Extending k-Coverage Lifetime of Wireless Sensor Networks with Surplus Nodes

Ryo Katsuma†, Yoshihiro Murata‡, Naoki Shibata*, Keiichi Yasumoto†, and Minoru Ito†

†Nara Institute of Science and Technology
Ikoma, Nara 630-0192, Japan, {ryo-k,yasumoto,ito}@is.naist.jp
‡Hiroshima City University, Hiroshima 731-3194, Japan, yosihi-m@hiroshima-cu.ac.jp
* Shiga University, Hikone, Shiga 522-8522, Japan, shibata@biwako.shiga-u.ac.jp

Abstract

In the wireless sensor network (WSN) for periodically sensing and gathering environmental information uniformly in a vast sensing field, reducing the system operating cost taking into account sensor node installation and WSN lifetime is essential. In this paper, we propose a method to prolong the lifetime of such a data gathering WSN, by randomly scattering "surplus" sensor nodes over the target field. In our method, we assume that each sensor node has three operation modes: sensing, relaying, and sleeping. Each sensing node senses environmental data and sends/relays the data to the sink node via multi-hop wireless communication. Each relaying node just forwards the data received from its uplink node to its downlink nodes. Each sleeping node does nothing and keeps its battery. We propose an algorithm that dynamically changes mode of each sensor node so that the WSN lifetime becomes as long as possible by switching the least number of nodes for achieving k-coverage of the field to sensing mode. Through computer simulations, we confirmed that our method can prolong the WSN lifetime almost proportionally to the number of deployed sensor nodes.

Keywords: wireless sensor network, k-coverage, lifetime, data gathering

1 Introduction

Wireless Sensor Networks (WSNs) that consist of a massive number of small sensor nodes capable of wireless communication have been widely studied for such purposes as environmental monitoring, border guards, and so on. Among many types of WSNs, data gathering WSN periodically collects to a *sink node* temperature information and amount of sunlight at each point in a wide agricultural area or forest. Data gathering WSNs must cover the target sensing field and operate for a long term. Thus, many research efforts have been devoted to the sensor node deployment problem and the WSN lifetime extension problem.

For the sensor node deployment problem, in order to cover the target sensing field with less cost, it is very important to decide where and how sensor nodes should be deployed. Surveillance WSNs (e.g., border guards [1]) need sufficient sensing quality and robustness of the system, and thus they may require k-coverage[1] of the target sensing field. To achieve k-coverage, sensor node deployment must be carefully treated. In addition, even when the optimal sensor locations for the required coverage are decided, the accurate deployment of sensors might be difficult due to deployment cost and/or other restrictions (e.g., for a vast and/or dangerous field, it would be desirable to scatter sensor nodes from the air).

In [2], Poduri et al. used mobile sensor nodes to k-cover the target sensing field in short time under the constraint that for each sensor node, k other sensor nodes always exist in its proximity. They also discussed about the optimal locations of sensor nodes for k-covering the field. Wang et al. proposed a method to guarantee k-coverage of the target field at initial deployment time for an environment where mobile and static sensor nodes co-exist[3]. The above existing studies utilize mobile sensor nodes to guarantee k-coverage of the field, but they are not applicable to all types of WSN applications, since mobile sensor nodes are more expensive than static ones and they cannot move as expected when the field includes obstacles, muddy places, and so on.

For the WSN lifetime extension problem, we should consider the fact that each sensor node consumes battery amount for sensing and also for receiving/transmitting the sensed data from/to other sensor nodes. The battery lifetime varies depending on the initial amount, the data size for transmission, the sensing frequency, and the radio transmission distance.

In [4], Tang et al. proposed a method for reducing power consumption by regulating communication frequency among sensor nodes. In [5], Heinzelman et al. reduced total data transmission by unifying the data received from multiple sensors. However, since the above existing approaches degrade sensing quality, some applications that always need sufficient sensing frequency and quality may not accept such a quality degradation.

According to the above discussion, we need a new method coping with the sensor node deployment problem and the WSN lifetime extension problem at the same time satisfying the following constraints: (1) Only inexpensive static sensor nodes are used; (2) Sensor nodes are not always located at accurate locations; (3) Sensing quality is not degraded; and (4) Target sensing field is k-covered.

In this paper, we propose a WSN lifetime extension method satisfying the above constraints (1) to (4), where sufficient number of static sensor nodes are deployed over the field, e.g., by scattering them from the air, and the WSN lifetime is extended in proportional to the number of surplus nodes.

In our proposed method, we assume that each node has three modes: sensing, relaying, and sleeping, and it can change

[1] Any point in the target area is covered by at least k sensor nodes.

its mode anytime. Our target problem is to decide the schedules for switching operation modes of all nodes, the minimum set of sensing nodes which k-covers the field, and a data collection tree connecting all the sensing nodes at each time, so that the estimated WSN lifetime is maximized. We believe that obtaining the optimal solution to the problem is computationally difficult (it is presupposed to imply a Dominating Set Problem that is NP-Complete as a special case[6]).

