound of Transmission Reliability.** Because even two LNs are within the “interference region,” they may have different parents and their transmissions may not collide. So based on the following assumptions, we will get a lower bound of the transmission reliability. The assumption is as follows: “if two LNs are within interference region of each other and transmit data in the same time slice, their transmissions will collide.” The lower bound will help us to understand the worst case of the transmission reliability of stair scheduling to direct cluster design. We can see that the area of the “interference region” is a function of  $i$  and  $h$ , where  $h$  is the distance from  $P$  to the inner border of the  $i$ th level ring. We denote the area by  $S_c(i, h)$ . Since LNs are uniformly deployed, the probability that there are  $k$  LNs within  $S_c$  is  $p(k, S_c) = ((\lambda S_c)^k / k!) e^{-\lambda S_c}$ . If all the other  $k-1$  LNs select

different slices from  $C$ 's transmission slice,  $C$ 's transmission will success. This probability is

$$p(i, h) = \sum_{k=0}^{\infty} \left(1 - \frac{1}{m}\right)^{k-1} \frac{(\lambda S_c(i, h))^k}{k!} e^{-\lambda S_c(i, h)} \quad (8)$$

$$= \frac{m}{m-1} e^{-\lambda S_c(i, h)/m},$$

where  $\lambda$  is node density and  $m$  is the number of slices. We can see that  $p(i, h)$  is a monotone decreasing function of  $S_c(i, h)$ . If we can find an upper bound for  $S_c(i, h)$ , we will get the lower bound of  $p(i, h)$ .

We firstly consider the case  $r_c < (i-1)r_h$ , where the “co-parent” radius is smaller than the inner radius of the ring. This is corresponding to the case when  $C$  locates in some outer levels in which  $i > 1 + r_c/r_h$ . Since  $r_c < 2r_h$  [8], this is corresponding to the case when  $i \geq 3$ .

It is easy to give an upper bound to  $S_c(i, h)$  for this case, because the parent node  $P$  should locate in the  $(i-1)$ th level. As shown in Figure 4, when it locates on the inner border of the  $i$ th level, the “co-parent circle” and the  $i$ th level ring will form the maximum overlapped area, which is indicated by EFGH. It is easy to verify that the area of EFGH is an upper bound of  $S_c(i, h)$ . We denote this upper bound interference area as  $S_{up}(i)$  as follows:

$$S_{up}(i) = \frac{2r_c^2}{\pi(\beta - \theta)} + \frac{2\alpha(ir_h)^2}{\pi}$$

$$- r_c(i-1)r_h \sin \beta - \frac{2\gamma((i-1)r_h)^2}{\pi} \quad (9)$$

$$+ r_c((i-1)r_h) \sin \theta$$

$$\geq S_c(i, h), \quad \forall h \in [0, r_h],$$

where

$$\alpha = \arccos \frac{(ir_h)^2 - r_c^2 + ((i-1)r_h)^2}{2i(i-1)r_h^2},$$

$$\beta = \arccos \frac{((i-1)r_h)^2 + r_c^2 - (ir_h)^2}{2(i-1)r_h r_c}, \quad (10)$$

$$\theta = \arccos \frac{((i-1)r_h)^2 + r_c^2 - (ir_h)^2}{2(i-1)r_h r_c},$$

$$\gamma = \pi - 2\theta.$$

Now we consider the cases when  $i = 1$  or  $i = 2$ . When  $i = 1$ , the LNs transmit directly to CH. The maximum “interference area” will be  $S_{up}(i) = \pi r_h^2$ . When  $i = 2$ , the upper bound of the “interference area” will be  $S_{up}(i) = \pi r_c^2 - r_h^2$ , as illustrated in Figure 5.

