Master's Thesis

Title

A Coverage Control Mechanism Satisfying Application Requirments in a Wireless Sensor Network

Supervisor

Professor Masayuki Murata

Author

Indika Suranjith Abeyweera

Feb. 14, 2007

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

A Coverage Control Mechanism Satisfying Application Requirments in a Wireless Sensor Network

Indika Suranjith Abeyweera

Abstract

Recently, a wireless sensor network is expected to be one of social infrastructures to create a safe and secure living environment. For this purpose, mechanisms to monitor conditions of environment, detect an event, and report obtained information promptly and reliably are required. Among sensing, decision, and transmission, in this thesis, we focus on the first, which is often referred as a coverage problem, i.e., how to keep the target region always monitored or watched by sensor nodes. We first propose three placement algorithms of additional sensor nodes to satisfy the required degree of coverage. They are coverage-based method, weighted coverage-based method, and void-based method. The region is divided into small pixels and then those methods decide which pixels to put additional nodes to have all pixels are monitored by the sufficient number of sensor nodes by using a centralized algorithm. The coverage-based and weighted coverage-based methods try to cover the maximum number of uncovered pixels by adding sensor nodes. On the other hand, the void method aims at reducing the number of voids, areas which are not covered enough. Through simulation experiments, we confirmed that the void-based method requires the slightly smaller number of additional nodes to attain the coverage degree of one. To cover a 50×50 m^2 region by sensor nodes of 10 m sensing radius, the required numbers are 24, 24, and 23 for the coverage, weighted-coverage, and void methods, respectively. On the other hand, for the higher coverage degree, the weighted coverage-based method outperforms the others by requiring only 36 sensor nodes against 41 for the coverage method and 80 for the void method. Next, we propose an energy-efficient scheduling mechanism with which redundant sensor nodes move to a sleep mode to save energy consumption, while keeping the required degree of coverage for the whole region. A sensor node follows a state transition, where a set of timers are used to determine the duration of each state. We first verify that the energy-dependent timer setting can balance the energy consumption among sensor nodes. However, we also find that timer setting is not trivial. Then, to investigate the influence of timer setting on the lifetime in a mathematical way, we model the state transition as a Markov chain. By using analytical results, we can make an energy-efficient scheduling mechanism, with which a sensor node appropriately sets timers depending on its local observation.

Keywords

Sensor Network

Sensing Coverage

Sleep Scheduling

State Transition

Markov Model

Contents

List of Figures

List of Tables

1 Introduction

Recently, a lot of research activities have been devoted into the wireless sensor network (WSN) field because of its various and attractive applications, including habitat monitoring, environmental control, agriculture, factory/building/home automation and security, disaster and crime prevention, and so on.

A sensor node is a small electronic device. It consists of one or more sensors for, for example, temperature, humidity, infrared, accelerator, and ultrasonic, a simple radio transceiver, and a processor of limited computational capability with a limited amount of memory. It also has a battery of limited capacity on whose power supply it operates. Through radio communication, sensor nodes deployed in the region to monitor organize themselves as a WSN. Each sensor node begins to monitor its surroundings by equipped sensors. For periodic monitoring applications, all or some of sensor nodes report their obtained sensor information to a designated sensor node or a special purpose node, which is called a base station or sink, at regular intervals. For an event-driven type of applications, a sensor node decides whether to take action or not depending on obtained sensor information. For example, a sensor node detects a fire when the concentration of smoke and the temperature in a room go beyond predetermined thresholds. Then it sends a notification to a base station through which the fire is reported to the observation center and, at the same time, it would send a message to other nodes in the sensor network to activate their fire alarm. Message transmission is done directly, i.e., single hop communication, or indirectly, i.e., multihop communication, to the destination. Since the energy efficiency is one of the major concerns in a WSN to prolong the lifetime and monitor the region as long as possible, multihop communication is preferred by limiting the transmission power of radio signals.

A WSN is responsible for sensing, decision, and transmission. A WSN has to ensure that the whole region to monitor is covered by sensors. It means that all points in the region must be within the sensing range of the sufficient number of active sensor nodes. Therefore, the number of sensor nodes deployed in the region must be large enough to achieve the degree of coverage required by an application. In addition, to save the battery consumption, sensor nodes have a sleep mode and a sensor node goes to sleep by turning off unnecessary modules when it considers it is redundant. Therefore, an appropriate sleep scheduling mechanism is required to keep the sufficient number of sensor nodes awake for the coverage, while making as many sensor nodes as possible sleep. These issues are called the coverage problem [1, 2]. Next, a WSN has to ensure that an event to report is detected and reported. For this, a sensor node has to collect enough information to make a correct decision. If sensors are unreliable for their sensing errors, a sensor node has to collect the larger number or amount of sensor information and conjecture the current status of environment. This issue is called the data fusion or data mining [3, 4]. Finally, a WSN has to ensure that sensor information required for an application is transmitted to a designated point, e.g., sink, in a WSN. A sensor node can send a message or information to a neighboring sensor node, which is awake with an active receiver and in the range of radio signals. At this time, the sensor node has a unidirectional link to the neighboring sensor node. If the sensor node is also in the range of radio signals of the neighboring sensor node, the link is bidirectional. For a message to reach the destination, there must be a series of unidirectional links towards the destination or bidirectional links to the destination. This is called the connectivity problem [5]. Among the three, we focus on the coverage problem in this thesis.

In this thesis, we assume that an application of a WSN has the requirement on the degree of coverage of the whole region depending on its purpose of monitoring the region. The degree of coverage is specified by *k-coverage*, which is considered to be satisfied when all points in the region are within the sensing range of more than k actively sensing sensor nodes. The degree of coverage of a point in the region, i.e., the number of sensor nodes which have the point in their sensing area, is called *coverage value* hereafter. Since sensor nodes may be deployed in the region in a random and unplanned manner, there can be areas which are fully uncovered or covered by less than k sensor nodes. Here in this thesis, those areas with the insufficient degree of coverage are called *sensing voids.* In other words, a sensing void is a region of points whose coverage values are less than the requirement k. Even if an administrative person carefully makes a distribution plan to achieve k-coverage with the optimal placement of the minimum number of sensor nodes, there is possibility of sensing voids due to inexact localization and sensor's sensitivity. Therefore, in the first part of the thesis, we propose three placement methods to reach the desired

k-coverage by adding the smallest number of sensor nodes at appropriate locations. They are the coverage-based, weighted coverage-based, and void-based methods. To simplify the problem, the region is first divided into rectangle or square pixels and coverage values are considered for pixels. Then, in all methods, sensor nodes are added one by one at appropriate pixels. With the coverage-based method, a sensor node is placed at a pixel where the total of the increase in coverage values is the highest. The weighted coverage-based method is similar to the coverage-based method, but it gives priority to pixels with a smaller coverage value. Finally, the void-based method aims at reducing the number of sensing voids.

To monitor the region as long as possible and prolong the lifetime of a WSN, there are two major approaches for the energy efficiency. One is limiting the transmission power of radio signals and the other is a sleep management, as stated above. From a viewpoint of coverage, it is a good idea to deploy more sensor nodes than the minimum requirement for k-coverage and then rotate a task of sensing among sensor nodes. Those sensor nodes without a sensing task can turn off unnecessary modules and sleep, while keeping k-coverage by actively sensing sensor nodes. In the second part of the thesis, we consider a sleep scheduling mechanism. We assume that a WSN is of K-coverage when all sensor nodes are awake and sensing and application's requirement is k-coverage where $k < K$. There have been many research works on sleep scheduling [6-11]. For example, in [6], they propose a sleep scheduling mechanism to satisfy the degree of coverage, which is different among areas. In their mechanism, all sensor node operate on the same control interval of T. Sensor node i becomes active and monitor its surroundings during $T_{front}+T_{end}$ around randomly determined timing of Ref_i in T. By exchanging information about Ref_i among neighboring sensor nodes, they can appropriately set Ref_i with which they can go to sleep alternatively while keeping the coverage. In [7], they propose a probabilistic scheme, called FCS (Fractional Coverage Scheme), where a sensor node does not need location information and decides whether to be active or not depending on the node density and the amount of residual energy. In [12], they propose a mechanism where a sensor node moves among on-duty, ready-to-off, off-duty, and ready-to-on states depending on the eligibility and timers. They propose the eligibility rule to determine whether a sensor node can sleep (eligible) or has to be active (ineligible). CCP (Coverage Configuration Protocol) [13, 14] is also a state-based mechanism, which consists of SLEEP, LISTEN, JOIN, ACTIVE, and WITHDRAW states. They propose K_s-coverage eligibility algorithm to decide whether to sleep (ineligible) or be active (eligible). In this thesis, we also consider a state-based mechanism, specifically, using the state transition diagram of CCP. However, to have more energy-efficient control, we take into account the amount of residual energy of sensor nodes in setting timers, which determines how long a sensor node stays in a state. For example, by making a sensor node with much residual energy awake longer than other poor sensor nodes, we can balance the energy consumption among sensor nodes and prolong the lifetime of a WSN. In addition, through analysis on a Markov chain model of the state transition of a sensor node, we investigate the relationship among the density of sensor nodes, timer setting, and the lifetime of a WSN.