In order to efficiently solve the above problem, we propose a heuristic algorithm called the *wakeup method*. The wakeup method finds the minimum number of nodes to k-cover the target sensing field and makes the remaining nodes sleep. This method initially supposes that all nodes are sleeping and then selects sensing nodes one by one in the order of the impact degree that the selected node has for k-coverage of the field. In general, power consumption for data communication depends on both data size and distance to the recipient node. So, we devised the *relay selection technique* which wakes up some of the sleeping nodes and utilizes the nodes as relay nodes for load distribution of heavy links in the data collection tree.

For evaluation of the proposed method, we conducted computer simulations for WSNs with several hundreds of sensor nodes, and compared the achieved WSN lifetime among our method and the conventional methods. As a result, we confirmed that the lifetime achieved by our method is much longer than other methods and our method can extend the lifetime almost in proportional to the number of surplus nodes randomly placed over the field.

2 Related Work

Most WSNs are expected to operate for a long time. However, the sensor nodes have poor computation power, and the battery amounts used for sensing and communication are also limited. MICA2 [7] is a typical sensor node with multiple sensors for sensing temperature, humidity, magnetism, and acceleration as well as wireless communication capability. AA cell batteries can be used in MICA2. Since each sensor node has a limited amount of battery, and its lifetime is also limited, many research efforts have studied the reduction of power consumption to extend the lifetime of the whole WSN.

Tang et al. extended the WSN lifetime by regulating the frequency of sending/receiving sensed data at each sensor node [4]. Sensor nodes with low battery power decreased the frequency of sending/receiving data to further reduce power consumption.

In [8], Cao, et al. proposed a sleep scheduling method which lets nodes sleep when they need not communicate, in order to save the overall power consumption in WSN. In this method, sleeping nodes consume small power, but do not communicate with other nodes, and become active after specified time interval. This method targets applications collecting events occurring rarely, thus it makes sleep schedules for all nodes that minimize the reception time at a sink node since an event occurred at a point, while letting as many sensor nodes as possible sleep.

The sensor node deployment problem for efficiently covering the target sensing field is another big topic. Here, a given geographical field is said to be *k-covered* if any point in the field is included in the sensing ranges of at least k sensor nodes. k-coverage is important, since in a WSN, information for the same place obtained by multiple sensor nodes is generally more accurate/effective than that from only one sensor node (e.g., a surveillance system with multiple cameras).

Wang et al. proposed a method to guarantee k-coverage [3] of the target field for an environment where mobile and static sensor nodes co-exist. In this method, sensor deployment must k-cover the entire field. This method conserves the moving power of mobile sensor nodes and deploys those mobile nodes to guarantee the k-coverage of the whole field, but it does not extend the WSN lifetime by reducing the power consumption for sending/receiving data.

Abrams et al. proposed set k-covering method to save power consumption for a WSN [9]. In this method, sensors are divided into several covers, and each cover acts iteratively in a round-robin fashion. They formulated the problem called set k-cover and proposed approximation algorithms. Also, they showed that the problem is NP-Complete and the proposed algorithms guarantee solutions better than $\frac{15}{16}$ of the optimal solution.

As discussed above, existing studies have tackled two big problems: WSN lifetime extension and sensor node deployment. However, to the best of our knowledge, no studies have achieved both coverage of the whole target field and maximization of the WSN lifetime under constraints (1)–(4) in Sect. 1.

3 Problem of WSN Lifetime Maximization

In this section, we present the WSN model and formulate the problem of maximizing WSN lifetime while k-covering the target field by deciding the schedules for operation modes of sensor nodes and constructing a data collection tree. The notations used in this paper are summarized in Table 1.

3.1 Model, Assumptions, and Definitions

(1) Assumptions on Target WSN

We suppose a WSN in which a massive number of small battery-driven sensor nodes are deployed in a *target field*. Sensor nodes periodically sense such environmental information as temperature, humidity, sunlight, or moving object, and send it by multi-hop communication to a base station called a *sink node*. We denote the target field, the sink node, and the sensing frequency as $Field$, Bs, and I, respectively. We denote the set of sensor nodes by $S = \{s_1, ..., s_l\}$.

Each sensor node has three operation modes: *sensing*, *relaying*, and *sleeping*. A node whose operation mode is sensing, relaying, or sleeping is called *sensing node*, *relaying node*, or *sleeping node*. We denote the sets of sensing, relaying, sleeping nodes by $U = \{u_1, u_2, ...\}$, $V = \{v_1, v_2, ...\}$, $W = \{w_1, w_2, ...\}$, respectively, where $U \cup V \cup W = S$. Each sensing/relaying node can change its mode instantly. Each sleeping node can change to another mode when it wakes up after a specified sleeping time elapses.