Therefore, we got the lower bound of the expected communication reliability of the  $i$ th hop as

$$\frac{P(i)}{m-1} = \frac{m}{m-1} e^{-\lambda S_{up}(i)/m} \leq P(i). \quad (11)$$

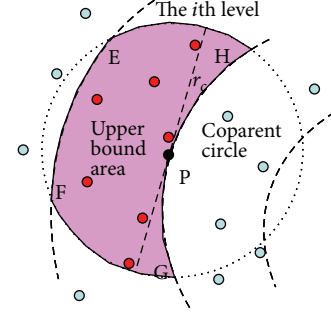


FIGURE 4: Upper bound when  $i \geq 3$ .

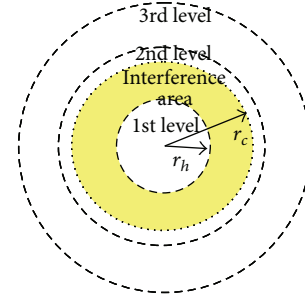


FIGURE 5: Upper bound of  $S_c(i, h)$  when  $i = 2$ .

Therefore, the lower bound of the multi-hop communication reliability when a message is forwarded from the level  $i + j$  to the level  $i$  is

$$\frac{P(i+j, i)}{m-1} = \prod_{k=i+1}^{i+j} \frac{P(i+k)}{m-1}. \quad (12)$$

With the parameter settings in Table 1, where  $P_{1-cover} = 0.99$ ,  $n = 12$  and  $r_c = 200$ , the lower-bound of the hop-by-hop communication reliability and the lower bound of the multi-hop communication reliability are shown in Figure 6. We can see basically the following

- (i) The lower bound varies with  $i$ . The lower bound increases as  $i$  decreases from 12 to 3. This is because the interference region becomes smaller as the level decreases.  $P(i)$  is small in the first and second levels, because, in such cases, the LNs are close. The number of LNs in the interference region becomes larger.
- (ii) The multi-hop communication reliability decreases with the increasing of the forwarding hops. This gives us hints that in designing cluster, we should limit the number of forwarding hops.

**4.3. Energy Consumption Model of Stair Scheduling.** After analyzing the communication reliability in each hop, we derive the energy model for LNs in different hops. We assume that when two messages collides the energy for message transmission and reception will be consumed.

For an LN in the outmost  $n$ th level, it transmits its sensed data to its parent in its “T-Slot” and receives a time

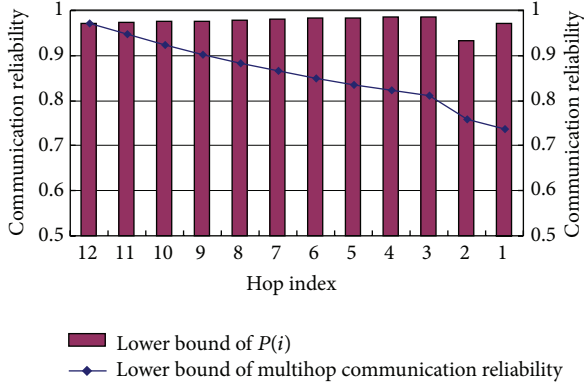


FIGURE 6: The lower bound of  $P(i)$  and the multihop reliability derived using the lower bound of  $P(i)$ .

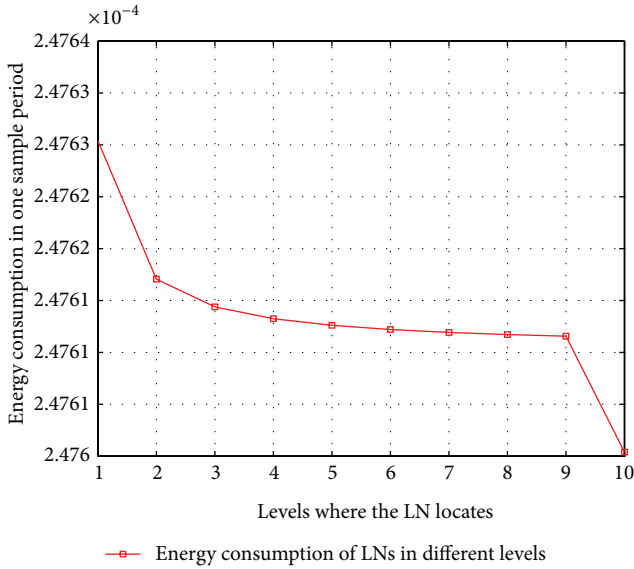


FIGURE 7: Energy consumption in one sample period where LNs locate in different levels.