The rest of the thesis is organized as follows. First in section 2, we propose three placement methods to satisfy the required k-coverage with the small number of additional sensor nodes. Then, we evaluate their performance through simulation experiments. Next in section 3, we propose a sleep scheduling mechanism where the amount of residual energy is taken into account in setting timer values. By comparing to CCP, we verify the effectiveness of our energy-dependent timer setting. Furthermore, we model the state transition as a Markov chain and show how to derive the steady state probability and the expected lifetime of a WSN. Finally, in section 4, we conclude this thesis and explain some future works.

2 Placement Methods of Additional Sensor Nodes to Satisfy k-Coverage

In this section, we propose three placement methods which determine the number and location of additional sensor nodes to achieve the required k-coverage. Our methods take different approaches to make the number of additional nodes as small as possible.

The assumptions for the initial condition are the same among methods. N sensor nodes are randomly distributed in the region of the size of $R_x \times R_y$ m^2. The geographical position of sensor node i, denoted as (x_i, y_i) can be obtained by a positioning device such as GPS (Global Positioning System) or other localization protocols [15]. The range of communication is R and the range of sensing is r. They are identical among all sensor nodes. By assuming that $2r \leq R$ holds for all nodes, satisfying the coverage guarantees the connectivity [13, 16, 17].

Placement methods adopt centralized algorithms. First, sensor nodes deployed into the region are all active to build a WSN. Next, they identify their positions and send position information to the server via a base station or a designated gateway node. Information transmission is done by some appropriate routing or data gathering protocol. Our methods does not depend on any specific communication protocol. Then, using collected information, the server evaluates the current coverage of the region and determines position of additional sensor nodes to obtain the target k-coverage.

2.1 Coverage on Sensor Network

A server first evaluates the degree of coverage of the region. The region of the size of $R_x \times R_y$ m^2 is first divided into $n \times m$ rectangle or square pixels as shown in Fig. 1, in which each square corresponds to a pixel. Consequently, the width of a pixel is $\Delta_x = R_x/n$ and the height of a pixel is $\Delta_y = R_y/m$. The coordinates of the center of a pixel at the column i and the row j $(0 \leq i \leq n-1, 0 \leq j \leq m-1)$, called pixel $p_{i,j}$, can be derived as $((i+1/2)\Delta_x, (j+1/2)\Delta_y)$ when the left bottom of the region is consider as the column 0 and the row 0. When N nodes are initially deployed in the region, we can evaluate the coverage value $c_{i,j}$ of pixel $p_{i,j}$ by using Algorithm 1 with the knowledge of position of sensor nodes. We should note here that the coverage value of pixel $p_{i,j}$ is equal to the

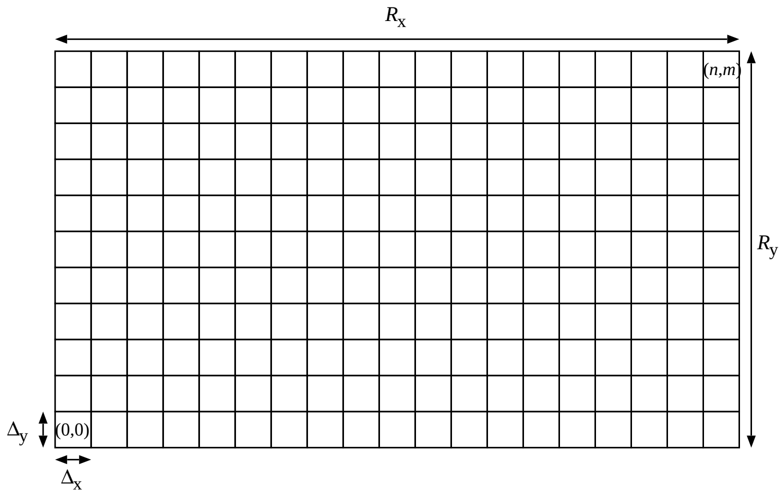


Figure 1: Region divided into pixels

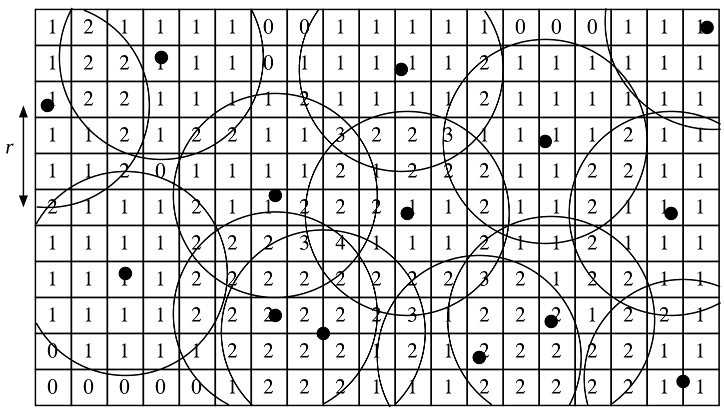


Figure 2: Coverage values

number of sensor nodes whose sensing area covers the center of the pixel. An example of coverage values is illustrated in Fig. 2, where each dot corresponds to a sensor node and an open circle centered at the dot indicates the sensing range of the node. The degree of coverage of the region is then given as $\min_{0 \le i \le n-1, 0 \le j \le m-1} c_{i,j}$.

2.2 Placement Methods

Now we describe our placement methods. First, we introduce the coverage-based method which puts an additional sensor node at such a pixel that the total of increase of coverage

Algorithm 1 Deriviation of coverage values

for $(i = 0; i \leq n-1; i++)$ **do**
  for $(j = 0; j \leq m-1; j++)$ **do**
    $c_{i,j} = 0;$
  end for
end for
for $(s = 1; s \leq N; s++)$ **do**
  for $(i = floor((x_s - r)/\Delta_x); i <= ceil((x_s + r)/\Delta_x); i++)$ **do**
    for $(j = floor((y_s - r)/\Delta_y); j <= ceil((y_s + r)/\Delta_y); j++)$ **do**
      if $(((i + 1/2) \times \Delta_x - s_x)^2 + ((j + 1/2) \times \Delta_y - s_y)^2 <= r^2)$ **then**
        $c_{i,j}++;$
      end if
    end for
  end for
end for

values becomes the highest among all possible pixels. Next, we describe the weighted coverage-based method in which a pixel of a smaller coverage value is covered before a pixel of a larger coverage value. Finally, the void-based method is proposed, which effectively reduces the number of sensing voids. In all methods, an additional node is placed at the center of a pixel.

2.2.1 Coverage-based Method

The gain $G_{i,j}^{(c)}$ of coverage value by putting one sensor node at a certain pixel $p_{i,j}$ can be derived as,

$$G_{i,j}^{(c)} = \sum_{c=0}^{k-1} n_{c \to c+1}, \tag{1}$$

where $n_{c \to c+1}$ corresponds to the number of pixels whose coverage value is increased from c to $c+1$ by being covered by the additional node. When we denote the number of pixels exist in the sensing area of a sensor node as n, $n = \sum_{c=0}^{\infty} n_{c \to c+1}$ holds. The coverage-based method puts additional sensor nodes one by one at such a pixel $p_{i,j}$ that leads to the maximum gain, i.e., $G_{i,j}^{(c)} = \max_{0 \leq k \leq n-1, 0 \leq l \leq m-1} G_{k,l}^{(c)}$.

2.2.2 Weighted Coverage-based Method

From a viewpoint of coverage, i.e., a demand for monitoring the region, uncovered pixels or pixels with a relatively small coverage value should be considered more important than, for example, pixels of coverage value of $k-1$. Therefore, the weighted coverage-based method introduces weight vector $\mathbf{w} = (w_0, w_1, \ldots, w_k)$ in calculating the gain $G_{i,j}^{(w)}$ as,

$$G_{i,j}^{(w)} = \sum_{c=0}^{k-1} w_c n_{c \to c+1}. \tag{2}$$

For example, weight vector $\mathbf{w}$ can be defined as,

$$w_i = \frac{1}{i+1}. \tag{3}$$

With weight vector $\mathbf{w} = (1, 1, 1, \ldots, 1)$, the weighted coverage-based method becomes equivallent to the coverage-based method.

2.2.3 Void-based Method

Due to the random deployment, there are sensing voids, i.e., sets of neighboring pixels whose coverage value is less than k. An example of sensing voids is illustrated in Fig. 3 as shaded pixels. The void-based method is different to the others in directly considering sensing voids. An additional sensor node is placed at a pixel so that the number of sensing voids decreases the most.

Algorithm 2 shows the pseudo code to identify sensing voids. The void-based method examines all pixels to find the pixel which leads to the highest gain. The gain $G_{i,j}^{(v)}$ of putting an additional node at pixel $p_{i,j}$ is derived as,

$$G_{i,j}^{(v)} = G_{i,j}^{(w)} + nD_{i,j}, \tag{4}$$

where n corresponds to the number of pixels in the sensing area of a sensor node and $D_{i,j}$ is the difference between the number of sensing voids before and after putting an additional sensor node at pixel $p_{i,j}$. This definition of gain implies that if the number of sensing voids cannot be decreased by putting an additional sensor node at any pixels, or, decrease in the number of sensing voids is the same among two or more pixels, one pixel is chosen based on the weighted gain $G_{i,j}^{(w)}$.