Table 1: Notations

Notation	Meaning
$Field$	the target sensing field
Bs and $Bs.pos$	sink node and its position
I	sensing frequency in Hz
D	size of data obtained by node
n	power attenuation co-efficient for antenna
$Trans(x, d)$	power required for a sensor node to transmit x bits for d meters
$Recep(x)$	power required for a sensor node to receipt x bits
$Sens()$	power required for a sensor node to sense
$Listen(y)$	power required for a sensor node to listen for y seconds
$Sleep(y)$	power required for a sensor node to sleep for y seconds
$C(s)$	energy consumption of sensor node s per seconds
R	sensing radius of each sensor
S	set of sensor nodes
U	set of sensing nodes
V	set of relaying nodes
W	set of sleeping nodes
$s.pos$, $s.energy[t]$, $s.range$, $s.desc$, and $s.send$	position, remaining battery amount at time t, sensing range, number of descendant nodes, and next hop node of a sensor node s
k	number of sensors to cover each point in target field.

Each sensing node covers a disk with radius R centered at the node. We denote the covered range of sensing node $s \in U$ by $s.range$. Each sensing node obtains data by sensing. We assume that the data size is fixed and the data are sent to the sink node without compression or unification along a multi-hop path (consisting of only sensing and relaying nodes) to the sink node. We denote the data size by D.

Each sensor node has a wireless communication capability and its radio transmission range is a disk with a certain radius centered on it. Sensing and relaying nodes can use the capability. Each sensing/relaying node can change its transmission power so that the radio transmission radius can be adjusted depending on the distance to the communicating node [2]. Since there is little influence on radio interference when sensing frequency I is small enough, we assume that there is no packet collision between nodes. A transmitted packet is always successfully received if the destination node (sensing/relaying node) is within the radio transmission range, and always fails if outside of the range. We assume that each node uses only one-hop unicast communication by designating a destination node.

We assume that each sensor node knows its position and sink node Bs is informed of positions of all nodes at their deployment time (e.g., with single-hop or multi-hop communication from each node to Bs). For each sensor node s, we denote its location by $s.pos$. Similarly, we denote the location of the sink node by $Bs.pos$. Based on the locations of all sensor nodes, the sink node calculates communication paths connecting all sensing/relaying nodes to the sink node and informs all sensing/relaying nodes of the new paths by single-hop or multi-hop flooding [3].

(2) Assumptions for Power Consumption

Each sensor node s has a battery, where the initial energy amount is denoted by e_{init} and the remaining energy amount at time t is denoted by $s.energy[t]$. Each node consumes energy for data transmission, data reception, and sensing data, and even during idle time and sleeping time.

Powers $Trans(x, d)$ and $Recep(x)$ required to transmit x[bit] for d[m] and receive x[bit] conform to formulas (1) and (2), respectively[5].

$$Trans(x, d) = E_{elec} \times x + \epsilon_{amp} \times x \times d^n \qquad (1)$$

$$Recep(x) = E_{elec} \times x \qquad (2)$$

Here, E_{elec} and ϵ_{amp} are constants representing the power required by information processing and the power for amplification, respectively. The value of $n(\geq 0)$ is defined by the antenna properties. If we use a strictly directed antenna in a vacuum, n is zero; if we use an omni-directional antenna, n is two. In reality, the value is somewhere between 0 and 2.

Powers $Sens()$, $Listen(y)$, and $Sleep(y)$ required to sense the information which is D[bit] data, listen to whether radio messages come or not for y [sec.], and sleep for y [sec.] conform to the following formulas (3), (4), and (5), respectively.

$$Sens() = E_{elec} \times D + E_{sens} \qquad (3)$$

$$Listen(y) = E_{listen} \times y \qquad (4)$$

$$Sleep(y) = E_{sleep} \times y \qquad (5)$$

Here, E_{sens}, E_{listen}, and E_{sleep} are constants representing the powers required for sensing data, listening for 1 second, and sleeping for 1 second, respectively.

[2]IRIS mote [10], for example, is capable of changing its transmission power from -17.2[dBm] to 3[dBm].

[3]For example, an architecture called BoostNet[11] allows Bs to broadcast critical information using large transmission range to reach all sensor nodes in one hop.

3.2 Problem Definition

3.2.1 Target Problem

If a particular set of sensing nodes are used for a long time, their batteries will be exhausted. Then, it is necessary to dynamically change the set of sensing nodes. So, we formulate a problem to derive the schedules of when and to which mode each sensor node should change at each time during WSN operation time. Fig. 1 shows an example of schedules.