synchronization message from this parent in the “Syn-Slot.” Its energy consumption in one sample period is

$$E(e_n) = l * E_{elec} + l * \epsilon_{amp} * (r_h)^y + l * E_{elec}. \quad (13)$$

For an LN in the  $i$ th level, it receives data in “R-Slot” from its children. In “T-Slot,” it senses local area and transmits the aggregated data to its parent. Since the received message is  $l$ -bits, and the local generated data is  $l$ -bits, the aggregated data with also be  $l$ -bits due to the average aggregation function. Then, in its “R-Slot,” it receives a time synchronization message from this parent. Its energy consumption in one sample period is

$$E(e_i) = \left(2 + \frac{N_{i+1}}{N_i}\right) * l * E_{elec} + l * \epsilon_{amp} * (r_h)^y. \quad (14)$$

The comparison of the energy consumption for LNs in different levels is shown in Figure 7, with parameter settings in Table 1. We can see the following.

TABLE 1: Setting of parameters.

| Parameter        | Value      |
|------------------|------------|
| $E_{elec}$       | $10^{-10}$ |
| $l$              | 100 (bits) |
| $r_s$            | 100 (m)    |
| $P_{1-cover}$    | 0.999      |
| $\epsilon_{amp}$ | $10^{-10}$ |
| $\gamma$         | 2          |
| $r_c$            | 200 (m)    |
| $n$              | 10         |

(1) LN in the first level has the highest energy consumption rate, and LN in the outmost level has the lowest energy consumption rate. This is due to that the first level has the fewest number of sensors, so they averagely have the heaviest burden to relay data. But the outmost LNs need not to relay data at all.

(2) The difference among energy consumption rates is very small. The difference between energy consumption of the first level LNs and the tenth level LNs is only  $(2.4763 - 2.476)/2.476 = 0.012\%$ . This balance owes to the “stair scheduling” and “average aggregation function” of stair scheduling. By “stair scheduling,” LNs in different levels activate for almost the same amount of slots. By average aggregation, different levels of LNs receive and forward nearly similar amount of data.

With the energy consumption model of stair scheduling, we can further analyze the energy consumption of the cluster and apply the analysis result to optimize the design of the tier structure.

**4.4. Optimize the Design of Tier Structure.** We can use the derived performances of stair scheduling to optimize the design of cluster. This is a typical multiple objectives optimization problem.

*The controllable variable* is the number of levels:  $n$ , which indeed determines the size of cluster. If a cluster has  $n$  levels, the size of one cluster is  $\pi * (nr_h)^2$ .

*The constraint* is that the cluster area should cover the full sensing field. For using the minimal number of clusters to cover the sensing field, the cell CH distribution [19] will be optimal. Suppose the area of sensing field is  $S$ , and the required number of CHs is

$$n_{CH} = \frac{2 * \pi}{3\sqrt{3}}. \quad (15)$$

*The energy metric* is to minimize the energy consumption of the first level, since it is the bottleneck of the lifetime of the cluster. So  $m_1 = E(e_1)$ .

The communication reliability metric is to maximize the reliability to forward a message from level  $n$  to level 1 as follows:

$$m_2 = \underline{P}(n, 1) = \prod_{k=1}^{n-1} \underline{P}(k). \quad (16)$$

The cost metric is to minimize the deployment cost of the sensors. Since the number of LNs is fixed by coverage preserving requirement. The cost is determined by the number of CHs, that is to minimize  $m_3 = n_{CH} * \text{cost}$ .