Algorithm 2 Finding sensing voids

for $(i = 0; i \leq n-1; i++)$ **do**
    for $(j = 0; j \leq m-1; j++)$ **do**
        $c_{i,j}.check = 0;$
        $c_{i,j}.value = 0;$
    end for
end for
void find_area (int i, int j, int c)
if ($c_{i,j}$.check $\neq 0$ or $c_{i,j}$.value $\neq 0$) **then**
    return
end if
$c_{i,j}$.check $= 1$
if $(i \geq 1)$ **then**
    find_area$(i-1, j, c)$
end if
if $(i < n-1)$ **then**
    find_area$(i+1, j, c)$
end if
if $(j \leq 1)$ **then**
    find_area$(i, j-1, c)$
end if
if $(j < m-1)$ **then**
    find_area$(i, j+1, c)$
end if

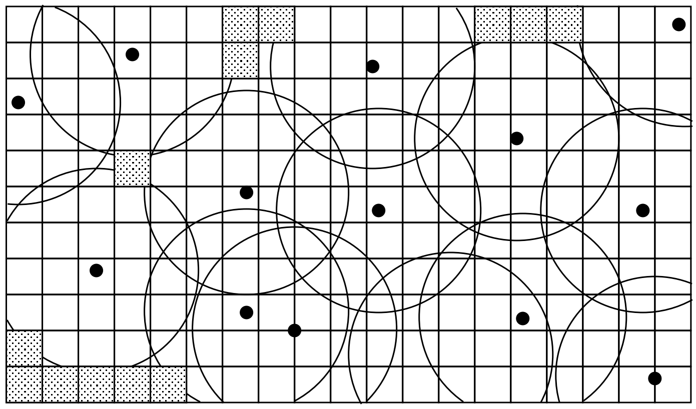


Figure 3: Sensing voids

2.3 Simulation Experiments

In this subsection, we compare three placement methods through simulation experiments. The region is of 50×50 m^2 and divided into 50×50 pixels of 1×1 m^2. Initially, 10 sensor nodes are randomly deployed in the region. The sensing range r is set at 10 m and identical among all sensor nodes. The required coverage k is set at 1, 2, 3, 4, or 5. To compare the effectiveness of placement, we consider the minimum coverage, coverage degree, and coverage ratio. The minimum coverage is the smallest coverage value among all pixels. This shows how fast sensing voids are covered. The coverage degree is the averaged coverage value, but coverage values exceeding k are regarded as k. Therefore, the coverage degree indicates how well the region is covered. Finally, the coverage ratio is the ratio of pixels whose coverage value is k or more to all 50×50 pixels. The coverage ratio shows how fast the required coverage k is satisfied. In the following figures, averaged values over 100 simulation experiments are shown.

Figures 4, 5, and 6 show results of the minimum coverage, coverage degree, and coverage ratio against the number of additional nodes, respectively, when the required coverage k is one. We should note here that for $k = 1$ the coverage degree and the coverage ratio are equivalent. In addition, the coverage-based and weighted coverage-based methods provide the same performance for $k = 1$. As shown in Fig. 4, the void-based method outperforms the others in covering uncovered pixels. This is contradicting intuitive expectation that the coverage-based and weighted coverage-based methods are better than the void-based for their greedy placement. Figure 7 explains the reason. The coverage-based and weighted coverage-based methods choose pixels for additional sensor nodes so that the number of newly covered pixels is the maximum. Consequently, there is possibility that an added sensor node fragments a sensing void into small pieces of sensing voids. Then, to cover those small sensing voids, redundant sensor nodes are required and the speed that all uncovered pixels are covered becomes slow. On the other hand, the speed of increase of the coverage degree and coverage ratio is faster for the coverage-based and weighted coverage-based methods than the void method as shown in Figs. 5 and 6. The void method first places additional sensor nodes at small sensing void to reduce the number of sensing voids effectively. As a result, the increase of the coverage degree and ratio is small. However, the average of required number of additional nodes for $k = 1$ is 13 with the void-based method and it is slightly smaller than the required number 14 for the other coverage-oriented methods. Depending on the initial condition of the region, including the size of the region, the number of initial sensor nodes, and the sensing range, the difference becomes larger.

Next, Figs. 8 through 10 show results of the case of $k = 2$. On average of 100 experiments, the required number of additional sensor nodes to satisfy 2-coverage is 31, 26, and 70 for the coverage-based, weighted coverage-based, and void-based methods, respectively. On the contrary to the results for $k = 1$, the void-based method requires the most additional sensor nodes among three methods. It is because that the void-based method considers regions of coverage values less than k are sensing voids equally, independently of their coverage values. Therefore, it keeps trying covering small sensing voids by putting many sensor nodes. One possible solution to this is to introduce a weight vector in deriving the number of sensing voids to make much of sensing void of small coverage values. We can also see that the weighted coverage-based method requires the smallest number of additional sensor nodes among three for the weighted gain $G_{i,j}^{(w)}$ in Eq. (2). In Figs. 11 through 19, results are shown for the cases of $k = 3$, 4, and 5 and Fig. 20 summarizes results of the required number of additional nodes against the required coverage k. For a comparison purpose, we also show results of the hexagonal method. In the hexagonal method, positions for additional nodes are predetermined and fixed as shown in Fig. 21 assuming the regular placement. Due to the regularity, the hexagonal method requires the

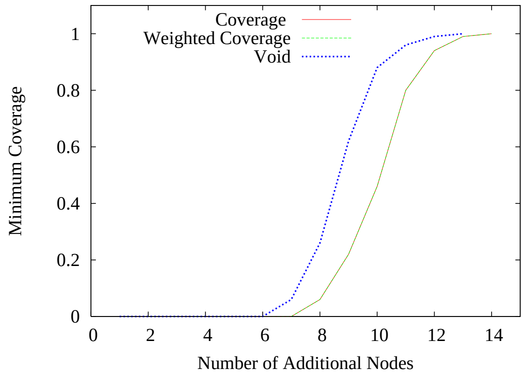


Figure 4: Additional nodes vs. minimum coverage (k=1, initial nodes=10)

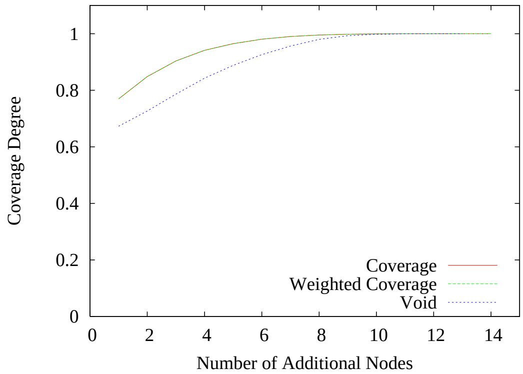


Figure 5: Additional nodes vs. coverage degree (k=1, initial nodes=10)

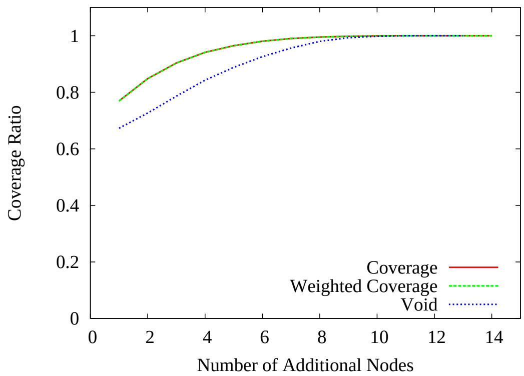


Figure 6: Additional nodes vs. coverage ratio (k=1, initial nodes=10)