Let t_0 and t_{end} denote the initial WSN deployment time and the time when the k-coverage of the WSN is no longer maintained due to battery exhaustion of some nodes. For each $s \in S$ and each $t \in [t_0, t_{end}]$, let $Mode(s,t)$ denote the operation mode of s at time t. Then, for each $s \in S$, we denote a *schedule* to switch the operation mode of s during time interval $[t_0, t_{end}]$ by the following formula.

$$schedule(s, [t_0, t_{end}]) = \bigcup_{t \in [t_0, t_{end}]} \{Mode(s,t)\}$$

Given the information on the target field $Field$, $s.pos$, $s.energy$, and $s.range$ for each sensor node $s \in S$, the position of a sink node $Bs.pos$, and constants E_{elec}, E_{sens}, E_{listen}, E_{sleep}, ϵ_{amp}, n, D, and I, our target problem for maximizing the WSN lifetime denoted by t_{life} is to decide the schedule $schedule(s, [t_0, t_{end}])$ for each node $s \in S$ that satisfies condition (6).

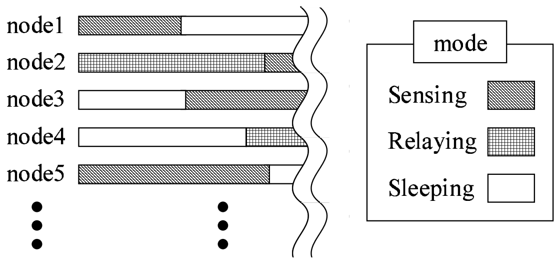


Figure 1: Schedule for Switching Operation Mode

$$\forall t \in [t_0, t_{end}],]\forall pos \in Field, |Cover(pos,t)| \geq k. \quad (6)$$

where

$$Cover(pos,t) \stackrel{def}{=} \{s | pos \in s.range \wedge Mode(s,t) = sensing \wedge s.energy[t] > 0\}. \quad (7)$$

The condition (6) guarantees the k-coverage of the target field. In general, k-coverage can be achieved by a part of all sensor nodes ($U \subseteq S$) whose remaining energy amounts are not exhausted.

We define the WSN lifetime t_{life} as the time from initial deployment to the time when the condition (6) is no longer satisfied for any set of sensing nodes.

Then, we define the following objective function (8):

$$\textbf{maximize } (t_{life}) \textbf{ subject to } (6) \quad (8)$$

3.2.2 Modified Target Problem

Our target problem consists of the following three sub-problems.

The first sub-problem is to decide the set of sensing nodes for maximizing t_{life} and satisfying condition (6). Since sensing nodes periodically carry out sensing operation they consume more energy than relaying and sleeping nodes. This problem is presupposed to imply a Dominating Set Problem (DS) that is NP-Complete as a special case[6].

The second sub-problem is to decide the set of relaying nodes for maximizing t_{life}, when the set of sensing nodes are given. Some remaining nodes can reduce critical nodes' transmission distance and transmission data amount so that the overall WSN lifetime is extended.

The third sub-problem is to decide the data collection tree for maximizing t_{life}, when the sets of sensing and relaying nodes are given. It is required to balance the energy consumption among all sensor nodes in the tree. Because a node near the sink node tends to consume more battery by forwarding the data transmitted from other nodes to the sink node.

Since the above problems are dependent on each other in maximizing the WSN lifetime, solving these problems at the same time is considered to be very difficult. Therefore, we adopt a heuristic that solves these problems stepping on the following stages.

(1) Solving the problem to find the minimum set of U satisfying the condition (6).

(2) Solving the problem to find a data collection tree that is rooted on sink node Bs and include all sensing nodes U and some relaying nodes $V \subseteq S - U$ for maximizing *the WSN forecast lifetime*.

(3) Sleeping nodes $W = S - U - V$ are set for a sleeping duration based on the *next battery exhaustion time*.

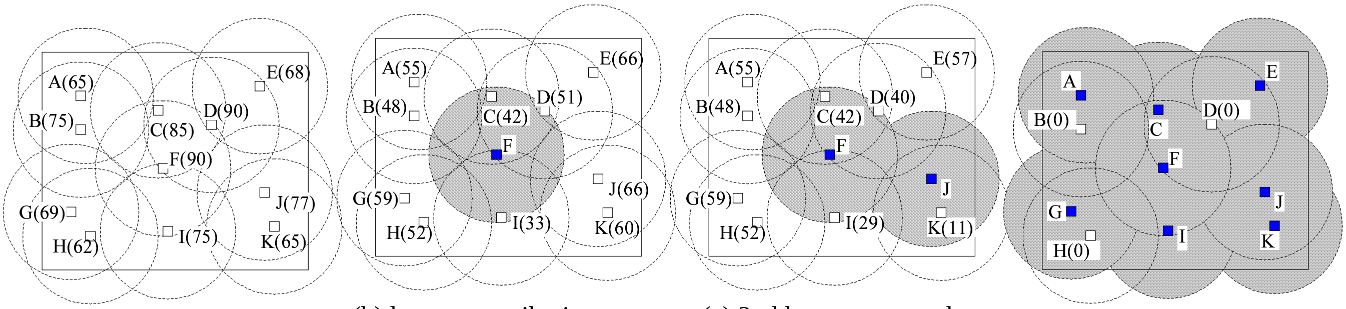
(4) At next battery exhaustion time, the stages (1), (2), and (3) are executed.