By assigning weights to different metrics, we can arrive at an optimized design of  $n$  by performance trading off as

$$n = \arg \min_n (w_1 m_1 - w_2 m_2 + w_3 m_3), \quad (17)$$

where  $w_1, w_2$ , and  $w_3$  are user assigned positive weights. Discussion of the cluster optimization will be out of the scope of this paper. We leave it to future work. We simply choose  $n = 10$  in the following evaluations.

## 5. Performance Evaluation

**5.1. Simulation Settings.** We build a discrete event simulator using MATLAB 7.0 to evaluate the performances of stair scheduling. We compare the performances of the proposed stair scheduling with traditional single-hop randomized sleeping scheduling (SRS) [3] and multi-hop random sleeping scheduling (MRS). The results shown in Figures 9–13 are the means of 100 independent runs. In each simulation run, we generate a certain number of LN nodes and randomly place them in a rectangle sensing area with size  $5000 * 5000 \text{ (m}^2\text{)}$ . The number of nodes is determined by the node density, which is calculated by (2) by varying the 1-coverage probability from 0.9 to 0.99. Each LN captures data and reports data to CH. The sensing radius of LN is  $r_s = 100$  meters and the communication radius  $r_c = 200$  meters. The sample period of LN is half an hour  $T = 1800,000 \text{ ms}$  and the time slice to finish a transmission event is fixed at  $t = 9 \text{ ms}$ . The length of the task slot is  $T_m = mt$ , where  $m$  varies in the range of (10, 100).

SRS uses single-hop cluster in which every LN directly transmits data to CH. The number of clusters is decided based on the constraint of cluster coverage [19]. In each cluster, the LNs are independently scheduled, with active probability  $p_{RSS} = T_m/T$ . Transmission will succeed if two transmissions do not collide.

Stair scheduling and MRS uses the same size of clusters and they both use LEB routing and divide the cluster into the same  $n$  levels. In MRS, LNs in each level are scheduled independently to capture, transmit, and forward data. When an LN in the  $i$ th level is activated, it broadcasts message towards level  $i - 1$ . If the message reaches its parent LN in the level  $i - 1$ , the transmission will succeed; otherwise, the transmission in this hop will fail. For the fairness of SRS, stair scheduling, and MRS, we set the average active duration in one period to be the same  $p_{SRS} = p_{MRS} = T_m/T$ . LNs in SRS, MRS, and stair scheduling share the same parameters and the same energy coefficients.

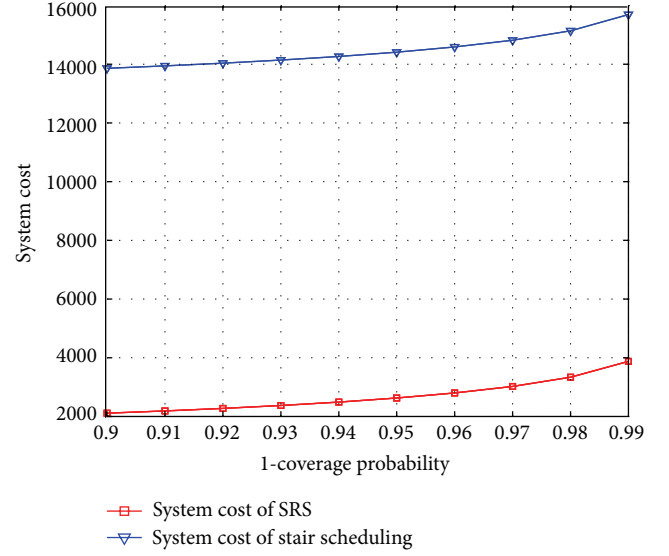


FIGURE 8: System cost as a function of 1-coverage probability for SRS, stair scheduling, and MRS.

