NAIST-IS-DD0861003

Doctoral Dissertation

A Study on Maximizing k-coverage Lifetime of Wireless Sensor Network

Ryo Katsuma

February 20, 2011

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Ryo Katsuma

Thesis Committee:

Professor Minoru Ito	(Supervisor)
Professor Minoru Okada	(Co-supervisor)
Associate Professor Keiichi Yasumoto	(Co-supervisor)

A Study on Maximizing k-coverage Lifetime of Wireless Sensor Network*

Ryo Katsuma

Abstract

In data gathering wireless sensor networks (WSNs), sensor nodes deployed over a target monitoring field periodically sense environmental information and send the information to a sink through multi-hop wireless communication. WSNs, in general, require a coverage of a target field by sensors and the long operation time. For sensing accuracy, some WSNs require k-coverage (any point in the target area is covered by at least k sensor nodes) of a field. There have been many studies for efficiently k-covering the field or extending lifetime in WSNs. However, there has been few study to maximize the WSN operation time during which the whole target field is k-covered (k-coverage lifetime). In this thesis, we provide three different approaches to maximize k-coverage lifetime.

In the first approach, we define a k-coverage lifetime maximization problem for WSNs consisting of both static and mobile sensor nodes sparsely deployed in the field. This problem is NP-hard since it implies the Minimum Geometric Disk Cover Problem as a special case. To quickly derive an approximate solution, we propose a genetic algorithm (GA) based method to find near optimal positions of mobile nodes and a data collection tree in practical time. To mitigate the problem that nodes near the sink consume a lot of energy for forwarding the data from upstream nodes, we construct a tree where the amount of communication traffic is balanced among all nodes and use it as an initial solution of GA. Through simulations, we confirmed that the k-coverage lifetime of our method is about 40% to 90% longer than conventional methods for 100 to 300 nodes WSNs.

* Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0861003, February 20, 2011.

In the second approach, we define a k-coverage lifetime maximization problem as a sleep scheduling problem of sensor nodes for dense WSNs where only static sensor nodes are densely deployed. In this problem, we assume that each sensor node has three operation modes: sensing, relaying, and sleeping. This problem is NP-hard since it implies the Dominating Set Problem as a special case. To quickly derive an approximate solution, we propose a heuristic method called *Single tree method*. In this method, we activate the minimal number of nodes required for k-coverage at each time. To do so, the sink initially regards all nodes as sleep nodes and selects sensing nodes one by one in the order of the impact degree the selected node has for k-coverage of the field. Then, the sink sends this calculation result to each node so that unselected nodes will sleep. The node that exhausted battery is replaced by another one at the next calculation. Through simulations, we confirmed that the proposed method achieves 1.1 to 1.7 times longer k-coverage lifetime than the other methods for 100 to 500 node WSNs.

In the third approach, we tackle a robustness problem such that extreme coverage loss occurs due to failure of a node near the sink. To cope with this problem, we propose a method called *Layer method* that achieves k-coverage lifetime maximization and robustness to node failure. This method groups all nodes into sets of nodes called *layers* such that each layer 1-covers the field, constructs a data collection tree for each layer, and activates only k layers at each WSN operation time. Through simulations, we confirmed that Layer method always keeps $k - 1$ or more coverage degree when one node failure occurs and the lifetime with Layer method is only 5% less than Single tree method for 600 to 1000 nodes WSNs.

Keywords:

wireless sensor network (WSN), lifetime maximization, k-coverage, mobile sensor node, sleep scheduling

Contents

List of Figures

List of Tables

1. Introduction

Wireless sensor networks (WSNs) are networks consisting of many small sensor nodes capable of wireless communication, and they are used for environmental monitoring [1] [2], border guards [3] [4], and so on. Among many types of WSNs, the data gathering WSN periodically collects to a *sink* environmental information such as temperature and amount of sunlight at each point in a wide agricultural area or forest. Some data gathering WSN applications need sufficient sensing quality and robustness of the system, and such systems may require k-coverage[1] of the target sensing field. Data gathering WSNs also require the long term operation. So far, many research efforts have been devoted to the k-coverage problem and the WSN lifetime extension problem.

In order to k-cover the field, Poduri et al. used mobile sensor nodes to k-cover the target sensing field in short time utilizing the constraint that for each sensor node, k other sensor nodes always exist in its proximity [6]. They also discussed about the optimal locations of sensor nodes for k-covering the field. Wang et al. extended the WSN lifetime with mobile sensor nodes [7]. When battery exhaustion and/or failure of a node occurs, other mobile sensor nodes move and quickly replace the failed one. Mei et al. proposed a method using mobile sensor nodes in addition to static sensor nodes, so that the mobile nodes repair broken communication paths due to node failure [8]. These methods do not consider maintaining k-coverage of the field for a long time though it can repair k-coverage in short time.

In order to make WSNs operate for a long term, there are many studies to reduce energy consumption. The energy consumption of a node depends on the communication frequency, the transmitted data amount, the time for radio listening, and so on. Tang et al. proposed a method for reducing energy consumption by regulating communication frequency among sensor nodes [9]. Heinzelman et al. proposed a method for reducing total data transmission amounts by merging the data received from multiple sensor nodes [10]. However, since the above existing approaches may deteriorate sensing quality with respect to spacial and temporal density of sensing data, some applications that always need sufficient sensing quality may not accept such a quality deterioration.

Another approach to reduce energy consumption in WSNs is the sleep scheduling

[1] Any point in the target area is covered by at least k sensor nodes [5].

of sensor nodes. Cao et al. proposed a sleep scheduling method which lets nodes sleep when they do not need communication, in order to save the overall energy consumption in WSNs [11]. Keshavarzian et al. proposed a method to minimize the number of active nodes and guarantee that the event information sensed by sensor nodes is delivered to the sink in a specified time [12]. In these methods, sleeping nodes consume little power and become active after a specified time interval. These existing methods target applications for collecting events that rarely occur and do not consider maintaining the field k-coverage for a long time.

As discussed above, there is few study that maximizes the operation time (called *k-coverage lifetime*, hereafter) of the data gathering WSN during which the whole target field is k-covered.

In this thesis, we provide three different approaches to maximize k-coverage lifetime. One of the approaches uses mobility of mobile sensor nodes [13] [14], and others use sleep mode of static sensor nodes [15] [16] [17].

In the first approach, we define a k-coverage lifetime maximization problem for WSNs consisting of both static and mobile sensor nodes sparsely deployed in the field. The target problem is to decide a moving schedule of mobile nodes (when and to which direction each mobile node should move at each time during WSN operation time) and a tree spanning all sensor nodes for data collection (we use a tree as data communication paths). This problem is NP-hard since it implies the Minimum Geometric Disk Cover Problem [18] as a special case. So, we propose a genetic algorithm (GA [19]) based scheme to find a near optimal solution in practical time. In order to speed up the calculation, we devised a method to check a sufficient condition of k-coverage of the field. To mitigate the problem that nodes near the sink consume a lot of energy for forwarding the data from upstream nodes, we construct a tree where the amount of communication traffic is balanced among all nodes, and add this tree to the initial candidate solutions of our GA-based algorithm. Through the simulations, we confirmed that our method achieves k-coverage lifetime about 40% to 90% longer than conventional methods for 100 to 300 node WSNs.

In the second approach, to maximize k-coverage lifetime of a WSN with densely deployed static sensor nodes, we define a problem to decide sleep schedules of all nodes and a data collection tree called *sleep scheduling problem*. In this problem, 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 node. Each sleeping node does nothing and keeps its battery. Deriving the optimal solution to the problem is NP-hard since it implies the Dominating Set Problem [27] as a special case. To quickly derive an approximate solution, we propose a method called *Single tree method* to solve the sleep scheduling problem by activating the minimal number of the nodes required for k-coverage, and replacing the node that exhausted battery by another one. This method chooses active nodes one by one in the order of the impact degree the selected node has for k-coverage of the field. In order to evaluate the effectiveness of our algorithm in terms of k-coverage lifetime, we compared our method with methods in which some of the proposed features are disabled. Through simulation-based comparison, we confirmed that the proposed method achieves 1.1 to 1.7 times longer k-coverage lifetime regardless of k and the number of nodes, than the other methods for 100 to 500 node WSNs.

In the third approach, we propose a method called *Layer method* to solve both the sleep scheduling problem and the robustness problem. The robustness problem means that the sink may not be able to collect sensing data from many nodes when a node belonging to the tree near the sink node fails. This method groups all nodes into sets of nodes called *layers* such that each layer 1-covers the field, constructs a data collection tree for each layer, and makes only k layers active in a time division manner. Through computer simulations, we confirmed that Layer method always keeps $k - 1$ or more coverage degree if one node fails and the lifetime with Layer method is only 5% less than Single tree method.

In Chapter 2, we describe the common WSN model, assumptions, and definitions used for the proposed methods. In Chapter 3, we target the k-coverage lifetime maximization problem for WSNs consisting of mobile and static sensor nodes. In Chapter 4, we solve the k-coverage lifetime maximization problem for WSNs where static sensor nodes are densely deployed. In Chapter 5, we introduce Layer method that enhances robustness to the problem in Chapter 4. In Chapter 6, we give conclusions learnt from the study in this thesis.

2. Common Wireless Sensor Network Model: Assumptions and Definitions

In this chapter, we present the common WSN model, assumptions, and definitions used throughout this thesis. Assumptions and definitions specific for each proposed method will be described later.

2.1 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 image, and send it by multi-hop wireless 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 $S = \{s_1, ..., s_l\}$.

Sensor nodes have 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 $U \cup V \cup W = S$. Once a node changes its mode to the sleeping mode, it does not wake up until the specified sleeping time elapses. Sleeping nodes can change their modes upon wakeup. Sensing nodes and relaying nodes can change their modes instantly.

A sensing node collects environmental data from 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 to the sink node. We use a tree connecting all sensing and relaying nodes to the sink node as communication paths called *data collection tree*. We denote the sensing data size by D.

Each sensing/relaying node has a wireless communication capability and its radio transmission range is a disk with a specified radius centered on it. Each node can change its transmission power to change the communication distance. 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 the node is 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 $Bs.pos$. The sink node conducts the centralized calculation and informs the solution to all nodes by single-hop or multi-hop flooding.

2.2 Assumptions for Energy Consumption

Each sensor node s has a battery, where the initial energy amount and the remaining energy amount at time t are denoted by e_{init} and $s.energy[t]$, respectively. 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 [10].

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

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

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.

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

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

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

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

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.