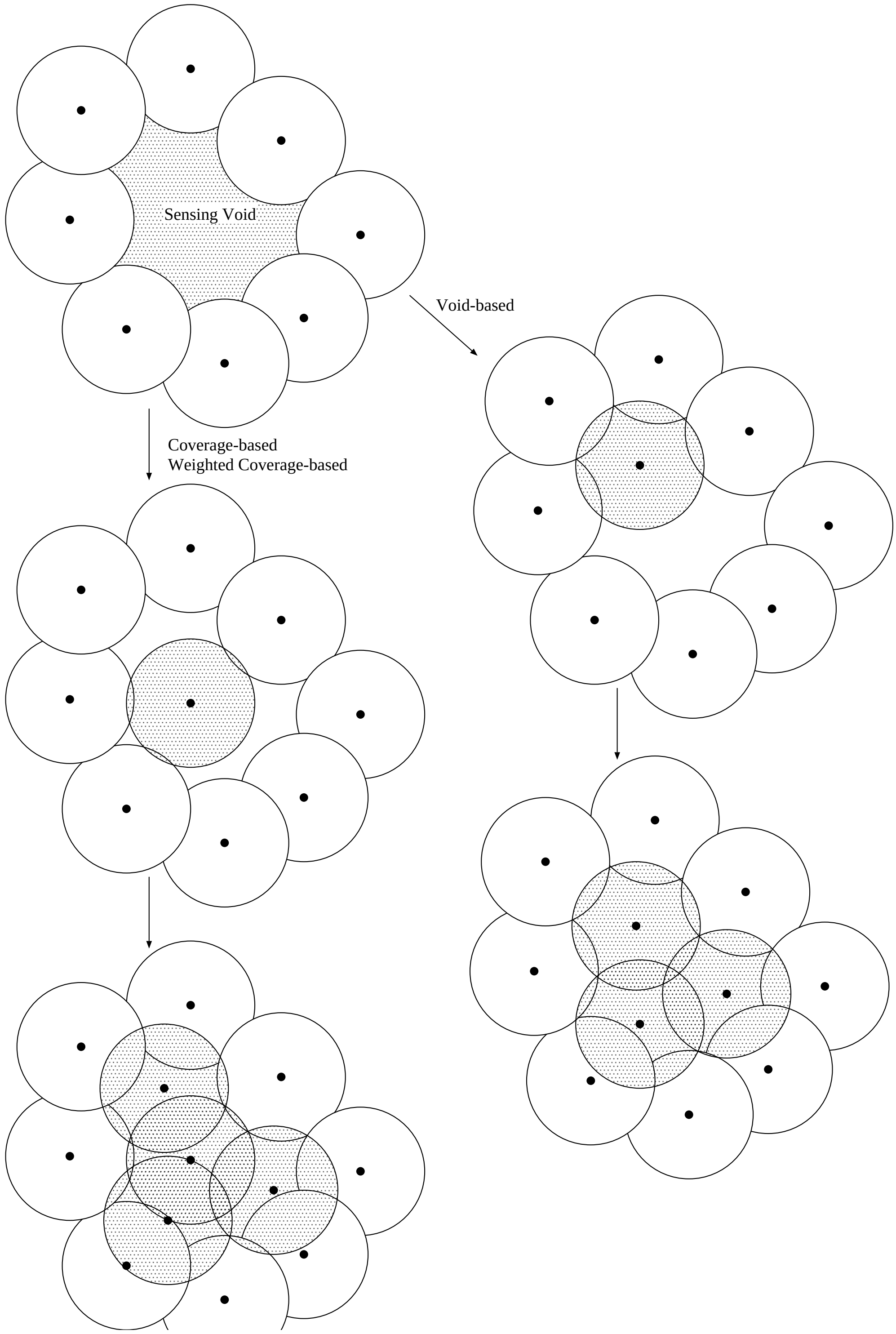


Figure 7: Coverage-based, weighted coverage-based, and void-based Methods

most additional nodes. The number of required additional nodes with the coverage-based and weighted coverage-based methods increases in proportional to the require coverage k.

Finally, results for the case where 50 sensor nodes are randomly deployed at first are shown in Figs. 22 through 34. Due to the higher density of sensor nodes, an initial WSN satisfies 1-coverage, where the minimum coverage value is 1 among all pixels. Instead of the initial density, methods show similar results to the cases of 10 initial nodes.

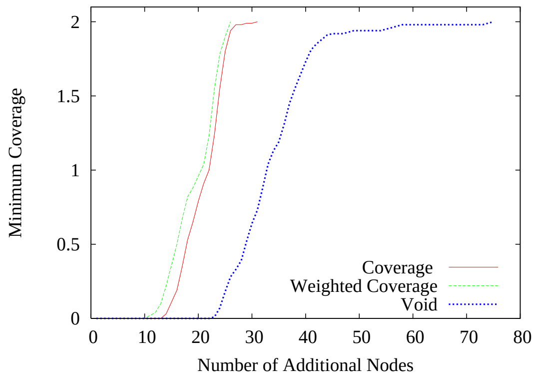


Figure 8: Addtional nodes vs. minimum coverage (k=2, initial nodes=10)

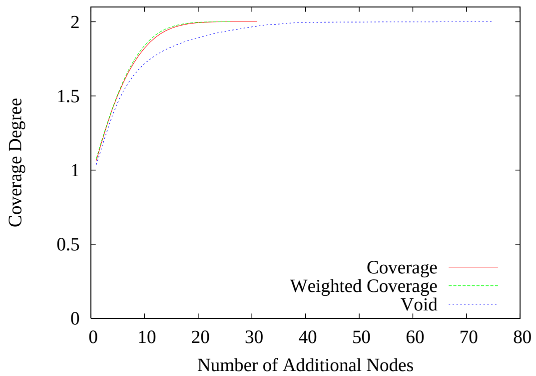


Figure 9: Additional nodes vs. coverage degree (k=2, initial nodes=10)

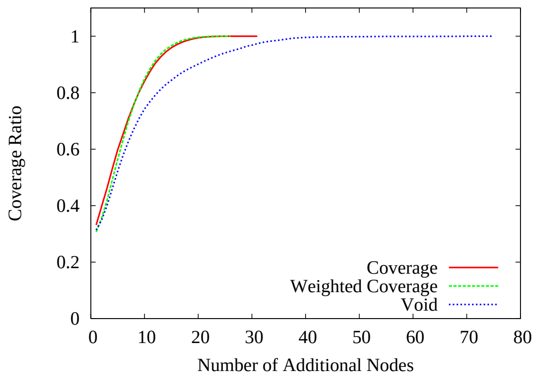


Figure 10: Additional nodes vs. coverage ratio (k=2, initial nodes=10)

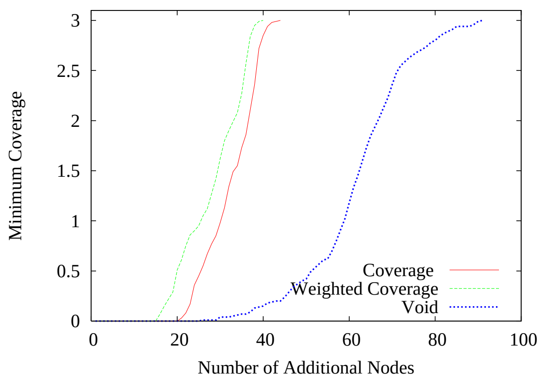


Figure 11: Addtional nodes vs. minimum coverage (k=3, initial nodes=10)

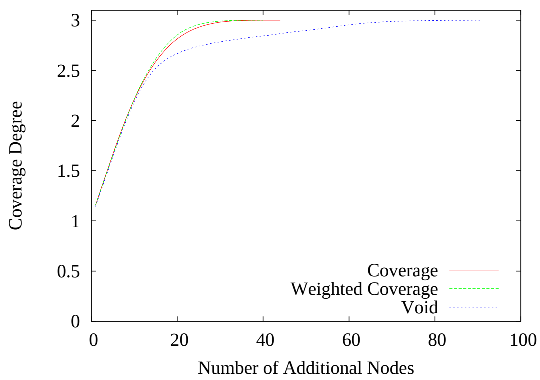


Figure 12: Additional nodes vs. coverage degree (k=3, initial nodes=10)

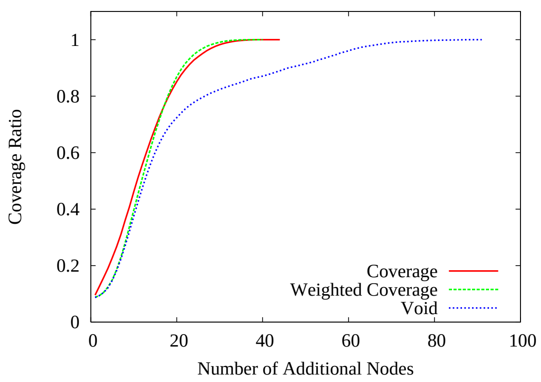
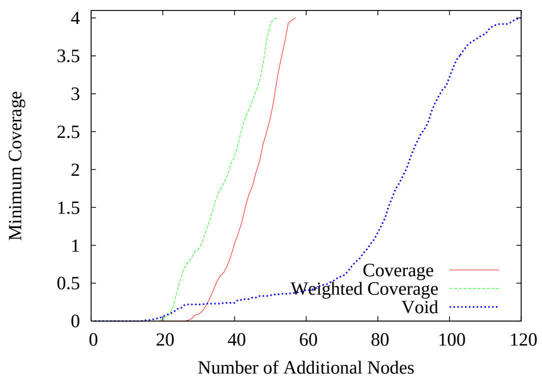


Figure 13: Additional nodes vs. coverage ratio (k=3, initial nodes=10)



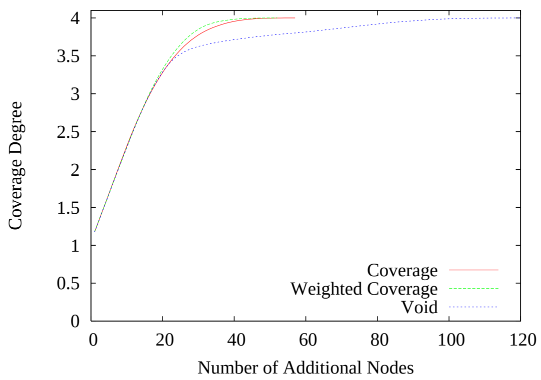


Figure 14: Addtional nodes vs. minimum coverage (k=4, initial nodes=10)

Figure 15: Additional nodes vs. coverage degree (k=4, initial nodes=10)

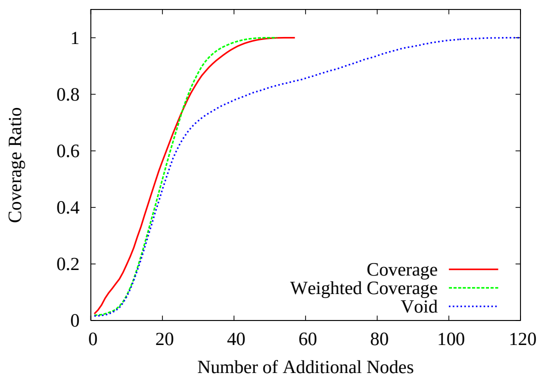


Figure 16: Additional nodes vs. coverage ratio (k=4, initial nodes=10)