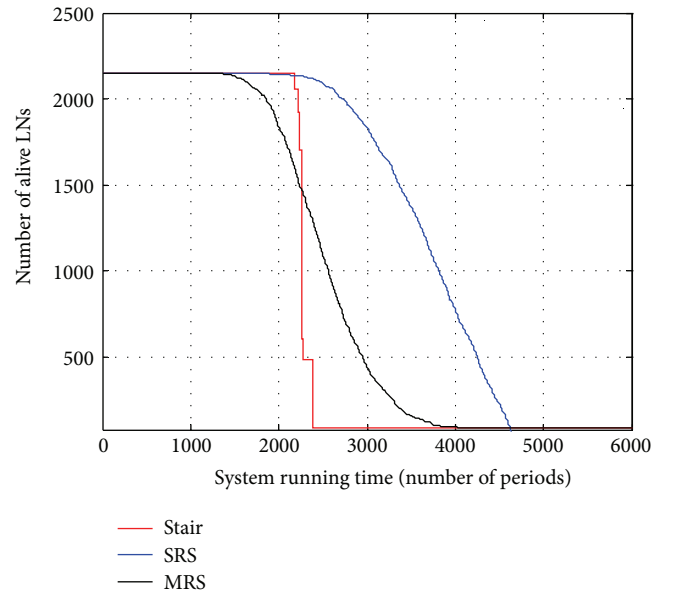


FIGURE 9: Number of alive LN sensors versus system running time.

### 5.2. Simulation Results

**5.2.1. System Cost.** We focus on the energy and communication reliability performances with the joint consideration of the system cost.

The system cost means the total cost of deployed LNs and CHs. We suppose that the cost of one CH node is 50 times of the cost of an LN. In SRS, the radius of the cluster is the same with the communication radius of LN. In MRS and stair scheduling, the radius of the cluster is fixed at 10 times of the average hop progress.

Figure 8 shows the influence of node density on the system cost performance. When the required 1-coverage



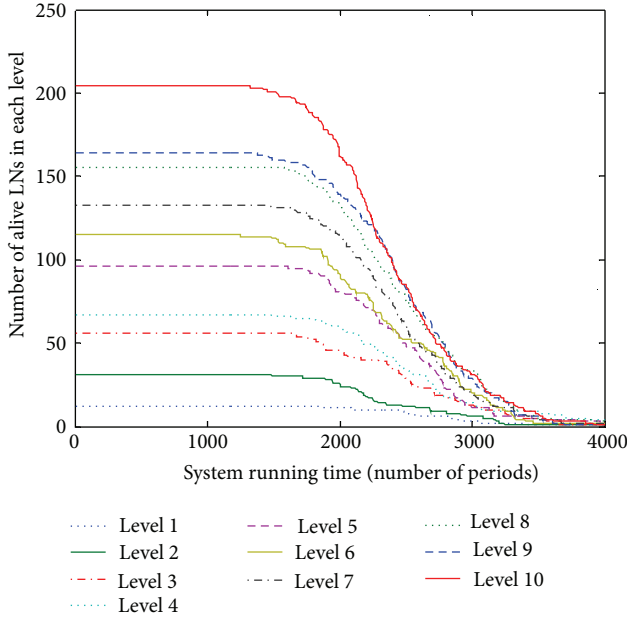


FIGURE 10: Alive LNs in each level versus system running time in MRS.