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

For each sensing node $s \in U$,

$$\begin{aligned} C(s) = I \times (Sens() + Recep(D \times s.desc) + \\ Trans(D \times (s.desc + 1), Dist(s, s.send)) + Listen(1) \end{aligned} \tag{6}$$

For each relaying node $s \in V$,

$$\begin{aligned} C(s) = I \times (Recep(D \times s.desc) \\ + Trans(D \times (s.desc), Dist(s, s.send))) + Listen(1) \end{aligned} \tag{7}$$

For each sleeping node $s \in W$,

$$C(s) = Sleep(1) \tag{8}$$

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

2.3 Definition of k-coverage

We define k-coverage as follows:

$$\forall t \in [t_0, t_{end}], \forall pos \in Field, |Cover(pos, t)| \geq k \tag{9}$$

where

$$Cover(pos, t) \stackrel{def}{=} \{s | pos \in s.range \wedge Mode(s, t) = sensing \wedge s.energy[t] > 0\} \tag{10}$$

The condition (9) guarantees the k-coverage of the target field. In general, k-coverage can be achieved by a subset U of S ($U \subseteq S$).

We define the k-coverage lifetime t_{life} of a WSN as the time since the initial deployment to the time when condition (9) cannot be satisfied by the remaining sensor nodes. Our objective is to maximize t_{life}.

3. k-coverage Lifetime Maximization Method Using Mobile Nodes

In this chapter, we formulate the problem to maximize the k-coverage lifetime by using mobile sensor nodes, propose an algorithm to solve the target problem, and show simulation results to validate the usefulness of our proposed method.

3.1 Assumptions and Problem Definition

In this section, we present assumptions for mobile nodes and formulate the problem of maximizing the k-coverage lifetime of a WSN with mobile sensor nodes.

3.1.1 Assumptions for Mobile Nodes

Both *static nodes* and *mobile nodes* are used as sensor nodes. Static nodes cannot be moved from their originally placed locations, while mobile nodes can move by wheels. We denote the sets of static and mobile sensor nodes by $P = \{p_1, ..., p_l\}$ and $Q = \{q_1, ..., q_m\}$, respectively. We assume that there is no obstacle in $Field$, and a mobile node can move straight to an arbitrary position in $Field$. The sensor nodes is deployed over the field without the excess and deficiency for k-coverage of the field. So, each static and mobile sensor node is always sensing node ($P \cup Q \subset U$).

Mobile nodes consume battery power not only by communication but also by movement. Power $Move(d)$ required to move d[m] conforms to formula (11) [7].

$$Move(d) = E_{move} \times d \tag{11}$$

Here, E_{move} is a constant. Each mobile node can move at V [m/s] where V is a constant value.

3.1.2 Problem Definition

When a WSN operates for a long time, batteries of some sensor nodes will be exhausted and k-coverage will be broken. Then, it is necessary to move mobile nodes one after another. So, we formulate a problem to find the data collection tree and the schedule of moving for all mobile nodes in order to maximize the k-coverage lifetime.

The initial WSN deployment time is denoted by t_0. t_{end} denotes the time when the k-coverage of the WSN cannot be maintained any longer due to battery exhaustion or failures of multiple nodes ($t_{end} \geq t_{life}$). For each $q \in Q$ and each $t \in [t_0, t_{end}]$, the speed (0 or V) and direction of q at time t is denoted by $Run(q, t)$. Then, for each $q \in Q$, the speed-direction schedule for q's movement during time interval $[t_0, t_{end}]$ is denoted as follows.

$$schedule(q, [t_0, t_{end}]) = \bigcup_{t \in [t_0, t_{end}]} \{Run(q, t)\} \tag{12}$$

Given the information on the target field $Field$, a sink node Bs and its position $Bs.pos$, $s.pos$, $s.energy$, and $s.range$ for each sensor node $s \in P \cup Q$, and constants E_{elec}, ϵ_{amp}, n, E_{sens}, E_{listen}, E_{move}, V, D, and I, our target problem for maximizing k-coverage lifetime denoted by t_{life} is to decide the schedule $schedule(s, [t_0, t_{end}])$ for each node $s \in P \cup Q$ and a data collection tree containing all sensor nodes that satisfies condition (9).

3.1.3 Modified Target Problem

Our target problem formulated in Section 3.1.2 is to decide speed-direction schedule of each mobile sensor node $q \in Q$ during time interval $[t_0, t_{end}]$. Then, we must decide a data collection tree including all sensor nodes whenever the positions of mobile nodes change. Solving the problem is considered to be very difficult because of the wide solution space. Therefore, we adopt a heuristic method to solve this problem stepping on the several stages as the following procedures:

1. Solving the problem to find the positions of mobile nodes and the data collection tree for maximizing *the WSN forecast endtime* (defined later) satisfying condition (9).

2. Whenever the battery of any sensor node is newly exhausted, go to step 1.

In the problem of step 1, its input is the same as the original problem. Its output is the new position of each mobile node $q \in Q$ denoted by $q.newpos$ satisfying condition (13) and the parent node of each sensor node $s \in P \cup Q$ denoted by $s.send$. We have the following constraint on $q.newpos$.

$$|q.pos - q.newpos| < \frac{V}{I} \tag{13}$$

Here, the new position of each mobile node is in the area where each mobile node can move in $\frac{1}{I}$ seconds.

The WSN lifetime t_{life} is the time of WSN termination considering the movement of mobile nodes in the future. It is difficult to calculate t_{life} strictly. So, we define the *WSN forecast endtime* when the battery of some sensor node is newly exhausted instead of t_{life}. Thus, we use the following objective function.

$$\textbf{maximize}\Big(t_{now} + \min_{s \in P \cup Q}\Big(\frac{s.energy}{C(s)} - \frac{Move(|s.pos - s.newpos|)}{C(s)}\Big)\Big) \tag{14}$$

where t_{now} is current time, and $C(s)$ is the energy consumption of sensor node s per second. If $s \in P$, $|s.pos - s.newpos| = 0$. So, $\frac{s.energy}{C(s)} - \frac{Move(|s.pos-s.newpos|)}{C(s)}$ means the time from present until the battery of the sensor node $s \in P \cup Q$ is exhausted.

3.1.4 Complexity

The Minimum Geometric Disk Cover Problem(GDC), which is an NP-hard problem, is defined as follows [18].

Problem GDC: Given a set N of RNs (points) distributed in the plane, place the smallest set M of Cover MBNs (disks) such that for every RN $i \in N$, there exists at least one MBN $j \in M$ exists such that $d_{ij} \leq r$.

The instance of GDC is set to (N, m, r), where N is a set of points, m is the number of disks, and r is a radius of each disk. Now, we assume that the WSN is constructed by m mobile nodes (no static nodes), sensing radius is set to r, and the target field is set to N. Existence of solution of GDC and 1-coverage of the field N using m mobile nodes are equivalent. Polynomial-time reduction from GDC to our target problem is possible. Thus, our target problem is NP-hard.

3.2 Algorithm

In this section, we describe the algorithm to solve the problem defined in Section 3.1.3.

3.2.1 Overview

Our algorithm decides the destinations of mobile sensor nodes and a data collection tree. Whenever the battery of any sensor node is newly exhausted, our algorithm is applied, as shown in Section 3.1.3. The proposed GA-based algorithm is supposed to be executed at the initial deployment time and ends when k-coverage of the target field is unable to be maintained.

In the calculation algorithm, each GA chromosome contains the positions for all the mobile nodes and the structure of the data collection tree. As a standard GA, it first generates initial candidate solutions to which it repeatedly applies GA operations. GA performance is largely influenced by the quality of the initial candidate solutions. To improve its performance, we provide trees for the initial candidate solutions by the *balanced edge selection method*, which is described later in Section 3.2.4. For calculation speed, we developed the *delta-k-coverage* judgment method that decides a sufficient condition for the k-coverage of the field by sensor nodes in Section 3.2.5.

3.2.2 Algorithm details

GA is a well-known meta-heuristic algorithm. The following is its basic procedure.

1. **Generation of initial candidate solutions**: N candidate solutions are randomly generated.

2. **Evaluation**: Objective function for each candidate solution is evaluated to grade each candidate solution.

3. **Selection**: N candidate solutions with better evaluations are selected.

4. **Crossover**: New candidate solutions are generated by mixing two randomly selected candidate solutions.

5. **Mutation**: Part of candidate solutions are randomly mutated.

6. **Check termination**: If the termination condition is met, the candidate solution with the highest evaluation is output as the solution. Otherwise, go to Step 2.

Below, we show our algorithm for each GA operation.

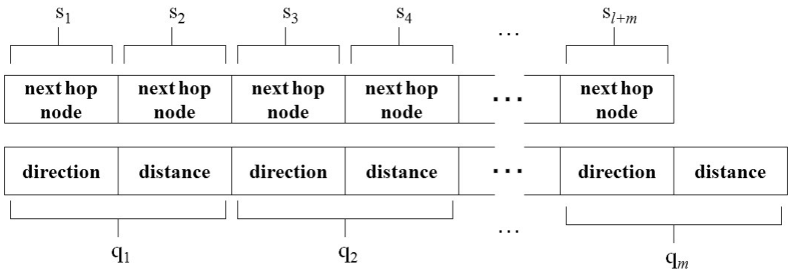
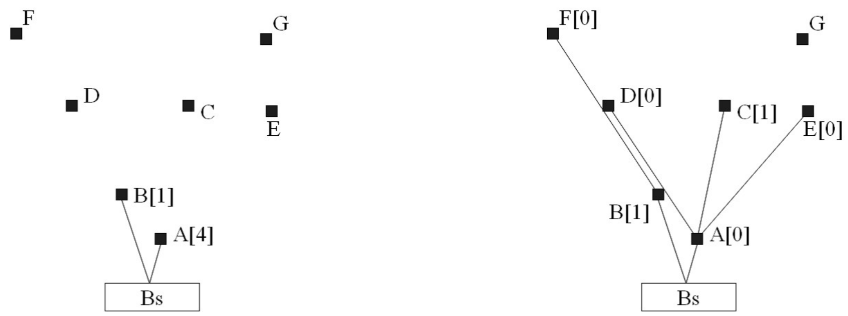


Figure 1. Encoding of Candidate Solution

Encoding of candidate solution: To apply a GA, each candidate solution has to be encoded, and the way of encoding sometimes greatly affects the algorithm performance. The coding in the proposed algorithm is shown in Fig. 1. Each candidate solution contains positions for $|Q|$ mobile nodes and the structure of the data collection tree consisting of $|P \cup Q|$ sensor nodes. The positions for the mobile sensor nodes are represented in polar coordinates to avoid generating impossible destinations of mobile sensor nodes. A data collection tree is represented by a set of node IDs.

Generation of initial candidate solutions: Initial candidate solutions are made from random variables. Angles and distances of mobile nodes are uniformly assigned distributed random values between 0 and 2π, and 0 and $Dist$, respectively (here, $Dist$ is a constant and typically set to the longest movable distance in the target field). As an initial parent node for each node, a node geographically closer to the sink node is randomly selected. For efficiency, three candidate solutions are added to the initial candidate solutions whose collection trees are made using the minimum cost spanning tree method where an edge cost is the square of the distance, the balanced edge selection method proposed in Section 3.2.4, and a method that directly connects all sensor nodes to the sink node.