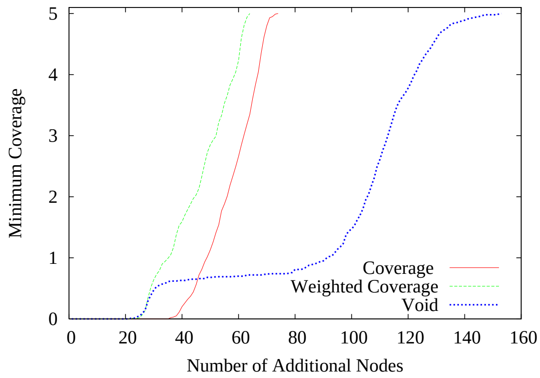


Figure 17: Addtional nodes vs. minimum coverage (k=5, initial nodes=10)

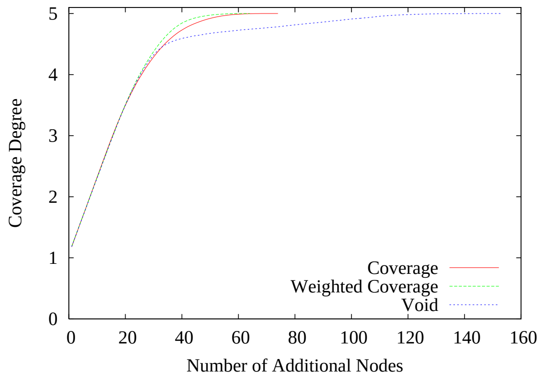


Figure 18: Additional nodes vs. coverage degree (k=5, initial nodes=10)

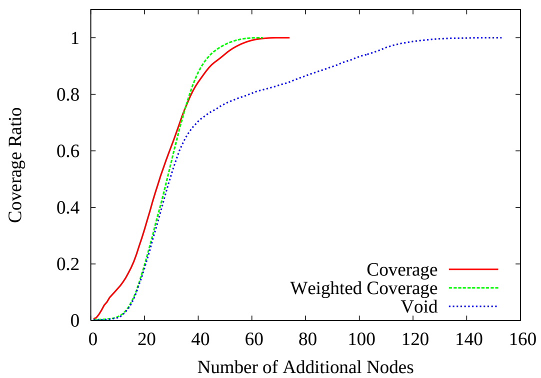


Figure 19: Additional nodes vs. coverage ratio (k=5, initial nodes=10)

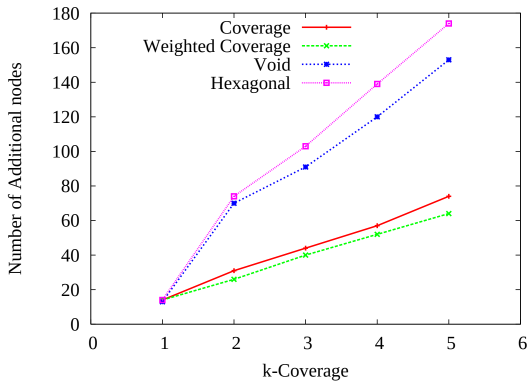


Figure 20: Required k-coverage vs. addtional nodes (initial nodes=10)

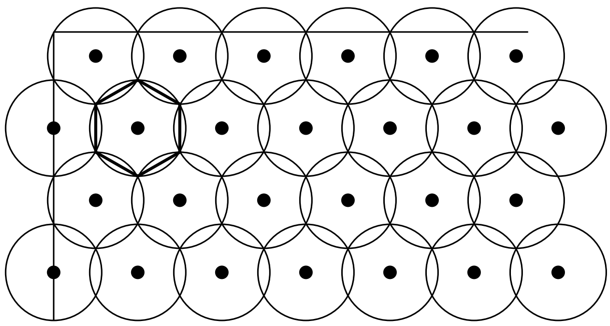
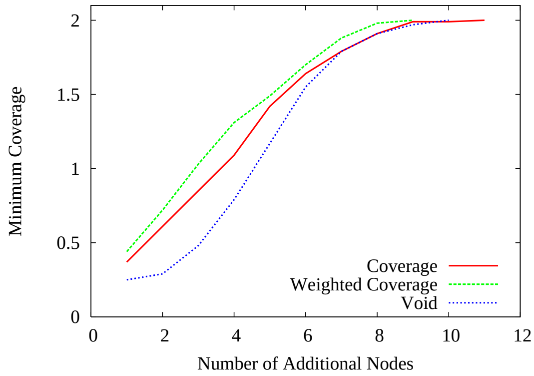


Figure 21: Hexagonal placement



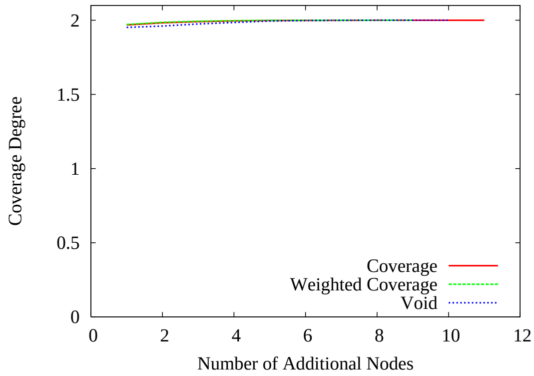


Figure 22: Addtional nodes vs. minimum coverage (k=2, initial nodes=50)

Figure 23: Additional nodes vs. coverage degree (k=2, initial nodes=50)

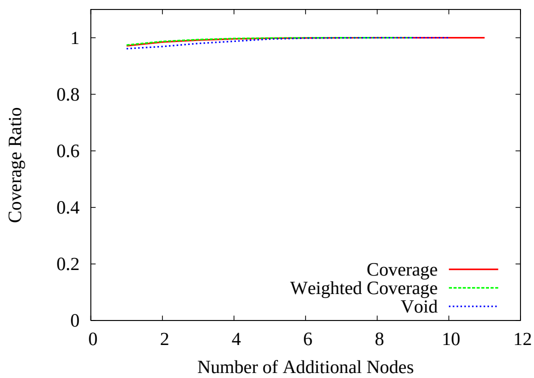


Figure 24: Additional nodes vs. coverage ratio (k=2, initial nodes=50)

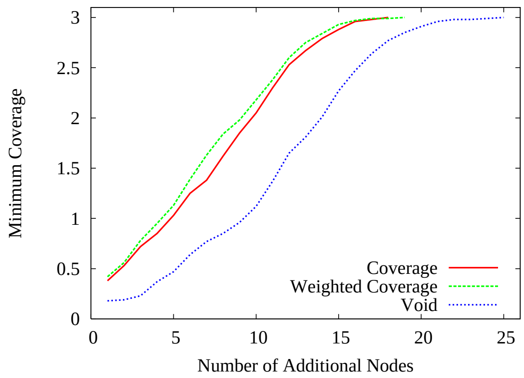


Figure 25: Addtional nodes vs. minimum coverage (k=3, initial nodes=50)

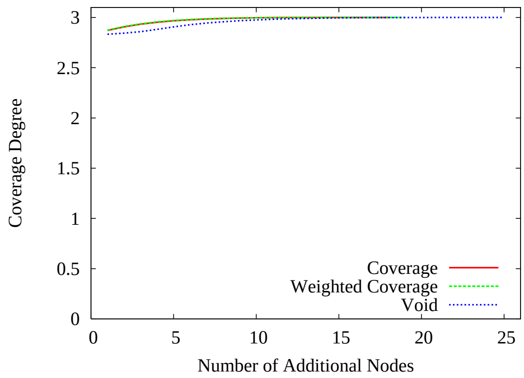


Figure 26: Additional nodes vs. coverage degree (k=3, initial nodes=50)

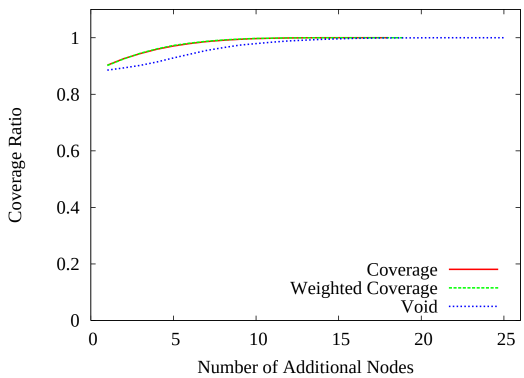


Figure 27: Additional nodes vs. coverage ratio (k=3, initial nodes=50)

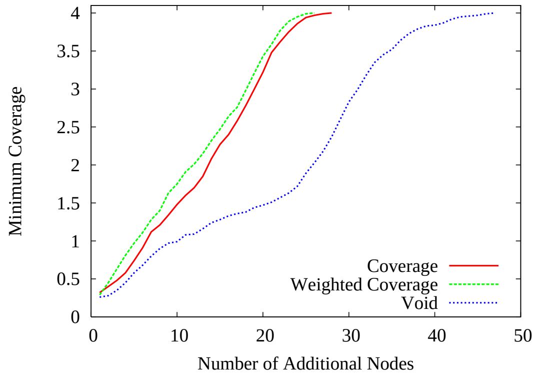


Figure 28: Addtional nodes vs. minimum coverage (k=4, initial nodes=50)