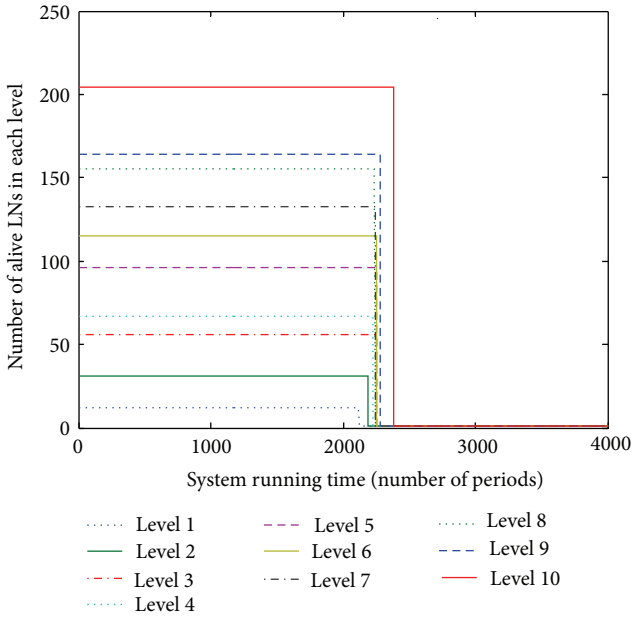


FIGURE 11: Alive LNs in each level versus system running time in stair scheduling.

probability varies from 0.9 to 0.99, the corresponding node density varies from  $0.73 \times 10^{-4}$  to  $0.1466 \times 10^{-3}$ . The corresponding system cost to cover the sensing field using SRS, stair scheduling, and MRS is plotted in Figure 8. We set the cost of one LN to one unit. We can see that the system cost of SRS is nearly seven times larger than the cost of stair scheduling and MRS. The cost of stair scheduling and MRS is the same since they use the same cluster structures.

**5.2.2. Energy Performance.** Figure 9 shows the number of alive LNs in the network as a function of system running time. The results show the following.

- (i) Basically, SRS has better LN lifetime performance. This is not a surprise because in SRS, each LN only senses and transmits local data. They do not relay other sensors message.
- (ii) The difference between SRS and stair scheduling and MRS is not very large. This owes to the average function hop-by-hop data aggregation method that we used in stair scheduling and MRS. With it, LNs in stair scheduling and MRS transfer a similar amount of data in each hop.

Figures 10 and 11 further show the number of alive LNs in each level of the cluster for MRS and stair scheduling. We can see the following

- (1) Basically, the energy is almost evenly consumed. Most LNs in stair scheduling and MRS can work for more than 2000 periods. In both MRS and stair scheduling, the LNs in the outmost level live the longest time and the LNs in the first level consume energy more quickly. This coincides with our analysis that LNs in the inner levels will have higher burden with message forwarding.
- (2) The curves of stair scheduling are much steeper than the curves of MRS. This means that LNs in different levels consume energy more evenly in stair scheduling than that in MRS. It will help LNs work together for longer time without performance degradation. The reason is due to the active slot control of the “stair scheduling.”

**5.2.3. Communication Reliability.** For communication reliability evaluation, to concentrate on the performance of transmission scheduling, we do not consider the path loss coefficient. In SRS, communication will fail only when more than one LN start data transmission to the same CH in the same time slot. In MRS, communication may fail due to two cases. (1) When a child node transmits data, its parent node is sleeping and is not aware of the transmission. (2) Multiple children transmit data to the same parent in the same time slot. In stair scheduling, communication will fail when more than one LN start data transmission to the same CH in the same time slot. Figure 12 compares the communication reliability of SRS, stair scheduling, and MRS in different levels. Cluster using SRS only has one level and each LN directly transmits to CH. We can see that the communication reliability is very high. Clusters using stair scheduling and MRS have multiple levels. The communication reliability of MRS is very low, because the child and parent are not jointly scheduled. The probability that a child and its parent are active together is very small. The reliability of stair scheduling is much better than that of MRS, because children and parents are jointly scheduled. When a child transmits data, his parent is active and is waiting for its transmission.