In the above stage (2), the WSN forecast lifetime is the approximated WSN lifetime without considering the changes of the mode of each sensor node in the future. We define the WSN forecast lifetime as follows:

$$t_{now} + \min_{pos \in Field} \left(\frac{\Sigma_{s \in Cover(pos, t_{now})}(s.energy[t_{now}])}{\Sigma_{s \in Cover(pos, t_{now})}(C(s))} \right) \quad (9)$$

where, t_{now} is current time, and $C(s)$ is the energy consumption of sensor node s per second, The WSN forecast lifetime is the earliest time when some point in the field is no longer k-covered due to battery exhaustion of some nodes.

Before sleeping nodes sleep, they must be set for the time to wake up. The modes of all sensor nodes are recalculated and informed to them by Bs when the battery of any sensor node is exhausted. When listening to the information of the next mode from Bs, sleeping nodes should be waking up. Therefore, the earliest time when the battery of some sensor



(a) initial state

(b) largest contribution area node selected

(c) 2nd largest area node selected

(d) resulting state

Figure 2: Example of Applying Wakeup Method

node is exhausted (called the *next battery exhaustion time*) is set as the time to wake sleeping nodes up. We define then next battery exhaustion time as follows:

$$t_{now} + \min_{s \in S} \left(\frac{s.energy[t_{now}]}{C(s)} \right) \quad (10)$$

where $\frac{s.energy[t]}{C(s)}$ is the time duration that the remaining battery amount of sensor node s at time t is exhausted.

The energy consumption of sensor node s per unit of time ($C(s)$) is as follows:

For each sensor node $s \in U$,

$$C(s) = I \times (Sens() + Recep(D \times s.desc)$$
$$+Trans(D \times (s.desc + 1), Dist(s, s.send)) + Listen(1) \quad (11)$$

For each sensor node $s \in V$,

$$C(s) = I \times (Recep(D \times s.desc)$$
$$+Trans(D \times (s.desc), Dist(s, s.send))) + Listen(1) \quad (12)$$

For each sensor node $s \in W$,

$$C(s) = Sleep(1) \quad (13)$$

where $s.desc$ is the number of sensing nodes except for s in the subtree of the data collection tree rooted on s, and $s.send$ is the destination node of data transmission by s, and $Dist(s_1, s_2)$ is the distance from s_1 to s_2.

4 Algorithm

In this section, we describe an algorithm to solve the modified target problem defined in Section 3.

4.1 Overview

Our algorithm finds operation modes for sensor nodes and a tree called a *data collection tree* that connects all sensing and relaying nodes to sink node Bs by multi-hop paths for data collection.

The algorithm is supposed to be executed at the initial deployment time and each of the next battery exhaustion time. The lifetime of the whole system ends when there are no sets of sensing nodes that satisfy condition (6).

Our algorithm consists of the following three methods: (1) Wakeup method, (2) Relay selection method, and (3) Mode switching method.

4.2 Wakeup Method

Wakeup method finds the minimum number of sensing nodes to k-cover the target field, by letting the more influential nodes to be sensing nodes one by one. We show the algorithm of Wakeup method below. Note that the sink node executes it to just derive the set of sensing nodes, and does not change nodes' actual operation modes.

1. First, all sensor nodes are regarded as sleeping nodes.
2. For each sleeping node, the area called *contribution area* that is not k-covered but included in its sensing range is calculated.
3. Select the node which has the largest contribution area as a sensing node. If there are more than one such nodes, one of those nodes is randomly selected and selected as a sensing node.
4. If there is no sleeping sensor nodes remaining, the algorithm terminates with no solution.
5. If the whole target field is k-covered, the algorithm terminates with the selected set of sensing nodes as a solution. Otherwise, go to Step 2.

We now show an example of finding the nodes to 1-cover the target field. Fig. 2 shows how the sensing nodes are selected by the Wakeup method. In the figure, the squares are sensor nodes, and dotted circles are the sensing ranges of sensor nodes. Each label like 'A(65)' represents the sensor node id 'A' and the contribution area size '65'. Fig. 2(b) shows the result after the first iteration of the algorithm. By selecting sensor node F as a sensing node, the corresponding contribution area has been 1-covered (gray circle in Fig. 2(b)). Then the algorithm is applied to other sensor nodes. Fig. 2(c) shows the result after the second iteration of the algorithm. In this case, nodes E and J have the same largest contribution area size 66, thus node J has been randomly chosen to be a sensing node. Fig. 2(d) is the result after the algorithm terminates with a solution.