Evaluation: The evaluation of each candidate solution verifies how long the target sensing field is k-covered by a simulation of WSN data transmission. The k-coverage lifetime is between the time when all mobile sensor nodes arrive at their new positions and the time when k-coverage cannot be maintained due to battery exhaustion of some nodes. If the decoded data collection network does not form a tree, the resulting evaluation is 0.



(a) Repeats 1 time (b) Repeats 3 times

Figure 2. Example of Balanced Edge Selection Algorithm

Strictly checking the k-coverage of the field is very expensive, and in the proposed algorithm, a sufficient condition for k-coverage is verified as described in Section 3.2.5.

Genetic operators: In our proposed method, we adopted roulette selection, an elite preservation strategy, uniform crossover, and mutation per locus. For uniform crossover, we treated each combination of angle and distance for a mobile sensor node as a gene. For mutation, random value is overwritten to a randomly selected locus.

Termination condition: The algorithm stops after a constant number of generations (one generation corresponds to one iteration of the GA algorithm in Section 3.2.2). In the experiment, we set 20 generations to this constant.

3.2.3 Local search technique

Our method uses the local search technique in addition to GA to improve the quality of solution.

For each mobile node $q \in Q$, we give moving destination randomly in a circle (radius is $1[m]$) centered on q. If WSN lifetime improves when all mobile nodes move to the destination, they move actually and are given new destinations. If it is not improved, this algorithm terminates.

3.2.4 Balanced edge selection method

The nodes near the sink node tend to consume more battery by forwarding the data transmitted from other nodes. In the balanced edge selection method, we first decide the set of nodes called *first-level nodes* directly connecting to the sink node. Next, we connect the remaining nodes to the first-level nodes one by one. The idea to select the first level nodes is as follows.

Step-1: The first level nodes is decided by testing Step-2 for every number of the nodes from 1 to $|P \cup Q|$ so that the maximum energy consumption by all the first level nodes is minimized. Here, we select each node in the increasing order of the distance from the node to the sink node.

Step-2: Data sent from the remaining nodes (other than the first-level nodes) must be forwarded through one of the first-level nodes to the sink node. Thus, the remaining nodes are distributed among the first-level nodes so that the energy consumption is balanced among the first-level nodes. Here, the energy consumption of each first-level node is estimated by the number of assigned nodes and the distance to the sink node.

Next, for each of the first-level nodes and the remaining nodes assigned to the node, we apply the above Step-1 and Step-2, recursively.

We will explain how the algorithm works using an example. Fig. 2(a) depicts the situation just after the first-level nodes A and B have been decided. In the figure, 'A[4]' means that the node A has been assigned 4 remaining nodes. Here, node A is closer to the sink Bs than node B, A has been assigned more remaining nodes. We suppose that A and B have been assigned {C, D, E, G} and {F}, respectively.

Next, the algorithm is recursively applied, and the second-level nodes are decided as shown in Fig. 2(b). Among nodes C, D, and E, D is closest to node A. Then, finally node G is assigned to node C, and the data collection tree completes.

3.2.5 Algorithm for checking k-coverage

Geometrically verifying whether any points of the target sensing field is contained by at least k sensor nodes' sensing ranges is very difficult.

In [21], Wang et al. proposed a sufficient condition for k-coverage, where the target field is divided into squares whose diagonals have the same lengths as the sensing radius to check if there is at least k sensor nodes in each square.

We propose a looser sufficient condition for the k-coverage of the target sensing field. In our method, we put checkpoints on grid points at intervals of δ in the target sensing field, and only check if each checkpoint is k-covered. However, even if all checkpoints are k-covered, some points between checkpoints may not be k-covered. The smaller δ is, the more the judgement accuracy improves. The judgment accuracy

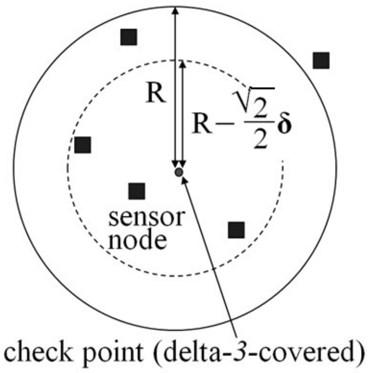


Figure 3. Condition of delta-k-coverage

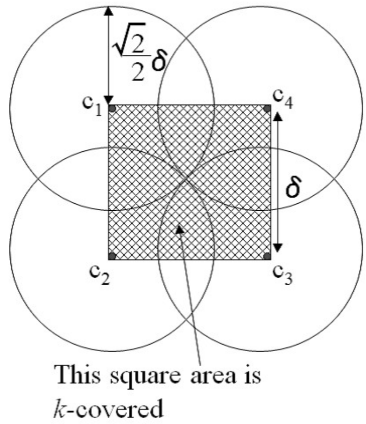


Figure 4. Checking k-coverage by delta-k-coverage

worsens when δ is too large. For $\delta < \sqrt{2}R$, we define delta-k-coverage which is a sufficient condition of k-coverage of the target field.

Definition 1

Checkpoint c is delta-k-covered if a circle whose center and radius are c and $R - \frac{\sqrt{2}}{2}\delta$, respectively, includes at least k nodes.

Fig. 3 shows that a checkpoint is delta-3-covered.

Theorem 1

Given checkpoints on grid points at intervals of δ ($\delta < \sqrt{2}R$) in a given field[2] , if each checkpoint is delta-k-covered, then the field is k-covered.

Proof

As shown in Fig. 3, if checkpoint c is delta-k-covered, then any points in the circle with radius $\frac{\sqrt{2}}{2}\delta$ centered at c are k-covered. Thus, as shown in Fig. 4, for neighboring checkpoints c_1, c_2, c_3, and c_4, if all are delta-k-covered, any points in the square formed by those checkpoints are k-covered. Therefore, Theorem 1 holds.

Theorem 1 only provides a sufficient condition of k-coverage. If we use a smaller value for δ, the condition is closer to the necessary and sufficient condition for k-coverage. However, the smaller value of δ will cause more checkpoints to be checked by delta-k-coverage. In our experiment in Section 3.3, $\frac{\delta}{R} = \frac{1}{10}$ is used.

[2] Note that outermost checkpoints must surround the target field.

3.3 Experimental validation

In this section, we show simulation results to validate the usefulness of Single tree method.

In order to evaluate the overall performance of Single tree method, we have measured the k-coverage lifetime, and compared it with the performance of other conventional methods including Wang's method [21], for several simulation configurations.

As a common configuration among the simulations, we used the parameter values shown in Table 1. Parameters of GA are determined by preliminary experiments as follows: the number of solution candidates is 20, the number of generations is 20, crossover rate is 1, and mutation rate is 0.01.

3.3.1 k-Coverage lifetime

We have compared k-coverage lifetime of our proposed method with conventional methods named as follows: (i) *Proposed Method* which uses all techniques in Section 3.2; (ii) *No Balancing Method* which randomly generates data collection trees as initial solution candidates in our method; (iii) *Static Method* which prohibits movement of mobile nodes in our method; and (iv) *Wang+Balancing Method* which decides the new positions of the mobile nodes by Wang's Method [21], uses Wang's k-coverage condition, and constructs a data collection tree by the balanced edge selection method and GA. In Wang's k-coverage condition, the field is divided into grids at intervals of $\frac{R}{\sqrt{2}}$, and the number of coverage is the number of the sensor nodes in each grid.

The configuration of this experiment other than Table 1 is provided as follows.

- Field size: 50m $\times$ 50m, 100m $\times$ 50m, and 100m $\times$ 100m
- Position of the sink node : around the south (bottom) end in the field
- Number of sensor nodes: 100, 200, and 300
- Proportion between numbers of static and mobile nodes: 25% and 75%
- Required coverage: $k = 3$

Note that the size of the target field should be appropriately decided so that the field can be sufficiently k-covered by a given number of nodes and coverage degree k. Thus,

Parameter	Value
Initial energy amount of each node	$s.energy$ = 32400 J (two AA batteries)
Energy consumption exponent	n = 2 (by referring to [7])
Energy consumption coefficient for data processing	E_{elec} = 50 nJ/bit (by [7])
Energy consumption coefficient for signal amplification	ϵ_{amp} = 100 pJ/bit/m^2 (by [7])
Energy consumption coefficient for moving	E_{move} = 8.267 J/m (by [22])
Energy consumption coefficient for sensing	E_{sens} = 0.018J/bit (by [7])
Energy consumption on idle time	E_{listen} = 0.043 J/s (by [23])
Energy consumption on sleeping time	E_{sleep} = 0.000054 J/s (by [23])
Radius of sensing range of each sensor	R= 20m (by [24])
Size of data for sensed information	D= 128bit (by [25])
Sensing frequency	0.1Hz (by [25])
Maximum radio transmission distance	300m (by [26])

Table 1. Common configuration for experiments

we used field size $50\text{m} \times 50\text{m}$ with 100 nodes for the basic case, and enlarged the field size proportionally to the number of nodes. In the experiment, the initial positions of nodes are given by uniform random variables.

We show simulation results in Figs. 5 and 6 for 3-coverage. These results are average of 30 trials.

Fig. 5 shows that two Proposed Methods (Balancing and No-Balancing) outperform Static Method to a great extent, independently of the number of nodes. The reason is that finding the appropriate positions of mobile nodes in a wide area greatly affects the performance. Wang+Balancing Method was not so different from Static Method. Initially, the field was k-covered by sensor nodes in all methods. In many cases, however, Wang's k-coverage condition was not satisfied. Then, Wang+Balancing Method moved mobile nodes to the new positions so as to satisfy Wang's k-coverage condition. When a node exhausted its battery, Wang+Balancing Method often could not find the new positions of mobile nodes satisfying Wang's k-coverage condition.

The figure also shows that Proposed Method achieves better performance than No Balancing Method. Thus, our proposed balanced edge selection algorithm is effective in extending the k-coverage lifetime. In the figure, 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 sink node Bs have to forward more data transmitted from their upstream nodes as the number of nodes increases, even though mobile nodes move closer to the sink node to help forwarding the data. In Fig. 5, the best and worst values of 30 trials by our algorithm were also shown. The difference of k-coverage lifetime of our algorithm was in the range from 84% to 109% compared with the average. We see that our algorithm does not output the solution with extremely bad performance.

Fig. 6 shows the computation time of each method. Proposed Method takes about 120 second in the case of 300 nodes for $k = 3$. This shows that it is possible to operate our method actually.