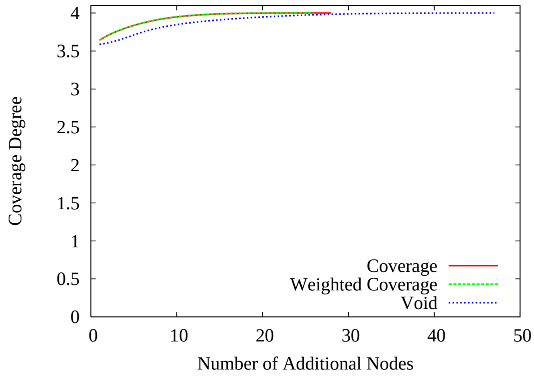


Figure 29: Additional nodes vs. coverage degree (k=4, initial nodes=50)

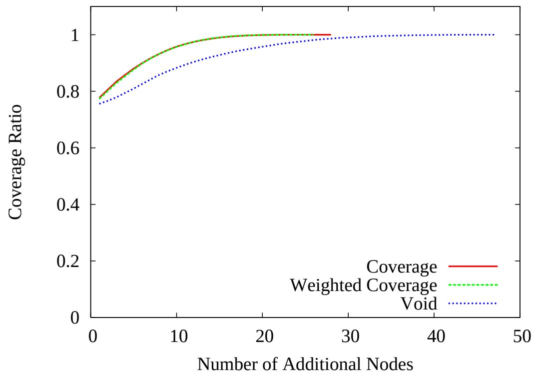
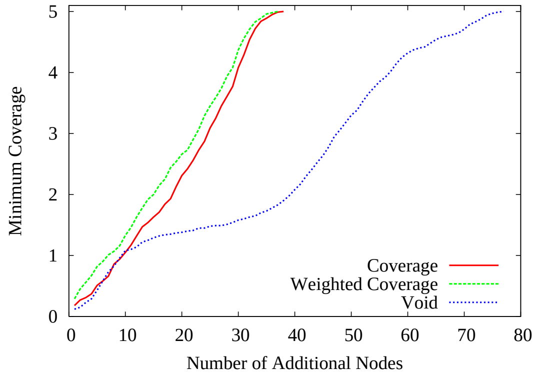


Figure 30: Additional nodes vs. coverage ratio (k=4, initial nodes=50)



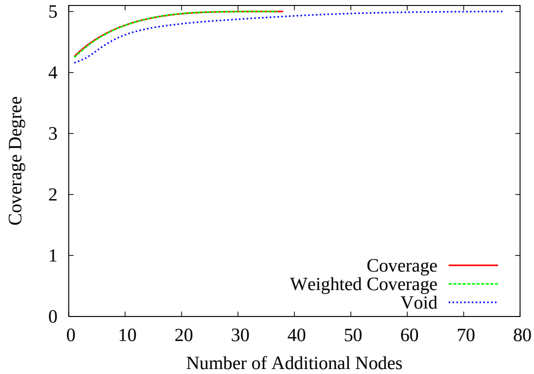


Figure 31: Addtional nodes vs. minimum coverage (k=5, initial nodes=50)

Figure 32: Additional nodes vs. coverage degree (k=5, initial nodes=50)

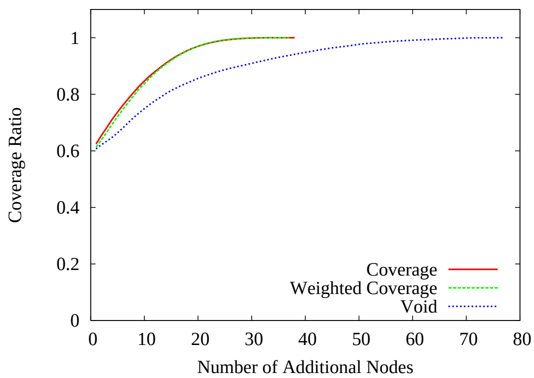


Figure 33: Additional nodes vs. coverage ratio (k=5, initial nodes=50)

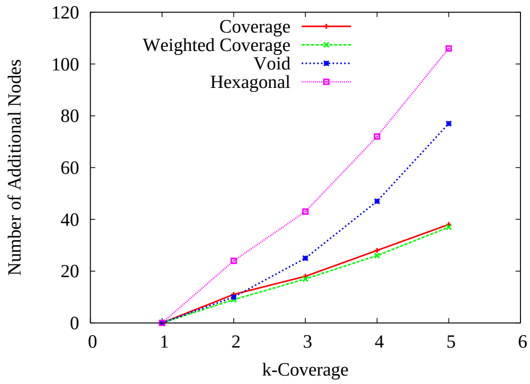


Figure 34: Required k-coverage vs. addtional nodes (initial nodes=50)

3 Energy-Efficient Sleep Scheduling Mechanism

In this section, we consider an energy-efficient sleep scheduling mechanism to prolong the lifetime of a WSN where the sufficient degree of coverage has already been satisfied by a placement method proposed in section 2.

A WSN has N sensor nodes and has K-coverage ($K \geq 2$). Application requires k-coverage, where $1 \geq k < K$ holds. Sensor node i ($1 \leq i \leq N$) is located at the geographic coordinate of (x_i, y_i). The range of communication is R, and range of sensing is r. We assume that the initial energy of a battery can be different among sensor nodes. We denote the initial energy of sensor node i as $e_i(0)$. When we consider a WSN deployed in a building, it is reasonable to assume the existence of power-supplied node, i.e., $e_i(0) = \infty$. A sleep scheduling mechanism should take into account the heterogeneity of the capacity of battery and efficiently make use of sensor nodes with sufficient residual energy or power supply. Sensing range r and communication range R are fixed and identical for all sensor nodes and $2r \leq R$ is assumed.

3.1 State transition of sensor node

A sensor node follows the state transition diagram illustrated in Fig. 35 [13, 14], depending on timers and its eligibility. Being eligible or "Eligibility=TRUE" means that a sensor node must be active and monitor around to satisfy the required k-coverage. On the other hand, ineligible sensor nodes can turn off unnecessary modules to avoid wasting the battery. The behavior of a sensor node in each state is given below. In all states except for the SLEEP state, a sensor node maintains a neighbor list. Every time it receives a message, the neighbor list is updated before checking the eligibility. A sender of a HELLO or JOIN message is added to the list and that of a WITHDRAW message is removed from the list. An entry of the list consists of an identifier, the coordinates, and the sensing range of a message sender.

- SLEEP: In entering the SLEEP state at time t, sensor node i clears the neighbor list and sets the sleep timer $T_s(i, t)$. In the SLEEP state, sensor node i turns off all modules except for the timer and saves the power consumption. When the timer expires at $t + T_s(i, t)$, sensor node i wakes up, turns on the transceiver, starts the

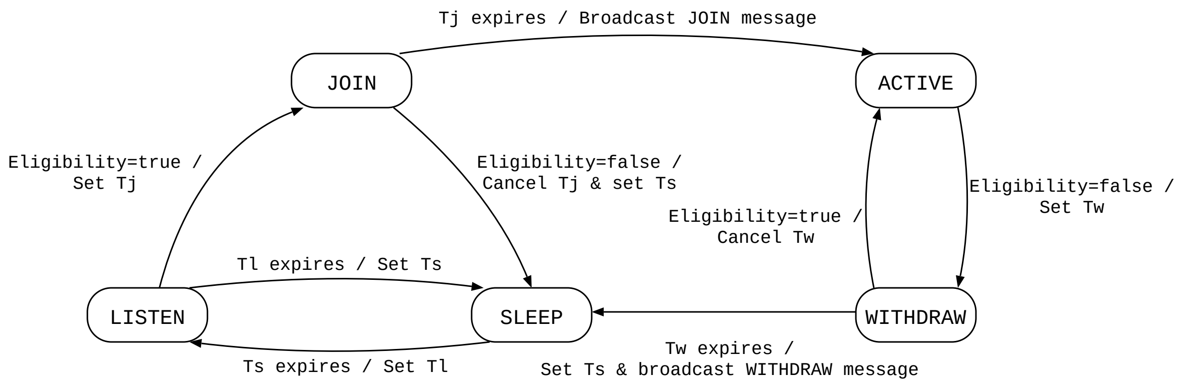


Figure 35: State transition diagram

listen timer, and moves to the LISTEN state.

- LISTEN: Entering the LISTEN state at time t, sensor node i sets the listen timer $T_l(i,t)$. In the LISTEN state, sensor node i waits for messages from neighboring sensor nodes. On receiving any of HELLO, WITHDRAW, or JOIN messages, it evaluates its eligibility. If it is eligible, i.e., required to be active, it cancels the listen timer, starts the join timer, and moves to the JOIN state immediately. When the listen timer expires at $t + T_l(i,t)$ without receiving any message, it considers that there is no sensor node in its vicinity. Then, it moves to the JOIN state while setting the join timer. Otherwise, on expiration of the listen timer, sensor node i moves back to the SLEEP state by setting the sleep timer.