Table 2: Simulation Configurations

Parameter	Value
Initial energy amount of each node	$s.energy = 32400$J (by referring to [5])
Power consumption coefficient for data processing	$E_{elec} = 50$ nJ/bit (by referring to [14])
Power consumption coefficient for signal amplification	$\epsilon_{amp} = 100$pJ/bit/m^2 (by referring to [14])
Power consumption coefficient for sensing	$E_{sens} = 0.018$J (by referring to [7])
Power consumption coefficient for idle time	$E_{listen} = 0.043$J/s (by referring to [7])
Power consumption coefficient for sleep time	$E_{sleep} = 0.000054$J/s (by referring to [7])
Power consumption exponent	$n = 2$ (by referring to [14])
Area of sensing disk of each sensor	$range = 20$m^2 (by referring to [15])
Degree of coverage in the target field	k = 1,2,and 3
Size of data for sensed information	$D = 128$bit (by referring to [16])
Sensing frequency	0.1Hz (by referring to [16])
Maximum radio transmission distance	300m (by referring to [10])

4.3 Relay Selection Method

The data size and the communication distance have large impact on energy consumption for data communication. We use the *Balanced edge selection method* proposed in [12] to balance transmitted data amount among all nodes. In order to reduce the communication distance, we propose Relay selection method.

In Relay selection method, the tree generated by Balanced edge selection method is modified to improve WSN lifetime by utilizing relay nodes. There are areas with shorter lifetime although the area is k-covered because of non-uniform node density. In some cases, the communication energy can be saved by relaying communication. The proposed relay selection algorithm is shown as follows.

Suppose that there is a link between sensor nodes $s_1 \in U \cup V$ and $s_2 \in U \cup V$. We choose a sleeping or relaying node $s_{relay} \in V \cup W$ such that distance between s_1 and s_{relay} is shorter than that between s_1 and s_2. By making s_{relay} relay the communication between the two nodes, the communication power can be reduced. If this change worsens the value of the objective function, the change is discarded. s_{relay} investigates all sleeping and relaying nodes in the ascending order of distance from s_1. This operation is performed to all links including the new links.

4.4 Mode Switching Method

This section describes how and when the operation mode of each sensor node is changed. The algorithm for switching operation modes of all sensor nodes is shown as follows:

1. After the initial deployment of sensor nodes, Bs decides the sets of sensing, relaying, and sleeping nodes and the data collection tree by Wakeup method, Balanced edge selection method, and Relay selection method.

2. Bs calculates the sleeping time of all sleeping nodes by formula (10).

3. Bs informs the information to all sensor nodes by single-hop or multi-hop flooding, that is the mode of each sensor node, the data collection tree, and next battery exhaustion time.

4. Each sensor node switches to the specified mode and sets the destination node..

5. WSN operates, and the energy of each sensor node is reduced as time passes.

6. At next battery exhaustion time, sleeping nodes wake up and prepare for listening the information from Bs.

7. The above steps 1 to 6 are repeated during the WSN lifetime.

5 Experimental Validation

In order to evaluate the overall performance of our proposed method, we have conducted computer simulations for measuring the WSN operation time during which the whole target field is k-covered (we call the time *k-coverage lifetime*, hereafter), and compared the k-coverage lifetime with other conventional methods, for several experimental configurations.

As a common configuration among the experiments, we used the parameter values shown in Table 2 by referring to existing literature.

For all the experiments, we used a WSN simulator which we implemented in Java and executed the simulator on a PC with Intel Core2Duo E6600 (2.4GHz), 1GB memory, WindowsXP Professional, and Sun Java Runtime Environment 1.6.0_02.

We have measured k-coverage lifetime among our proposed method and several other conventional methods named as follows: (i) *Proposed Method* which uses all techniques in Section 4; (ii) *Balanced Edge Only* which is the method same as the Proposed Method without Relay selection method; (iii) *Dijkstra* which is the method using a minimum spanning tree