3.3.2 Efficiency of k-coverage judgment algorithms

We have measured and compared the accuracy and computation time of our delta-k-coverage judgment method and Wang's method [21]. Both methods are based on their own sufficient conditions for checking k-coverage. Our k-coverage condition is described in Section 3.2.5. Wang's k-coverage condition is that the field is divided

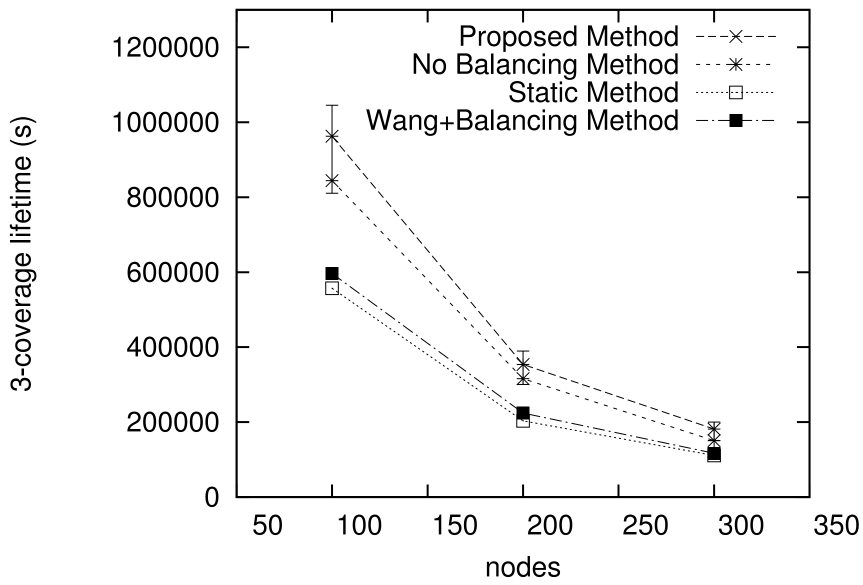


Figure 5. 3-coverage lifetime

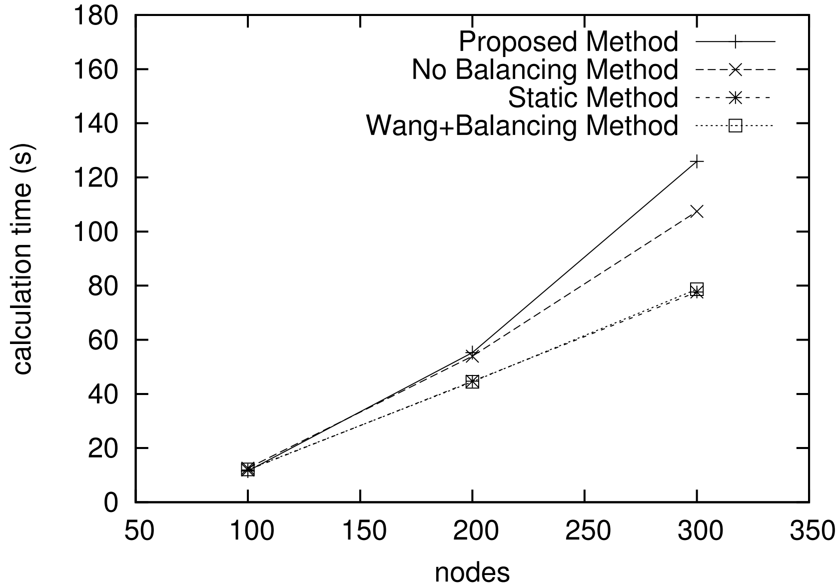


Figure 6. 3-coverage computation time

	k=1	k=2	k = 3
Wang's Method [21]	93	44	4
Proposed Method ($\delta = 0.5$m)	100	100	100
Proposed Method ($\delta = 1.0$m)	100	100	100
Proposed Method ($\delta = 2.0$m)	100	100	100
Proposed Method ($\delta = 4.0$m)	100	100	100
Proposed Method ($\delta = 8.0$m)	100	100	100
Proposed Method ($\delta = 12.0$m)	100	100	97
Proposed Method ($\delta = 16.0$m)	96	82	48
Proposed Method ($\delta = 20.0$m)	39	1	0

Table 2. The number of occurrences that the field is judged as k-covered (out of 100 simulation runs)

into grids at intervals of $\frac{R}{\sqrt{2}}$, and the number of coverage is the number of the sensor nodes in each grid (described in Section 3.3.1). Thus, if one of the methods judges affirmatively, then the field is actually k-covered. Conversely, even if both methods judge negatively, it is not always the case that the field is not k-covered. The higher the ratio to judge that the field is k-covered is, the higher the judgment accuracy is.

In this experiment, 300 static nodes are randomly deployed in the $100\text{m} \times 100\text{m}$ field. In this case, the field is almost always 3-covered. Therefore, it is expected to judge that 1, 2, and 3-coverage of the field are satisfied in all trials. We conducted the above simulation 100 times and measured the number of the occurrences that the field is judged to be k-covered out of the 100 simulations. Note that on some occurrences, the whole field is not actually k-covered since node positions are randomly decided.

We conducted the above simulations by changing the value of δ from 0.5m to 23.5m by 0.5m step for our delta-k-coverage judgment method, while the diagonal length of all squares in Wang's method is fixed to $10\sqrt{2}$m, which is the sensing radius of sensor nodes, and cannot be changed.

The experimental results on measured accuracy is shown in Table 2. Note that Table 2 shows part of the results for some important δ values.

Table 2 suggests that our delta-k-coverage judgment method is better than Wang's

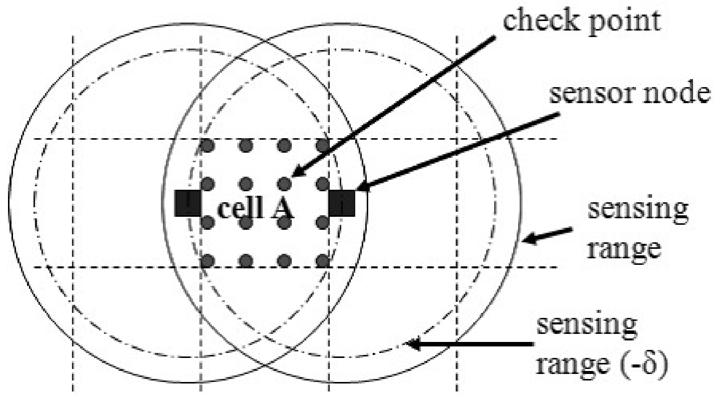


Figure 7. Example of Misjudge by Wang's Method

method for all numbers for k when δ is no bigger than 16m. The difference becomes bigger as k increases. Especially, when δ is no bigger than 12m, our algorithm almost perfectly judged k-coverage of the field, whereas Wang's method judged that only 4 occurrences out of 100 was 3-covered. Fig. 7 shows the example of node positions such that the difference of the judgement between our method and Wang's method is extreme. In Fig. 7, cell A is 2-covered actually. Wang's method judges that cell A is not covered because there is no sensor node in cell A. On the other hand, our method judges that cell A is 2-covered, since each check point is delta-2-covered. Wang's method takes a constant computation time around 0.13ms, while our method takes longer computation time, which is inversely proportional to δ, for example, 159ms for $\delta = 1$m, 2ms for $\delta = 8$m, and 1ms for $\delta = 12$m.

As a result, our algorithm takes longer computation time, however it is much more practical since it is adjustable depending on the required accuracy of k-coverage judgment within allowable computation time.

3.3.3 Influence of mobile nodes ratio for k-coverage lifetime

It is obvious that using n mobile nodes will achieve longer k-coverage lifetime than using n static nodes. However, a mobile node is much more expensive than a static node. In order to investigate the influence of mobile nodes ratio to all nodes, we measured k-coverage lifetime for 100, 200, and 300 sensor nodes, changing the mobile

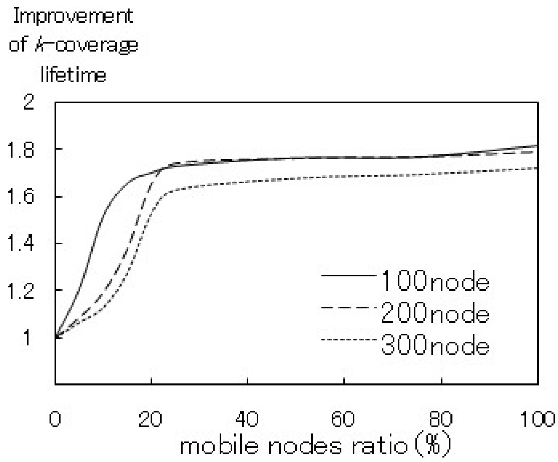


Figure 8. Improvement of k-coverage Duration for Mobile Nodes Ratio

nodes ratio from 0% to 100% by 5% step.

We show the results in Fig. 8. The results are average values of 30 simulations. Fig. 8 suggests that the k-coverage lifetime increased sharply in the ratio from 0% to 25%, and loosely from 25% to 100%. That means about 25% ratio of mobile nodes will be the best when we consider the deployment cost.

3.4 Conclusion for the Method Using Mobile Nodes

We formulated a k-coverage lifetime maximization problem for a WSN with mobile and static sensor nodes. This problem is NP-hard. So, we proposed a GA-based algorithm to decide the positions of mobile sensor nodes and to construct a data collection tree with balanced energy consumption for communication among nodes. We also defined a new sufficient condition for k-coverage based on checkpoints and proposed an algorithm to accurately judge k-coverage in reasonably short time. Through computer simulations, we confirmed that our method improved k-coverage lifetime to about 140% to 190% compared with other conventional methods for 100 to 300 nodes. Also, we confirmed that the best cost-performance is achieved when the mobile nodes ratio is about 25%. We also confirmed that our k-coverage judgement condition is much more accurate than Wang's method.

4. k-coverage Lifetime Maximization Method by Sleep Scheduling

In this chapter, we formulate the problem to maximize the k-coverage lifetime by sleep scheduling, propose an algorithm to solve this problem, and show simulation results to validate the usefulness of this method.

4.1 Problem Definition

In this section, we formulate a problem to maximize the k-coverage lifetime of WSN by scattering more-than-enough number of static sensor nodes at random over the field.

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.

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 ($t_{end} \geq t_{life}$). 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[3] 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 t_{life} is to decide the schedule $schedule(s, [t_0, t_{end}])$ for each node $s \in S$ that satisfies condition (9).

4.2 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 (9). Since sensing nodes periodically carry out sensing operation

[3] We assume that the time domain is discrete.

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 [27].

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 (9).

(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:

$$t_{now} + \min_{pos \in Field} \left(\frac{\Sigma_{s \in Cover(pos, t_{now})}(s.energy[t_{now}])}{\Sigma_{s \in Cover(pos, t_{now})}(C(s))} \right) \tag{15}$$

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 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:

$$t_{now} + \min_{s \in S} \left(\frac{s.energy[t_{now}]}{C(s)} \right) \tag{16}$$

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.