- JOIN: Entering the JOIN state at time t, sensor node i sets the join timer $T_j(i,t)$. The JOIN state is an intermediate state to confirm its eligibility. On receiving a HELLO or JOIN message, it evaluates its eligibility. If it becomes ineligible before the join timer expires, it immediately cancels the join timer, starts the sleep timer, and moves to the SLEEP state. Otherwise, if the join timer expires at $t + T_j(i,t)$ and sensor node i is still eligible, it broadcasts a JOIN message to advertise that it is going to be active, and moves to the ACTIVE state.

- ACTIVE: Entering the ACTIVE state, sensor node i activates sensors to monitor its surroundings. In the ACTIVE state, a sensor node i takes appropriate actions, including sensing, transmission of sensor information or a message, evaluation of sensor information, and activation of modules, e.g., fire alarm, depending on application's requirements. In addition, sensor node i broadcasts HELLO messages at regular intervals of T_{hello}. The interval is short enough for neighboring sensor nodes in the LISTEN, JOIN, or WITHDRAW state to receive at least one HELLO message and recognize the existence of the sender. On receiving a HELLO or JOIN message from a neighboring sensor node, a sensor node evaluates its eligibility. If it considers itself as redundant, i.e., ineligible, it starts the withdraw timer and moves to the WITHDRAW state. Otherwise, if it is eligible, it stays in the ACTIVE state.

- WITHDRAW: Entering the WITHDRAW state at time t, sensor node i sets the withdraw timer $T_w(i,t)$. The WITHDRAW state is an intermediate state to confirm its ineligibility. When sensor node i receives a WITHDRAW message from a neighboring sensor node, it evaluates its eligibility. If it is no more ineligible, it cancels the withdraw timer and moves back to the ACTIVE state immediately. Otherwise, when the withdraw timer expires at $t+T_w(i,t)$ and sensor node i is still ineligible, it starts the sleep timer, broadcasts a WITHDRAW message to advertise its intention to sleep, and moves to the SLEEP state by turning off unnecessary modules.

In all states except for the SLEEP state, a sensor node evaluates its eligibility on receiving a message from a neighboring node in the range of radio signals. There is a variety of ways of evaluating the eligibility. The K_s-coverage eligibility algorithm proposed in [13, 14] takes a geometrical approach, where the degree of coverage of only intersection points of sensing ranges of neighboring sensor nodes are considered. If all of intersection points are in the sensing range of the sufficient number of active sensor nodes, a sensor node is considered ineligible. We also adopt the K_s-coverage eligibility algorithm.

3.2 Energy-dependent Timer Setting

The amount of energy consumed is different among states. Table 1 summarizes states of sensor, transmitter, and receiver modules in each state. e_{RX} and e_{TX} correspond to the amount of energy consumed in receiving and transmitting a message, respectively. e_{TX} includes the energy spent in the transmitter circuits. k_{RX} and k_{TX} correspond to the number of messages received and transmitted in the state. $e_{LISTENING}$ is for the

	sensor	transmitter	receiver	consumption
ACTIVE	on	on	on	$e_{RX}k_{RX} + e_{TX}k_{TX} + e_{SENSING}d_{ACTIVE}$
WITHDRAW	on	off	on	$e_{RX}k_{RX} + e_{TX} + e_{SENSING}d_{WITHDRAW}$
JOIN	off	off	on	$e_{RX}k_{RX} + e_{TX} + e_{LISTENING}d_{JOIN}$
LISTEN	off	off	on	$e_{RX}k_{RX} + e_{LISTENING}d_{LISTEN}$
SLEEP	off	off	off	$e_{SLEEP}d_{SLEEP}$

Table 1: State of modules in each state

amount of energy consumed in listening the wireless channel per time unit. $e_{SENSING}$ is for the amount of energy consumed in sensing and listening the wireless channel per time unit. We assume that a sensor node in the SLEEP state also consumes low level energy of e_{SLEEP} per time unit. $d_{statename}$ is the duration of time that a sensor node stays in the state. e_{TX} in the WITHDRAW and JOIN states are for a WITHDRAW and JOIN message, respectively.

Although the amount of energy consumed by modules fully depends on devices and it is considerablly different among them [18], it is apparently true that the SLEEP state consumes much less energy than the other states.

To prolong the lifetime of a WSN and monitor the region as long as possible with limited battery power supply, it is necessary to set timers in accordance with the amount of residual energy. For example, by keeping a sensor node with more residual energy active, low-power sensor nodes having overlapping sensing area can save battery consumption by being ineligible and sleeping. Therefore, in this thesis, we propose energy-dependent timer setting. For this purpose, we assume that a sensor node knows the amount of residual energy of its battery and HELLO, WITHDRAW, and JOIN messages contain the information about the residual energy of a sender at the timing of message emission. At time t, on moving from one state to another, sensor node i sets its timer depending on energy information in all messages received while it is awake, i.e., in the LISTEN, JOIN, ACTIVE, and WITHDRAW states. The maximum among all residual energy is denoted as $e_{max}(t)$ and derived as,

$$e_{max}(t) = \max_{j \in \mathbf{S}} e_j(t_j), \tag{5}$$

where **S** is a set of sensor nodes from which sensor node i has received messages while it is awake, $t_j < t$ corresponds to the time that the latest message from sensor node j was issued, and $e_j(t)$ is for the amount of residual energy of sensor node j at time t.

The sleep timer $T_s(i,t)$ is set in inverse proportional to the relative amount of residual energy, so that a sensor node with the large amount of residual energy wakes up earlier than the others. $T_s(i,t)$ is given as,

$$T_s(i,t) = \frac{T_s(0)}{e_i(t)/e_{max}(t)}, \tag{6}$$

where $T_s(0)$ is the initial setting of the sleep timer.

Next, the withdraw timer $T_w(i,t)$ is set in proportional to the relative amount of residual energy. By keeping a sensor node with the large amount of residual energy stay in the WITHDRAW state longer, the sensor node has more chances to receive WITHDRAW messages from others and thus it is more likely to become eligible again. $T_w(i,t)$ is given as,

$$T_w(i,t) = e_i(t)/e_{max}(t) \times T_w(0), \tag{7}$$

where $T_w(0)$ is the initial setting of the withdraw timer.

Then, the join timer $T_j(i,t)$ is set in inverse proportional to the relative amount of residual energy. By making the join timer expire faster for a sensor node with the large amount of residual energy, it suppresses activation of sensor nodes with less residual energy. Therefore,

$$T_j(i,t) = \frac{T_j(0)}{e_i(t)/e_{max}(t)}, \tag{8}$$

where $T_j(0)$ is the initial setting of the join timer.

Finally, the listen timer $T_l(i,t)$ is set independently of the residual energy. The duration of the LISTEN state should be long enough to receive messages of neighboring nodes. Among messages, HELLO messages are the most important for sensor node i to evaluate its eligibility, since HELLO messages imply that there are actively sensing nodes in the vicinity and it would make the sensor node ineligible. Consequently, $T_l(i,t)$ is given as,

$$T_l(i,t) = T_l(0) = a \times T_{hello}, \tag{9}$$

where $T_j(0)$ is the initial setting of the listen timer and a is a constant and greater than 1 ($a \geq 1$).

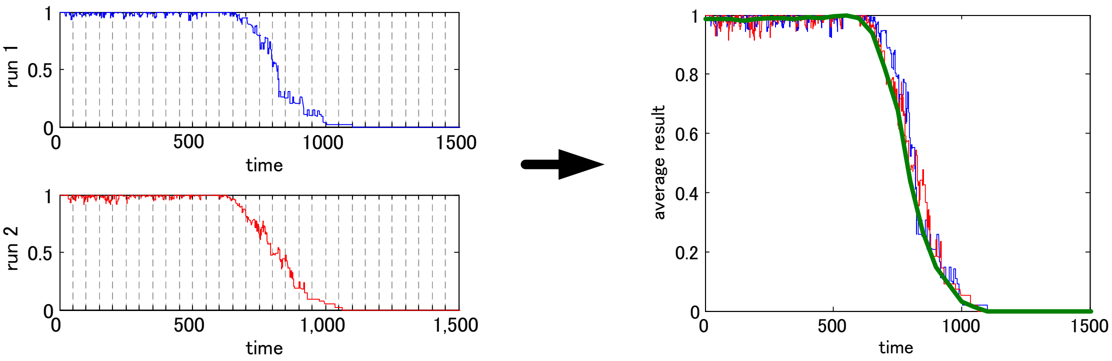


Figure 36: Method of obtaining average values over different simulation runs

3.3 Preliminary Simulation Experiments

In this subsection, we conduct preliminary simulation experiments to verify the effectiveness of energy-dependent timer setting. The region is of 50×50 m^2 and divided into 50×50 pixels of 1×1 m^2. Initially, sensor nodes are deployed in the region to satisfy 5-coverage, whose distribution is obtained by the weighted coverage-based method. The sensing range r is set at 12 m and identical among sensor nodes. The transmission range R is set at 24 m and identical among sensor nodes. The required coverage k is set at 3. We evaluate the coverage degree and coverage ratio against time. In addition, the average and standard deviation of the amount of residual energy are also considered.