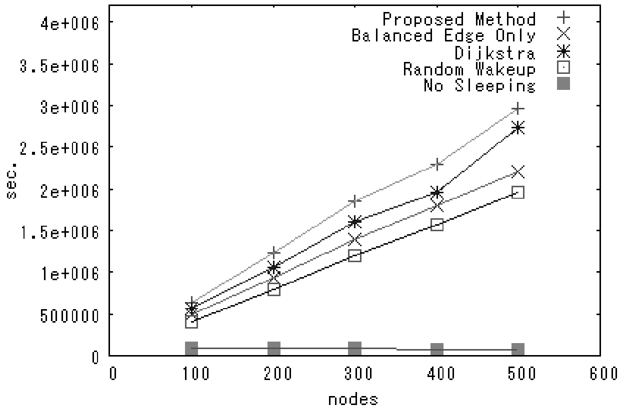


Figure 3: 1-Coverage Lifetime

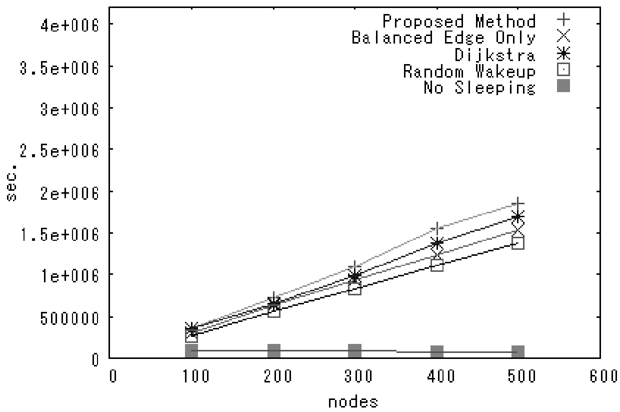


Figure 4: 2-Coverage Lifetime

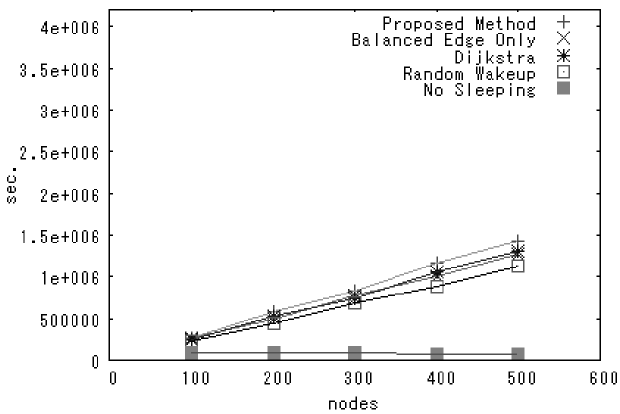


Figure 5: 3-Coverage Lifetime

instead of a data collection tree generated by Balanced edge selection method in Proposed Method; (iv) *Random Wakeup* which is the method using random selection to find a minimal set of sensing nodes for k-coverage instead of Wakeup Method in Proposed Method; and (v) *No Sleeping* which is the method letting all nodes to be sensing nodes and gathering sensed data from all nodes to the sink node.

For the above conventional algorithm (iii) , we constructed minimum cost spanning trees by Dijkstra method [13] as data collection trees, where cost of each edge is the square of the distance. For the conventional algorithm (iv), we show the detail of Random wakeup method below:

1. First, all sensor nodes are set to sleep mode.
2. A sleeping sensor node is selected randomly, if its sensing range includes the area that is not k-covered, it is set to a sensing node.
3. If there is no sleeping sensor nodes remaining, the algorithm terminates.
4. If the whole target field is k-covered, the algorithm terminates. Otherwise, go to Step 2.

The difference from Wakeup method is the way of node selection in the above step 2. Random wakeup method selects a sleeping node randomly, and if the sensing area of the node includes the area which is not k-covered, its mode is changed to sensing mode. On the other hand, Wakeup method sequentially selects a sleeping node whose sensing area covers the widest area which is not k-covered, and changes its mode to sensing mode.

The configuration of this experiment other than Table 2 is provided as follows.

- Field size: 50m $\times$ 50m
- Position of the sink node: around the south (bottom) end in the field
- Number of sensor nodes: 100, 200, 300, 400, and 500
- Coverage degree: k=1, 2, and 3

Note that the size of the target field should be appropriately decided so that the field can be sufficiently k-covered for a given number of nodes and coverage degree k. Thus, we used field size 50m $\times$ 50m, that is, when 100 sensing nodes are randomly deployed in the target field, there will be extremely surplus nodes for k=1, 2, and 3. In the experiment, the initial positions of nodes are given in the target field by uniform random variables.

We show experimental results obtained through computer simulations in Fig. 3 for 1-coverage, Fig. 4 for 2-coverage, and Fig. 5 for 3-coverage. These results are average of 40 trials.

Figs. 3, 4, and 5 show that Proposed Method, Balanced Edge Only, Dijkstra, and Random Wakeup outperform No Sleeping to a great extent, independently of k and the number of nodes. The reason is that these four methods were able to use the sleep mode well, and reduce the power consumption on idle time of some sensor nodes. The figures also show that Proposed Method achieves better performance than Balanced Edge Only. This is an evidence that our proposed Relay Selection Method is effective to extend the k-coverage lifetime. The figures also show that Proposed Method achieves better performance than Dijkstra. This is an evidence that our proposed balanced edge selection algorithm is effective to extend the k-coverage lifetime. The figures also show that Proposed Method achieves better performance than Random Wakeup Method. This is an evidence that to select sequentially a node which is the most effective to the k-coverage node guarantees longer k-coverage lifetime than to select randomly.