4.3 Algorithm of Single Tree Method

In this section, we describe an algorithm to solve the problem defined in Section 4.2.

4.3.1 Overview

Our algorithm finds operation modes for sensor nodes and a data collection tree for each unit time. In our algorithm, we make the minimal number of the nodes required for k-coverage active, and replacing the node that exhausted battery by another one.

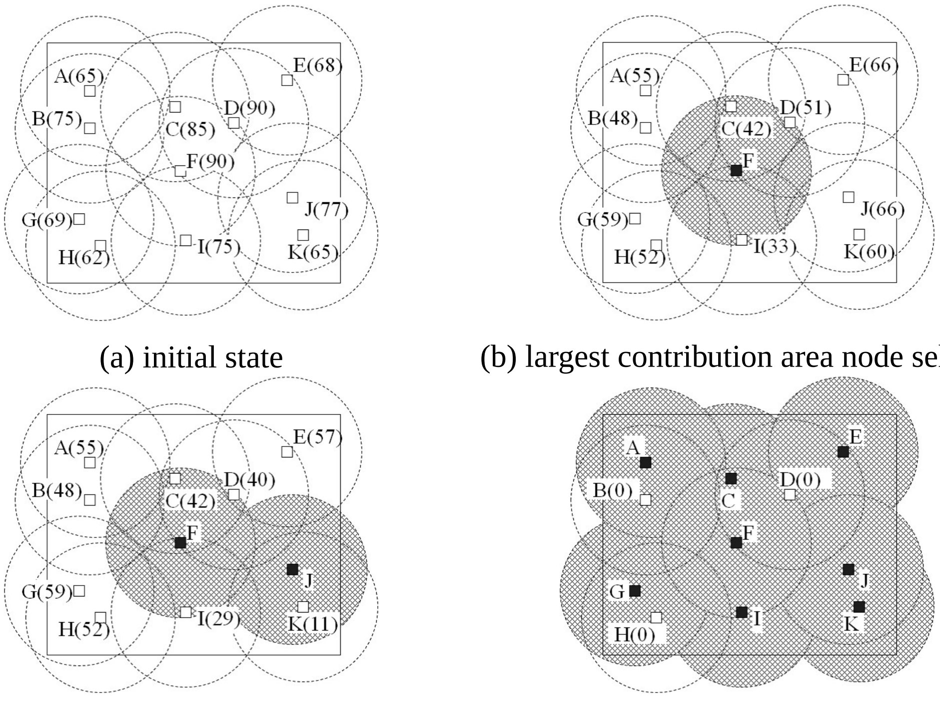
The algorithm is supposed to be executed at the initial deployment time and each of the next battery exhaustion time. The lifetime of the whole system ends when there are no sets of sensing nodes that satisfy condition (9).

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

Furthermore, we improve this method to a method having robustness. We describe the method in Chapter 5.

4.3.2 Wakeup Method

Wakeup method finds the minimal 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



(a) initial state

(b) largest contribution area node selected

(c) 2nd largest area node selected

(d) resulting state

Figure 9. Example of Applying Wakeup Method

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. 9 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. 9 (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. 9 (b)). Then the algorithm is applied to other sensor nodes. Fig. 9 (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. 9 (d) is the result after the algorithm terminates with a solution.

4.3.3 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 Section 3.2.4 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.

4.3.4 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 (17).

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.

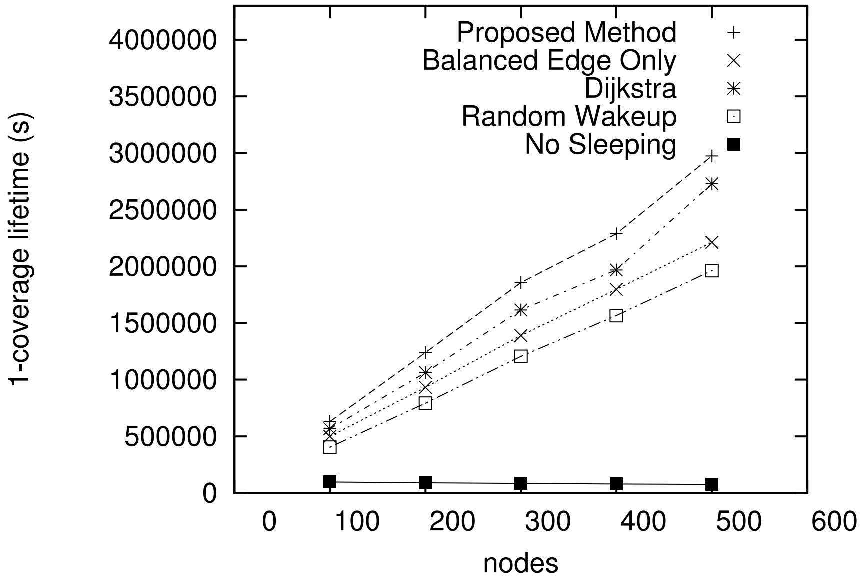


Figure 10. 1-Coverage Lifetime

We define the earliest time when the battery of some sensor node is exhausted (called the *next battery exhaustion time*) as follows:

$$t_{now} + \min_{s \in S} \left(\frac{s.energy[t_{now}]}{C(s)} \right) \tag{17}$$

where, t_{now} is current time and the energy consumption of sensor node s per unit of time ($C(s)$) is calculated by formula (6), (7), or (8).

4.4 Experimental Validation

In order to evaluate the overall performance of our proposed method, we have conducted computer simulations for measuring the k-coverage lifetime, 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 1.

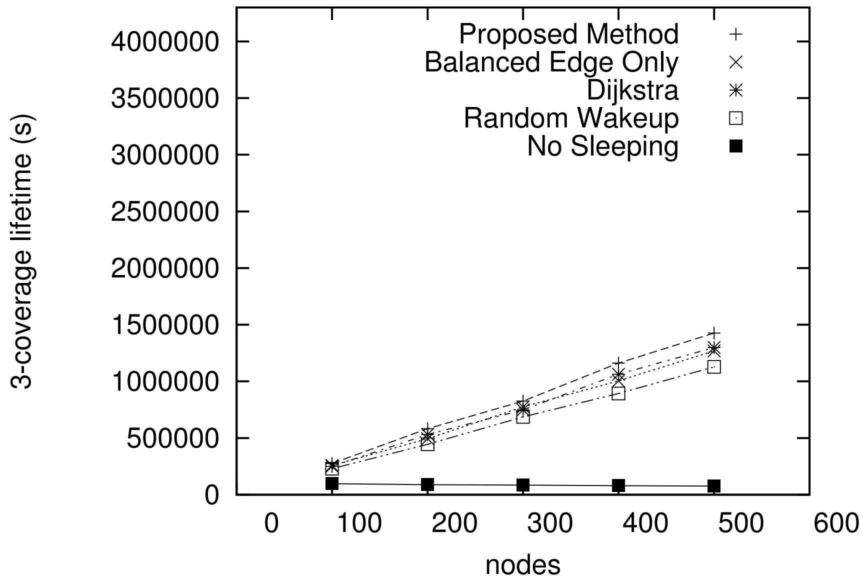


Figure 11. 3-Coverage Lifetime

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 4.3; (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 constructed by Dijkstra method [20] 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 [20] 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 1 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
- Required coverage: k=1 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 values.

We show experimental results obtained through computer simulations in Fig. 10 for 1-coverage and Fig. 11 for 3-coverage. These results are average of 40 trials.

Figs. 10 and 11 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 energy 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 our proposed Wakeup method that greedily selects a node the most effective to the k-coverage guarantees longer k-coverage lifetime than selecting nodes at random.

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.

4.5 Conclusion for Single tree method

In this chapter, we formulated a k-coverage lifetime maximization problem for a data gathering WSNs where only static sensor nodes are densely deployed. This problem is NP-hard. To quickly derive an approximate solution, we proposed Single tree method that activates the minimal number of nodes required for k-coverage at each time and replaces the node that exhausted battery by another one.

As a result, we confirmed that the proposed methods achieve 1.1 to 1.7 times longer k-coverage lifetime regardless of k and the number of nodes, than the other conventional methods for 100 to 500 node WSNs.

5. Robust Sleep Scheduling Method for k-coverage lifetime maximization

In this chapter, we describe Layer method that has robustness to solve the modified target problem defined in Section 4.1 and show simulation results to validate the robustness and k-coverage lifetime of Layer method.

5.1 Layer method

If a node belonging to the tree near the sink node failed, the field coverage degree may be lost by breaking many paths including the failed node. WSNs often require the continuation of its operation even if a node failure occurs. However, in Single tree method, we do not suppose the node failure. So, we propose Layer method for reducing coverage loss by a node failure in k-coverage lifetime maximization.

To achieve robustness, Layer method constructs a data collection tree by merging multiple independent sink-rooted trees. Layer method finds the solution of the problem defined in Section 4.1 and periodically applies it to the WSN with time interval T in two steps: *layer division* and *layer selection*. In layer division step, our method subdivides the set of all nodes into as many subsets of sensor nodes as possible where each subset 1-covers the field. Each of these subsets are called a *layer*. Then it constructs a sink-rooted tree for each layer. In order to k-cover the field, it selects k layers at the layer selection step. 1-coverage is always guaranteed even in the case of simultaneous failure of up to $k-1$ nodes.

After the layer division step, some nodes that do not belong any layer may remain. We call such nodes *free nodes*. We make free nodes relay the radio communication and repair the broken layer by adding some of the free nodes to the broken layer. Here, the broken layer means the layer in which some nodes exhausted battery and 1-coverage was broken.

Layer method is robust in the sense that it guarantees 1 or more coverage even if $k-1$ or less nodes fail. In the case that k-coverage breaks, it is easy to recover k-coverage by replacing the broken layer by one of non-activated layers.

The algorithm of Layer method is shown below. It starts just after the initial deployment.

1. Given the set of all nodes S.

2. *Layer division algorithm* (described in Section 5.1.1) constructs as many layers $L = \{L_1, L_2, ..., L_i\}$ as possible.

3. Balanced edge selection method constructs a data collection tree G_i for each layer L_i. Let $G = \{G_1, G_2, ..., G_i\}$ denote the set of data collection trees for all layers.

4. *Layer selection algorithm* (described in Section 5.1.2) selects k layers and put all nodes that belong to the selected layers into the set of sensing nodes U. If the number of available layers is less than k, the WSN terminates.

5. *Relay algorithm* (described in Section 5.1.3) decides the set of relaying nodes V and modifies G with V.

6. The set of sleeping nodes W is decided as $W = S - U - V$.

7. *Recalculation time* (described later) is calculated.

8. The sink node informs the calculation result (U, V, W, G, and recalculation time) to each node.

9. Each node changes its operation mode along with the result informed by the sink node. Each sleeping node sets its timer so that it will wake up at the recalculation time.