Figure 13 compares the multi-hop communication reliability. The communication reliability of stair scheduling is

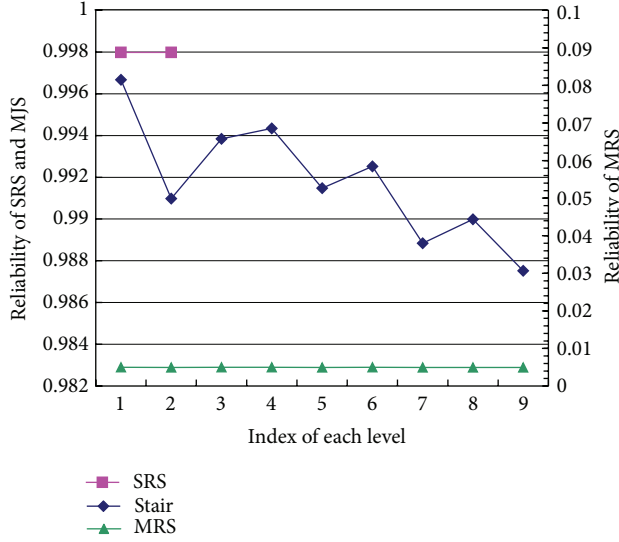


FIGURE 12: Communication reliability in each level.

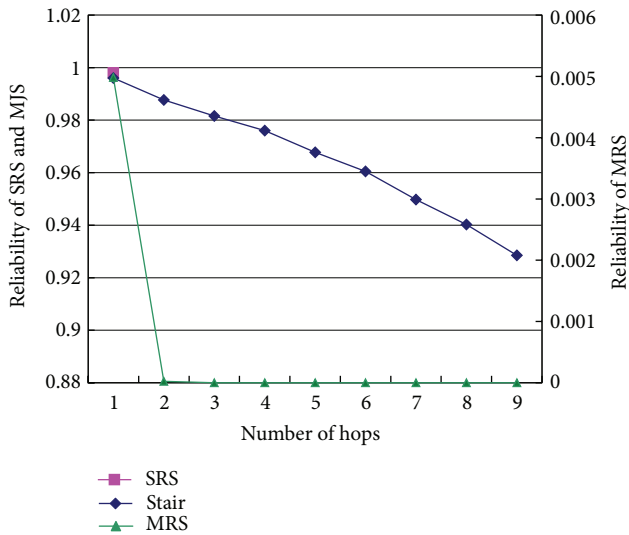


FIGURE 13: Multi-hop communication reliability versus number of hops.

still larger than 0.9 after nine hops. But the communication reliability of MRS is lower than 0.001 after only two hops. The results further confirm that joint scheduling is necessary in multi-hop clusters and stair scheduling provides satisfactory communication reliability.

**5.2.4. Performance Summary.** With the above simulation results, we can see that stair scheduling is more applicable for large-scale sensor networks than SRS and MRS. Although SRS has good energy and reliability performances, its main drawback is the high cost for deployment of large amount of CHs. The main drawback of MRS is the low reliability in multi-hop communication. The proposed stair scheduling on one hand has good energy performance and on the other

hand can provide satisfactory multi-hop communication reliability.

## 6. Conclusion

In this paper, we have proposed a stair scheduling method for data collection in tiered large-scale sensor network. The hop-by-hop time synchronization and data aggregation are proposed to make sensors in one cluster sleep and work cooperatively to conserve energy as well as to support multi-hop communication.

Particularly, in “Stair Scheduling,” Child node in the  $i$ th level always activates and sleeps one slot earlier than its parent, so that LNs in each level can sleep for most of the time and can forward data reliably. Hop-by-hop time synchronization is proposed to avoid time drifting and make the “Stair Scheduling” work in a fully distributed manner. Simulation results have verified that the proposed stair scheduling provides efficient energy and communication reliability performances, which makes it more applicable than MRS and SRS. In future work, we will study the link reliability by introducing enhanced mechanisms of communication collision avoidance and the cluster optimization methods.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 61202360, 61073174, and 61033001.