In these figures, all methods except for No Sleeping extended k-coverage lifetime almost proportionally to the number of surplus nodes. The reason is that until sensing nodes exhaust their battery, surplus nodes are able to keep their battery by sleeping.

In the No Sleeping, we see that the k-coverage lifetime of all methods decrease as the number of nodes increases. The reason is that the nodes that directly connects to the sink node Bs have to forward more data transmitted from their upstream nodes as the number of nodes increases. We see in the figures that the k-coverage lifetime decreases gradually as k increases. This is because more nodes are required to achieve k-coverage of the field as k increases.

We also confirmed that our proposed algorithm (decision of sensing nodes and construction of a data collection tree) takes reasonably short calculation time. In these experiments, maximum calculation time of the proposed algorithm was 1.2 seconds when the number of nodes is 500.

6 Conclusion

In this paper, we formulated a problem that maximizes the k-coverage lifetime of the data gathering WSN using surplus static sensor nodes with sleeping mode. We proposed Wakeup method to decide the modes of sensor nodes, and Relay selection method to modify the data collection tree which includes sensing and relay nodes.

As a result, we confirmed that our method improved k-coverage lifetime to a great extent compared with other conventional methods for several hundreds of sensor nodes in practical time.

REFERENCES

[1] Chen, A., Kumar, S., and Lai, H. T.: "Designing Localized Algorithms for Barrier Coverage," *Proc. of 13th Int'l Conf. on Mobile Computing and Networking (MobiCom 2007)*, pp. 63–74 (2007).

[2] Poduri, S., Sukhatme, G.S.: "Constrained coverage for mobile sensor networks," *Proc. of Int'l Conf. on Robotics and Automation (ICRA2005)*, pp. 165–171 (2005).

[3] Wang, W., Srinivasan, V., and Chua, K. C.: "Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks," *Proc. of 13th Int'l. Conf. on Mobile Computing and Networking (MobiCom 2007)*, pp. 39–50 (2007).

[4] Tang, X. and Xu, J.: "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," *Proc. of Infocom2006*, pp. 1–12 (2006).

[5] Heinzelman, W.R., Chandrakasan, A., and Balakrishnan, H.: "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. of the 33rd Hawaii Int'l. Conf. on System Sciences (HICSS 2000)* 2000.

[6] Yang, S., Cardei, M., Wu, J., and Patterson, F.: "On Connected Multiple Point Coverage in Wireless Sensor Networks," *Proc. of Int'l Journal of Wireless Information Networks*, Vol.13, No.4, pp. 289–301 (2006).

[7] Crossbow Technology, Inc. 2003: "MICA2: Wireless Measurement System," http://www.xbow.com/Products/Product_pdf_files/ Wireless_pdf/MICA.pdf.

[8] Cao, Q., Abdelzaher, T., He, T., Stankovic, J.: "Towards Optimal Sleep Scheduling in Sensor Networks for Rare-Event Detection," *Proc. of Information Processing in Sensor Networks (IPSN2005)*, pp. 20–27 (2005).

[9] Abrams, Z., Goel, A. and Plotkin, S.: "Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks," *Proc. of 3rd Int'l. Symp. on Information Processing in Sensor Networks*, pp. 424–432 (2004).

[10] Crossbow Technology, Inc. 2008: "IRIS mote," http://www.xbow.jp/mprmib.pdf.

[11] Liu, S., Fan, K.W., and Sinha, P.: "Dynamic Sleep Scheduling using Online Experimentation for Wireless Sensor Networks," *Proc. of Int'l. Workshop on Measurement, Modeling and Performance Analysis of Wireless Sensor Networks (SenMetrics2005)* (2005).

[12] Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito, M.: "Extending k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes," *Proc. of 5th Int'l. Conf on Wireless and Mobile Computing, Networking and Communications (WiMob'2009)*, pp. 48–54 (2009).

[13] Dijkstra, E.W.: "A Note on two Problems in Connection with Graphs," *Numerische Mathematik* (1959).

[14] Wang, G., Cao, G., La Porta, T., and Zhang, W.: "Sensor Relocation in Mobile Sensor Networks," *Proc. of Infocom 2005*, Vol. 4, pp. 2302–2312 (2005).

[15] Ganeriwal, S., Kansal, A., and Srivastava, M. B.: "Self aware actuation for fault repair in sensor networks," *Proc. of Int'l Conf. on Robotics and Automation (ICRA2004)* (2004).

[16] Kamimura, J., Wakamiya, N., and Murata, M.: "Energy-Efficient Clustering Method for Data Gathering in Sensor Networks," *Proc. of First Workshop on Broadband Advanced Sensor Networks (BaseNets2004)* (2004).