10. WSN operates until the recalculation time.

11. At the recalculation time, each sleeping node wakes up and starts waiting the calculation result informed from the sink node.

12. The sink node checks 1-coverage of each layer and repairs the broken layer by *layer recovering algorithm* (described in Section 5.1.4).

13. The sink node constructs a data collection tree for each repaired layer and removes the layer that was not successfully repaired in step 12.

14. The nodes that exhausted their battery are removed from S, return to step 4.

For fault tolerance, our algorithm awakes all sleeping nodes and recalculates periodically with time interval T. However, in the case that one or more nodes will exhaust their battery until time T elapses, our algorithm will recalculate at the time calculated by formula (17). So, given t_{now} as the present time, *recalculation time* is set as follows.

$$min(t_{now} + T, t_{exh}) \tag{18}$$

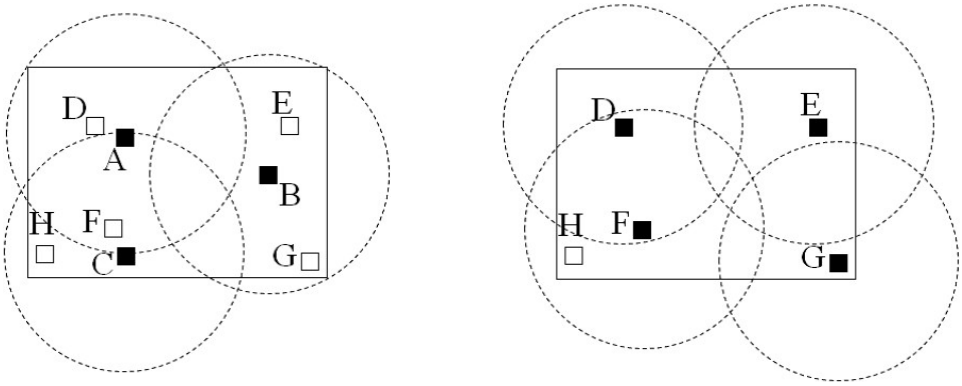
The algorithm of Layer method consists of four parts: (1) Layer Division Algorithm, (2) Layer Selection Algorithm, (3) Relaying Algorithm, (4) Layer Recovering algorithm. We describe each part of this algorithm in the following section.

5.1.1 Layer Division Algorithm

Layer division algorithm is shown as follows:

1. Let N, L, and i denote the set of nodes, the set of layers, and the variable of natural number, respectively. Set $N = S$, $L = \emptyset$, and $i = 0$ (S is the set of all nodes).
2. Make layer L_i from N by Wakeup method described in Section 4.3.2. and input L_i into L. If the new layer cannot be made, this algorithm terminates.
3. $N \leftarrow N - L_i$.
4. $i \leftarrow i + 1$, return to step 2.

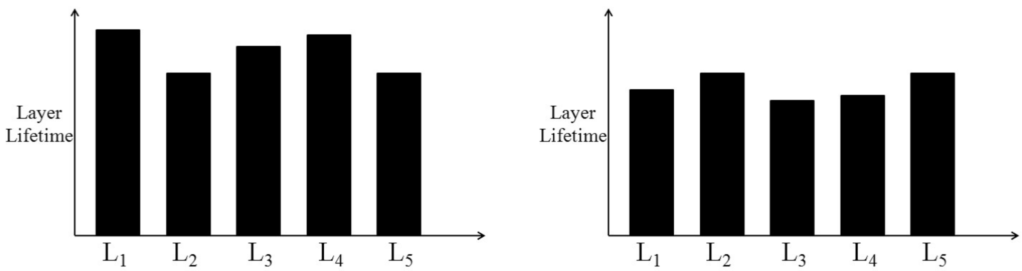
The example of layer division algorithm is shown in Fig. 12. As Fig. 12 (a), eight nodes (from A to H) are deployed over the field. First, wakeup method selects a set of nodes 1-covering the field from all nodes (here, node A, B, and C are selected). Layer division algorithm puts the selected nodes into layer L_1 ($L_1 = \{A, B, C\}$). Next, wakeup method also selects a set of nodes 1-covering the field from all nodes except for the nodes that belong to layer L_1 (here, node A, B, and C are selected as Fig. 12 (b)). The selected nodes D, E, F, and G are put into layer L_2 ($L_2 = \{D, E, F, G\}$). Here, all nodes except for the nodes that belong to layer L_1 and L_2 are only H. Layer division algorithm terminates since the new layer will be no longer made.



(a) Applying One Time

(b) Applying Two Times

Figure 12. Example of Applying Layer Division Algorithm



(a) First State

(b) Next State

Figure 13. Example of Applying Layer Selection Algorithm

5.1.2 Layer Selection Algorithm

Layer selection algorithm is shown as follows:

1. Given the time of calling Layer selection algorithm (t_{now}) and the set of all layers $L = \{L_1, L_2, ..., L_i\}$.
2. Calculate the *layer lifetime* for each layer in L. Here, layer L_j's layer lifetime means that the lowest remaining battery of the node in L_j at t_{now}.
3. Select top k layers in the descending order of the layer lifetime.
4. Algorithm terminates.

We describe the example of layer selection algorithm. Now, we assume that WSN application requires $k = 3$ and there are five layers L_1, L_2, L_3, L_4, and L_5 whose layer lifetimes are as Fig. 13 (a). In order to consume each layer lifetime averagely, layer selection algorithm activates 3 layers whose layer lifetimes are top 3 of all (L_1, L_3, and L_4). At the time of next calculation, these activated layers have consumed their layer lifetimes as Fig. 13 (b). Layer selection algorithm also activates 3 top layers (L_1, L_2, and L_5). Each layer lifetime is averagely consumed by continuation of this procedure.

5.1.3 Relay Algorithm

Relay algorithm is shown as follows:

1. Let Q denote the set of free nodes. Given the set of nodes $P = U$ (U is the set of sensing nodes) and the set of relaying nodes $V = \emptyset$.

2. Select the node $s \in P$ such that the s's energy consumption for data communication (calculated by formula (1)) is the largest of P.

3. Find the node $s_{relay} \in Q$ such that s_{relay} is the non-sensing node and the nearest from the center point of the link s and $s.send$.

4. Let s_{relay} relay the link s and $s.send$ and add s_{relay} to V, if the s's energy consumption per unit time $C(s)$ is smaller than the case of no-relaying. Otherwise, do nothing.

5. Remove s from P.

6. This algorithm terminates if $P = \emptyset$, otherwise, return to step 2.

The change from relay selection method is the condition of the selecting relaying nodes. In order to save the battery of the non-free node, relaying nodes are selected from free nodes. We focus on the battery consumption of sensing nodes and make free nodes relay the communication only if the battery consumption of sensing node reduces by this relaying.

5.1.4 Layer Recovering Algorithm

Layer recovering algorithm is shown as follows:

1. Given the set of free nodes N, the set of all layers L, the broken layer L_1, the area 1-covered by L_1 $Area$, and the target field $Field$.

2. Find the node s_1 such that the intersection of s_1's sensing area and the lost coverage area $Field - Area$ is the widest. Add s_1 into L_i and remove s from N. If $N = \emptyset$ or s_1 cannot be found, this algorithm terminates as incomplete.

3. Recalculate $Area$. If L_1 1-covers the field, this algorithm terminates as complete, otherwise, return to step 2.

5.2 Experimental Validation for Layer Method

In order to evaluate the overall performance of Layer method, we have conducted computer simulations for measuring robustness and k-coverage lifetime, and compared it with the performance of other conventional methods including Single tree method, for several experimental configurations.

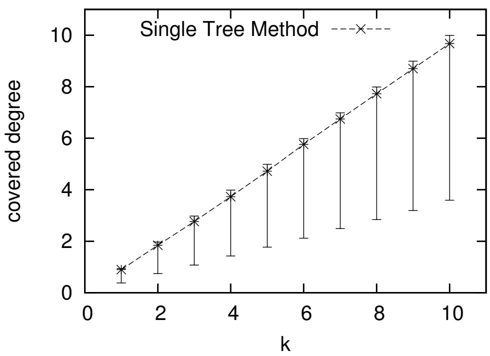
First, we measured the amount of lost field coverage degree by a node failure for the single tree method and Layer method. Next, we measured the k-coverage lifetime of Layer method and other conventional methods by changing the number of deployed nodes. Finally, we measured the k-coverage lifetime of Layer method and other conventional methods by changing k value.

As a common configuration among the experiments, we used the parameter values shown in Table 1 by referring to existing literatures. 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.

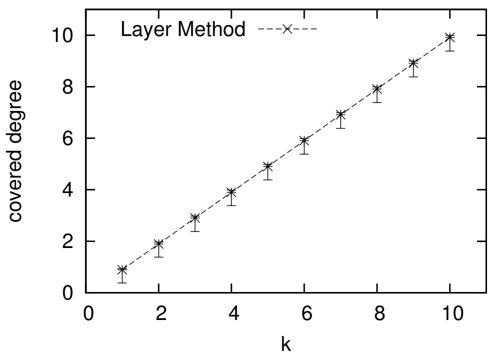
5.2.1 Loss of Field Coverage

In this experiment, we made one node to fail during each experiment and observed to what extent coverage degree is lost. We observed *average coverage degree*, which is the average number of coverage for all lattice points with distance of one meter. If the number of coverage for a point in the target field exceeds k, we regarded that the coverage degree is k for that point.

The configuration of this experiment other than Table 1 is provided as follows.



(a) Single Tree Method



(b) Layer Method

Figure 14. Covered Degree in the Case of One Node Failure

- field size: $100\text{m} \times 100\text{m}$
- the position of the sink node: bottom center of the field
- the number of nodes: 300
- k value: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

We conducted 300 trials. Figs. 14 (a) and (b) show the maximum, average, and minimum values of average coverage degree maintained by Layer method and Single tree method. Here, minimum value of average coverage degree means that a node failure occurs at the worst position of the data collection tree.

In Figs. 14 (a) and (b), we confirmed that sometimes extreme coverage loss occurs with the Single tree method while Layer method always keeps $k-1$ or more coverage degree. The average coverage degree of Layer method is 3% better than the Single tree method. This is the evidence that Layer method has less nodes whose paths to the sink node are cut by the node failure than Single tree method.