## References

- [1] Y. Wang and Y. Wang, *Distributed Storage and Parallel Processing in Large-Scale Wireless Sensor Networks*, vol. 20 of *High Performance Computing: From Grids and Clouds to Exascale*, IOS Press, 2011.
- [2] M. Medidi and Y. Zhou, “Extending lifetime with differential duty cycles in wireless sensor networks,” in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 1033–1037, November 2007.
- [3] M. Demirbas, K. Y. Chow, and C. S. Wan, “INSIGHT: internet-sensor integration for habitat monitoring,” in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '06)*, pp. 553–558, IEEE Computer Society, June 2006.
- [4] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, “Scheduling sleeping nodes in high density cluster-based sensor networks,” *Mobile Networks and Applications*, vol. 10, no. 6, pp. 825–835, 2005.
- [5] D. Tian and N. D. Georganas, “A coverage-preserving node scheduling scheme for large wireless sensor networks,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 32–41, ACM, September 2002.
- [6] X. Wu, J. Cho, B. dAuriol, and S. Lee, “Sleep nodes scheduling in Cluster-Based heterogeneous sensor networks using AHP,” in *Embedded Software and Systems*, Y. Lee, H. Kim, J. Kim, Y. Park, L. Yang, and S. Kim, Eds., vol. 4523 of *Lecture Notes in Computer Science*, pp. 437–444, Springer, Berlin, Germany, 2007.

- [7] J. Paek, B. Greenstein, O. Gnawali et al., "The tenet architecture for tiered sensor networks," *ACM Transactions on Sensor Networks*, vol. 6, no. 4, article 34, 2010.
- [8] Y. Wang, Y. Wang, H. Tan, and F. C. M. Lau, "Maximizing network lifetime online by localized probabilistic load balancing," in *Proceedings of the 10th International Conference on Ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW '11)*, pp. 332–345, Springer, 2011.
- [9] S. Bhattacharjee and N. Das, "Distributed data gathering scheduling in multihop wireless sensor networks for improved lifetime," in *Proceedings of the International Conference on Computing: Theory and Applications (ICCTA '07)*, pp. 46–50, IEEE Computer Society, March 2007.
- [10] Y. Wang, D. Wang, W. Fu, and D. P. Agrawal, "Hops-based sleep scheduling algorithm for enhancing lifetime of wireless sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS '06)*, pp. 709–714, October 2006.
- [11] W. Pak, J. G. Choi, and S. Bahk, "Tier based anycast to achieve maximum lifetime by duty cycle control in wireless sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '08)*, pp. 123–128, August 2008.
- [12] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos, "New results for Energy-Efficient broadcasting in wireless networks," in *Proceedings of the 13th International Symposium on Algorithms and Computation (ISAAC '02)*, pp. 332–343, Springer, 2002.
- [13] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "Robust multi-hop time synchronization in sensor networks," in *Proceedings of the International Conference on Wireless Networks (ICWN '04)*, pp. 454–460, June 2004.
- [14] S. Lindsey and C. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in *Proceedings of the IEEE Aerospace Conference Proceedings*, vol. 3, pp. 3-1125–3-1130, 2002.
- [15] S. Commuri and V. Tadigotla, "Dynamic data aggregation in wireless sensor networks," in *Proceedings of the IEEE 22nd International Symposium on Intelligent Control (ISIC '07)*, pp. 1–6, October 2007.
- [16] C. Wu and Y. Chung, "Heterogeneous wireless sensor network deployment and topology control based on irregular sensor model," in *Proceedings of the 2nd International Conference on Advances in Grid and Pervasive Computing (GPC '07)*, pp. 78–88, Springer, 2007.
- [17] L. Zhao, X. Hong, and Q. Liang, "Energy-efficient self-organization for wireless sensor networks: a fully distributed approach," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, pp. 2728–2732, December 2004.
- [18] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multicast in multi-hop wireless networks," in *Proceedings of the 1st International Conference on Wireless Internet (WICON '05)*, pp. 47–54, July 2005.
- [19] Y. Wang, D. Han, Q. Zhao, X. Guan, and D. Zheng, "Clusters partition and sensors configuration for target tracking in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Software and Systems (ICESS '05)*, pp. 333–338, Springer, 2005.