5.2.2 k-coverage Lifetime by Changing the Number of Nodes

We have measured k-coverage lifetime by changing the number of nodes among our proposed method and other conventional methods named as follows: (i) *Single Tree* which uses all techniques in Section 4.3; (ii) *Layer Changing* which is Layer method;

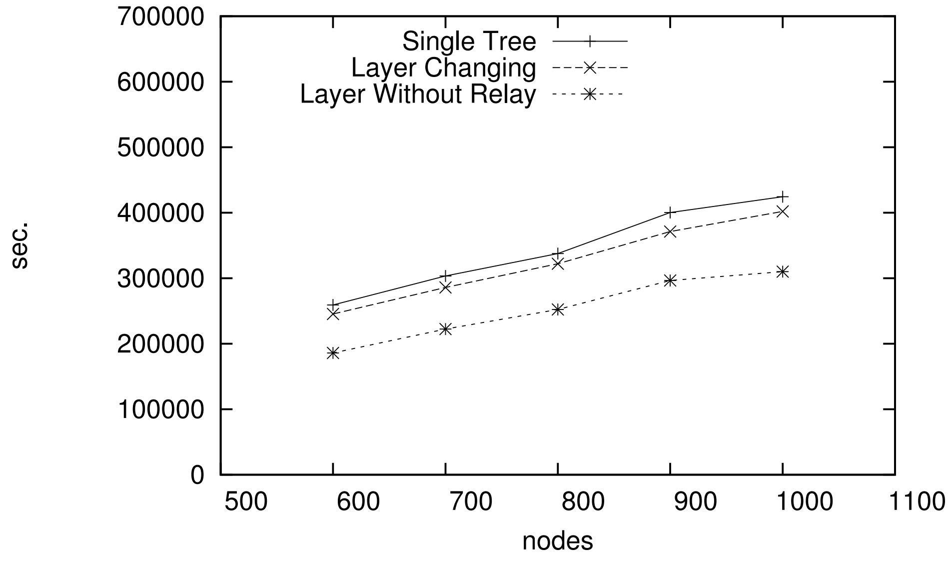


Figure 15. k-coverage Lifetime Depending on the Number of Sensor Nodes

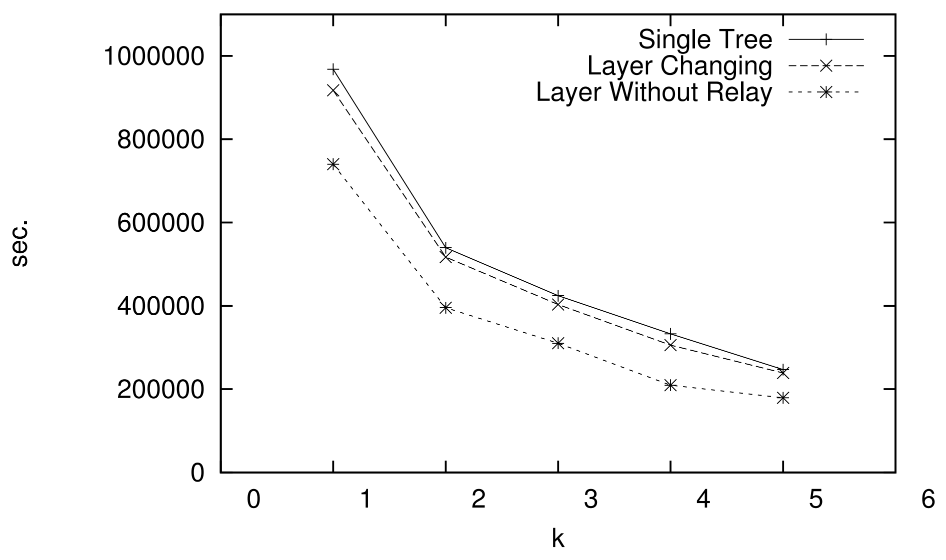


Figure 16. k-coverage Lifetime Depending on k

and (iii) *Layer without Relay* which is the method same as the Layer Method without Relay algorithm.

In this experiment, we assume that each node does not fail. The sink informs the calculation result to each sensor node by flooding. To guarantee that each node completely receives the calculation result, the communication range radius of each node is larger than twice the sensing radius.

Each sleeping node periodically wakes up before each recalculation time and informs its battery amount to the nearest sensing or relaying node. Each sensing or relaying node sends both all battery amount data it has and sensing data to the sink.

The configuration of this experiment other than Table 1 is provided as follows.

- field size: $200\text{m} \times 200\text{m}$
- the position of the sink node: bottom center of the field
- the number of nodes: 600, 700, 800, 900, and 1000
- k value: 3
- battery amount data of one node: 25bit

Note that the size of the target field should be appropriately decided so that the field can be k-covered without extremely surplus nodes for a given number of nodes and coverage degree k. Thus, we used field size $200m \times 200m$, that is, when 600 sensing nodes are randomly deployed in the target field, there will be extremely surplus nodes for k=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. 15. These results are average of 50 trials.

Fig. 15 shows that the k-coverage lifetime of Layer Changing is only 5% less than Single Tree. The reason is the overhead of layer division. The range of the data collection tree and the each node's operation mode that can be changed is limited in Layer method. The figure also shows that the k-coverage lifetime of Layer Changing is 20% more than Layer without Relay. This is an evidence that Relaying algorithm is effective to extend the k-coverage lifetime.

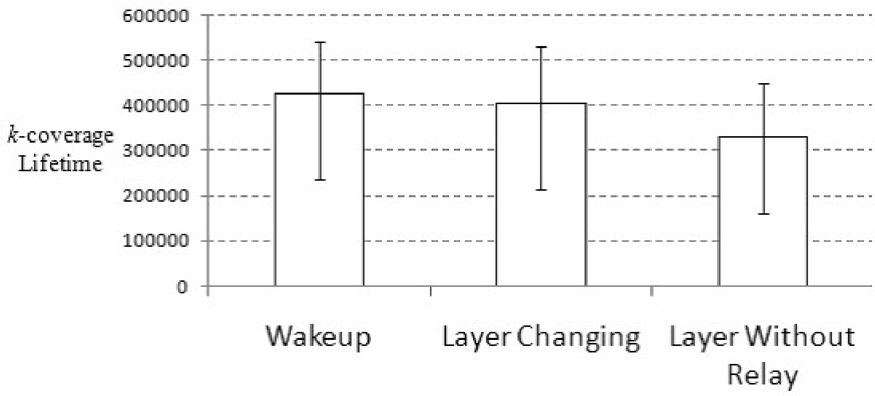


Figure 17. Maximum and Minimum Value of 3-coverage Lifetime

We also confirmed that Layer method (decision of sensing nodes and construction of a data collection tree) takes few calculation time. In these experiments, maximum calculation time of Layer method was 0.9 seconds when the number of nodes is 500.

5.2.3 k-coverage Lifetime by Changing k Value

We have measured k-coverage lifetime by changing k among our proposed method and other conventional methods described in Section 5.2.2. In this experiment, we assume that each node does not fail.

The configuration of this experiment is almost same as Section 5.2.2. The difference from Section 5.2.2 is provided as follows.

- number of nodes: 1000
- k: 1, 2, 3, 4, and 5

We show experimental results obtained through computer simulations in Fig. 16. These results are average of 50 trials. The figure shows the average values of k-coverage lifetime by large bars and maximum and minimum values of k-coverage lifetime by small error bars.

Fig. 16 shows that each k-coverage lifetime decreases by increasing k value. The reason is that many nodes cannot save their battery by increasing k value, because the number of sensing nodes for k-coverage of the field depends on k value. The figure also shows that the range between the maximum and minimum k-coverage lifetime is wide. The reason is that k-coverage lifetime strongly depends on the positions of sensor nodes.

We confirmed that Layer method is more effective for k-coverage lifetime than the other methods, regardless of k value.

5.3 Conclusion for Sleep Scheduling Methods

In this chapter, we proposed Layer method to solve both sleep scheduling problem and robustness problem. Layer method achieves robustness by constructing multiple trees. This method groups all nodes into layers such that each layer 1-covers the field, constructs a data collection tree for each layer, and makes only k layers active.

Through simulations, we confirmed that Layer method always keeps $k-1$ or more coverage degree if one node failure happens and the lifetime with Layer method is only 5% less than Single tree method.

6. Conclusion

In this thesis, we studied k-coverage lifetime maximization problem for data gathering WSNs through three different approaches.

In the first approach, we supposed WSNs with both static and mobile sensor nodes and proposed a GA-based algorithm to derive the semi-optimal solution for positions of mobile sensor nodes and a data collection tree with balanced energy consumption for communication among nodes. We also defined a new sufficient condition for k-coverage based on checkpoints and proposed an algorithm to accurately judge k-coverage in reasonably short time. Through computer simulations, we confirmed that a data collection tree constructed with Balanced edge selection method can extend the k-coverage lifetime to a great extent by effectively balancing the communication load among sensor nodes. We also confirmed that our k-coverage judgement condition is much more accurate than Wang's method. Also, the proposed condition takes reasonably short evaluation time, although the time is longer than Wang's. Furthermore, we confirmed that the best cost-performance is achieved when the mobile nodes ratio is about 25%.

In the second and third approaches, we targeted dense WSNs with only static sensor nodes. We proposed Single tree method that k-covers the field by the minimal number of active nodes, and maintains it by replacing the node that exhausted battery by another one. Through simulations, we confirmed that Single tree method achieves k-coverage lifetime almost proportional to the density of sensor nodes. We also proposed Layer method that improves the robustness of WSNs by using multiple independent data collection trees. Through simulations, we confirmed that Layer method always keeps $k - 1$ or more coverage degree if one node fails. We believe that Layer method can maintain the 1-coverage of the field even if up to $k - 1$ nodes fail at the same time. We also confirmed that the overhead of the proposed mechanism used in Layer method is reasonably small.

There are some challenges to make our algorithms practically applicable for data gathering WSNs. The first one is about radio interference among sensor nodes. In this thesis, we assumed that the sensing frequency is not very high. For the high sensing frequency WSNs, however, we must consider the way to cope with the interference problem. The second one is to develop the distributed algorithm of each proposed method. Currently, the proposed three algorithms are centralized algorithms executed by the sink. Calculating a solution in a centralized fashion and disseminating it to all nodes by flooding/broadcasting will take too much cost for large scale WSNs. So, we will develop the distributed algorithms. The last one is construction of a WSN testbed and evaluation of our methods on the testbed. There are some differences between the sensor node's energy consumption model and real energy consumption. So, we will develop a testbed consisting of multiple static and mobile sensor nodes capable of wireless communication using LEGO mindstorms NXT [28] and Arduino [29] and evaluate our methods on the testbed.

Acknowledgements

First of all, I would like to express my gratitude to Professor Minoru Ito for his support and constant encouragement. This work could not be achieved without his support.

I am deeply grateful to Professor Minoru Okada for his useful comments and valuable discussions.

I deeply thank Associate Professor Keiichi Yasumoto for his kind support and suggestions throughout my whole graduate study. I learned a lot from him about academic thinking, technical writing, presentation skills and how to do research. His advice will be very helpful in my future academic activity.

I would also like to thank Associate Professors Naoki Shibata of Shiga University and Yoshihiro Murata of Hiroshima City University for their support and valuable technical discussions. I learned a lot from them about technical things such as algorithms and programming techniques.

I would also like to thank Assistant Professors Weihua Sun and Yukiko Yamauchi for giving me many valuable comments and suggestions.

I would also like to thank all current and previous members in Foundation of Software laboratory. Particularly, I would like to thank members of the sensor network group and multimedia interaction group, Keiji Matsumoto, Shun Sakaguchi, Kyohei Fujimoto, and Koji Noguchi.

Finally, I would like to thank all people from whom I have received a kindness.

References

[1] Warneke, B., Last, M., Liebowitz, B. and Pister, K.S.J.: "Smart Dust: communicating with a cubic-millimeter computer," *Proc. of Computer, Vol. 34, Issue 1*, pp. 44–51, (2001).

[2] Fukatsu, T. and Hirafuji, M.: "Long-Term Monitoring System using Field Monitoring Servers," *Proc. of AFITA/WCCA Joint Congress on It In Agriculture*, pp. 685–691 (2004).

[3] Balister, P., Bollobas, B., Sarkar, A. and Kumar, S.: "Reliable Density Estimates for Coverage and Connectivity in Thin Strips of Finite Length," *Proc. of the 13th annual ACM Int'l conference on Mobile computing and networking (Mobicom 2007)*, pp. 75–86, (2007).

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

[5] Abrams, Z., Goel, A. and Plotkin, S.: "Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks," *Proc. of the third Int'l symposium on Information processing in sensor networks*, pp. 424–432, (2004).

[6] Poduri, S. & Sukhatme, G.S.: "Constrained coverage for mobile sensor networks," *Proceedings of International Conference on Robotics and Automation (ICRA2004)*, pp. 165–171, (2004).

[7] Wang, G., Cao, G., La Porta, T., & Zhang, W.: "Sensor Relocation in Mobile Sensor Networks," *Proceedings of The 29th IEEE International Conference on Computer Communications (INFOCOM 2005)*, pp. 2302–2312, (2005).

[8] Mei, Y., Xian, C., Das, S., Charlie Hu, Y., and Lu, Y.: "Repairing Sensor Network Using Mobile Robots," *Proc. of IEEE Workshop on Wireless Ad hoc and Sensor Networks(WWASN 2006)*, (2006).

[9] Tang, X. & Xu, J.: "Extending Network Lifetime for Precision-Constrained Data Aggregation in Wireless Sensor Networks," *Proceedings of The 30th IEEE*

International Conference on Computer Communications (INFOCOM 2006), pp. 1–12, (2006).

[10] Heinzelman, W.R., Chandrakasan, A., & Balakrishnan, H.: “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS 2000)*, pp. 1–10, Vol. 2, (2000).

[11] Cao, Q., Abdelzaher, T., He, T., & Stankovic, J.: “Towards Optimal Sleep Scheduling in Sensor Networks for Rare-Event Detection,” *Proceedings of The 4th International Symposium on Information Processing in Sensor Networks (IPSN2005)*, pp. 20–27, (2005).

[12] Keshavarzian, A., Lee, H., & Venkatraman, L.: “Wakeup scheduling in wireless sensor networks,” *Proceedings of The 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc2006)*, pp. 322–333, (2006).

[13] Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: “Maximizing k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes,” *IPSJ Transactions on Mathematical Modeling and its Applications* (in Japanese), Vol.2, No.3, pp. 75–86 (2009).

[14] Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., & Ito, M.: “Extending k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes,” *Proceedings of The 5th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob2009)*, pp. 48–54, (2009).

[15] Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: “Sleep Scheduling Method Based on Node Set Division for Maximizing Wireless Sensor Network Lifetime,” *IPSJ Transactions on Mathematical Modeling and its Applications* (in Japanese), Vol. 3, No. 3, pp. 140–153 (2010).

[16] Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., & Ito, M.: “Extending k-Coverage Lifetime of Wireless Sensor Networks with Surplus Nodes,v *Proceedings of The 5th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2010)*, pp. 9–16, (2010).

[17] Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: “Constructing Robust k-covered WSN with Multiple Data Collection Trees,” *Proc. of the 18th IEEE Int’l. Conf. on Network Protocols (ICNP 2010) Poster Session*, (2010).

[18] Srinivas, A., Zussman, G., & Modiano, E.: “Mobile Backbone Networks - Construction and Maintenance,” *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2006)*, pp. 166–177, (2006).

[19] Goldberg, D. E.: “Genetic Algorithms in Search, Optimization, and Machine Learning,” *Addison-Wesley Publishing Company, Inc.* (1989).

[20] Dijkstra, E.W.: “A Note on two Problems in Connection with Graphs,” *Journal of Numerische Mathematik*, Vol. 1, pp. 269–271, (1959).

[21] Wang, W., Srinivasan, V., & Chua, K. C.: “Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks,” *Proceedings of The 13th Annual International Conference on Mobile Computing and Networking (MobiCom 2007)*, pp. 39–50, (2007).

[22] Dantu, K., Rahimi, M., Shah, H., Babel, S., Dhariwal, A., & Sukhatme, G. S.: “Robomote: enabling mobility in sensor networks,” *Proceedings of The 4th International Symposium Information Processing in Sensor Networks (IPSN 2005)*, pp. 404–409, (2005).

[23] Crossbow Technology, Inc.: “MICA2: Wireless Measurement System,” *http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA.pdf*.

[24] Ganeriwal, S., Kansal, A., & Srivastava, M. B.: “Self aware actuation for fault repair in sensor networks,” *Proceedings of International Conference on Robotics and Automation (ICRA2004)*, pp. 5244–5249, (2004).

[25] Kamimura, J., Wakamiya, N., & Murata, M.: “Energy-Efficient Clustering Method for Data Gathering in Sensor Networks,” *Proceedings of the First Workshop on Broadband Advanced Sensor Networks (BaseNets2004)*, pp. 31–36, (2004).

[26] Crossbow Technology, Inc.: “IRIS mote,” *http://www.xbow.jp/mprmib.pdf*.

[27] 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).

[28] The LEGO Group: "LEGO Mindstorms NXT," http://mindstorms.lego.com/.

[29] Arduino: "an Open-Source Electronics Prototyping Platform," http://www.arduino.cc/, (2010).

List of Major Publications

Journal Papers

1. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Maximizing k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes," *IPSJ Transactions on Mathematical Modeling and its Applications* (in Japanese), Vol.2, No.3, pp. 75–86 (Dec. 2009). (corresponding to Chap. 3)

2. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Sleep Scheduling Method Based on Node Set Division for Maximizing Wireless Sensor Network Lifetime," *IPSJ Transactions on Mathematical Modeling and its Applications* (in Japanese), Vol. 3, No. 3, pp. 140–153 (Oct. 2010). (corresponding to Chap. 4 and Chap. 5)

International Conference

1. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Maximizing Lifetime of Wireless Sensor Networks with Mobile Sensor Nodes," *Proc. of 2008 Int'l Workshop on Sensor Network Technologies for Information Explosion Era (SeNTIE 2008)*, pp. 141–148, (Apr. 2008).

2. 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 the 5th IEEE Int'l. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob'2009)*, pp. 48–54, (Oct. 2009). (corresponding to Chap. 3)

3. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Extending k-Coverage Lifetime of Wireless Sensor Networks with Surplus Nodes," *Proc. of the 5th Int'l. Conf. on Mobile Computing and Ubiquitous Networking (ICMU 2010)*, pp. 9–16, (Apr. 2010). (corresponding to Chap. 4)

4. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Constructing Robust k-covered WSN with Multiple Data Collection Trees," *Proc. of the 18th*

IEEE Int'l. Conf. on Network Protocols (ICNP 2010) Poster Session, (Oct. 2010). (corresponding to Chap. 5)

Book Chapter

1. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Maximizing Lifetime of Data Gathering Wireless Sensor Network," *IN-TECH book "Sustainable Wireless Sensor Networks," chapter 19*, pp. 431–451, (Jan. 2011).

Other Publications

International Conference

1. Matsumoto, K., Katsuma, R., Shibata, N., Yasumoto, K., and Ito M.,: "Minimizing Localization Cost with Mobile Anchor in Underwater Sensor Networks," *Proc. of the 4th ACM International Workshop on UnderWater Networks (WUWNet'09),* (Nov. 2009)

Domestic Conference

1. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Low Power Routing in Wireless Sensor Networks with Movable Sensor Nodes," *Proc. of the 2007 Multimedia, Distributed, Cooperative and Mobile Symposium (DICOMO2007)* (in Japanese), pp. 966–971, (Jul. 2007)

2. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Using Mobile Sensor nodes for Maximizing Lifetime of K-Coverage Wireless Sensor Networks," *Proc. of the 2008 Multimedia, Distributed, Cooperative and Mobile Symposium (DICOMO2008)* (in Japanese), pp. 382–395, (Jul. 2008)

3. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Maximizing k-Coverage Lifetime of Wireless Sensor Networks Using Mobile Sensor Nodes," *Technical Report of IPSJ* (in Japanese), 2009-MPS-73, pp. 49–52 (Mar. 2009).

4. Matsumoto, K., Katsuma, R., Shibata, N., Yasumoto, K., and Ito M.,: "Localization with Moving Anchor Node for UnderwaterWireless Sensor Networks," *Technical Report of IPSJ* (in Japanese), 2009-MBL-49 No. 5, pp. 1–7 (May 2009).

5. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Extending Data Aggregation Lifetime in Wireless Sensor Networks with Surplus Nodes," *Proc. of the 2009 Multimedia, Distributed, Cooperative and Mobile Symposium (DICOMO2009)* (in Japanese), pp. 322–332, (Jul. 2009)

6. Noguchi, K., Katsuma, R., Nagataki, H., Yamauchi, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Remote Lecturing Systems with Screen Layout Selection De-

pending on User's Situation," *Technical Report of IPSJ* (in Japanese), 2009-CE-103 No. 1, pp. 1–8 (Mar. 2010).

7. Matsumoto, K., Katsuma, R., Shibata, N., Yasumoto, K., and Ito M.,: "Low-Cost Localization Method for Underwater Sensor Nodes using Mobile Anchor," *Technical Report of IPSJ* (in Japanese), 2010-MPS-78 No. 5, pp. 1–6 (May 2010).

8. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Sleep Scheduling Method Based on Node Set Division for MaximizingWireless Sensor Network Lifetime," *Technical Report of IPSJ* (in Japanese), 2010-MPS-78 No. 6, pp. 1–6 (May 2010).

9. Katsuma, R., Murata, Y., Shibata, N., Yasumoto, K., and Ito M.,: "Extending Data Aggregation Lifetime in Wireless Sensor Networks with Surplus Nodes," *Proc. of the 2009 Multimedia, Distributed, Cooperative and Mobile Symposium (DICOMO2009)* (in Japanese), pp. 540–550, (Jul. 2010)