We first focus on the effect of energy-dependent setting of the sleep timer $T_s(i,t)$, which is defined by Eq.(6). The initial values of timers are, $T_s(0) = 10$, $T_w(0) = 10$, $T_j(0) = 5$, $T_l(0) = 2T_{hello}$, and $T_{hello} = 2$. We assume that all sensor nodes are in the ACTIVE state at the beginning of a simulation run, but we introduced a random wait following the exponential distribution of average 2 before emission of the first HELLO message to avoid synchronized behavior. The initial amount of residual energy is set at 10000 energy units for all sensor nodes. For the energy consumption model, $e_{RX} = 7$, $e_{TX} = 8$, $e_{SENSING} = 10$, $e_{LISTENING} = 5$, and $e_{SLEEP} = 0.01$. In the following figures, the averaged value over two simulation experiments are shown.

In order to compare the resulting performance metrics obtained by simulation, we must at this point describe the methodology of averaging results. The traces from simulation are simply taken at the occurrence of each event, e.g., reception of a message. However, when more than one simulation run is considered it is not easily possible to obtain average values since the events occur at different time instants in each simulation. We, therefore, must preprocess the data to obtain the statistical quantities we wish to examine.

Let us assume that simulation run 1 has the event instants at t_i, $i = 1, \ldots, M$ and simulation run 2 at t'_j, $j = 1, \ldots, M'$. We now discretize the time axis into units of a certain window size τ. Since several events may fall into a τ window, depending on its size, we average over all values v_i at t_i, $k\tau \leq t_i < (k+1)\tau$. Resulting from this, we have $\bar{v}_k$ and $\bar{v}'_k$, $k = 1, \ldots, \min(\lceil \frac{t_M}{\tau} \rceil, \lceil \frac{t'_{M'}}{\tau} \rceil)$ which can now be related to each other over the same time interval which they both represent. When we have several runs available, we can find the mean values over all runs by using this method and obtain a "smooth" curve as shown in Fig. 36, where the thick green curve shows the smoothed average.

In Figs. 37, 38, 39, results are shown for the proposal and CCP. As shown in the figures, the energy-dependent sleep timer setting does not necessarily lead to the longer lifetime of a WSN. The reason of sharp decrease of the coverage degree and coverage ratio observed for the proposal is that sensor nodes are likely to deplete their battery at the same time due to the balanced energy consumption verified in Fig. 39. We also conducted several experiments by changing simulation setting including timers and sensing range. We found that the performance heavily depended on setting of initial timers for the node density. In addition, the energy-dependent timer setting proposed in subsection 3.2 is not necessarily optimal for its simplicity. Therefore, in the next subsection, we build a mathematical model to analysis the relationship among timer setting, the node density, and the lifetime.

3.4 Analysis of Energy-dependent Timer Setting

Although the energy-dependent setting of timers can balance the residual energy among sensor nodes as shown in section 3.3, appropriate setting of timers depends not only on the amount of residual energy, but also on the density of sensor nodes and it is very hard to determine. In this subsection, we analytically investigate the relationship among timer setting, the density, and the resultant lifetime of a WSN by regarding the state transition as a Markov chain [9, 19].

We first divide the state transition diagram of Fig. 35 into two state transition diagrams depending on the eligibility. An eligible sensor node follows the state transition illustrated

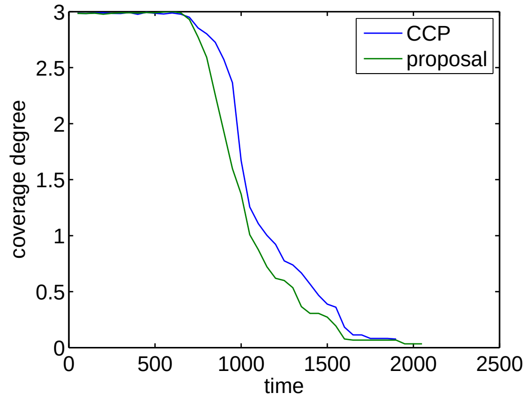


Figure 37: Comparison of coverage degree

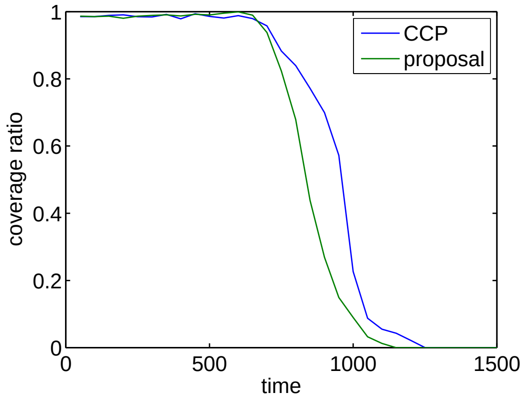


Figure 38: Comparison of coverage ratio

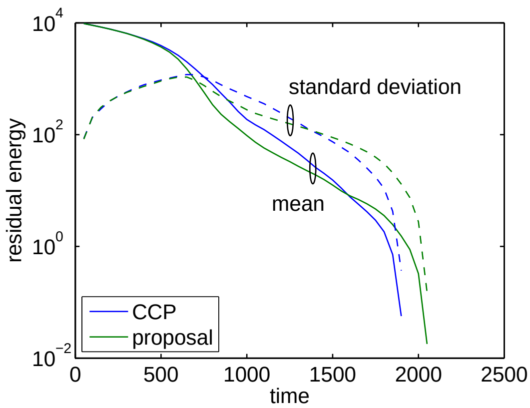


Figure 39: Comparison of residual energy

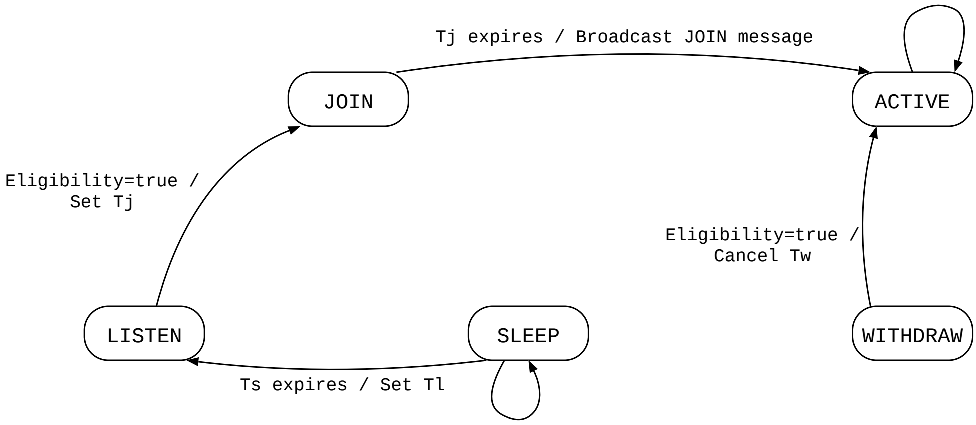


Figure 40: State transition diagram (eligible)

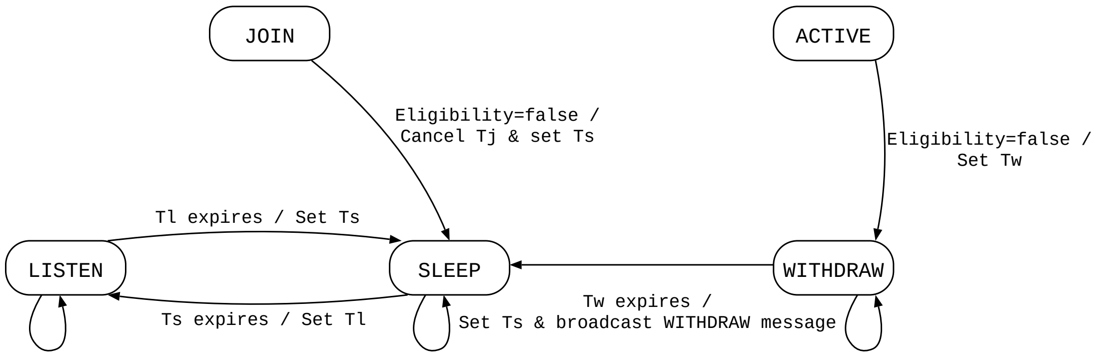


Figure 41: State transition diagram (ineligible)

in Fig. 40 and an ineligible sensor node follows the transition in Fig. 41. Then, by assuming the exponential distribution of timers and identical nodes, we translate the diagrams into the Markov transition probability matrices $\mathbf{M}$ in Eq. 10 and $\bar{\mathbf{M}}$ in Eq. 11. In the matrices, L, J, A, W, and S stand for the LISTEN, JOIN, ACTIVE, WITHDRAW, and SLEEP states, respectively.

$$\mathbf{M} = \begin{array}{c|ccccc} & L & J & A & W & S \\ \hline L & 0 & 1 & 0 & 0 & 0 \\ J & 0 & 1-\frac{1}{T_j} & \frac{1}{T_j} & 0 & 0 \\ A & 0 & 0 & 1 & 0 & 0 \\ W & 0 & 0 & 1 & 0 & 0 \\ S & \frac{1}{T_s} & 0 & 0 & 0 & 1-\frac{1}{T_s} \end{array} \tag{10}$$

$$\bar{\mathbf{M}} = \begin{array}{c|ccccc} & L & J & A & W & S \\ \hline L & 1-\frac{1}{T_l} & 0 & 0 & 0 & \frac{1}{T_l} \\ J & 0 & 0 & 0 & 0 & 1 \\ A & 0 & 0 & 0 & 1 & 0 \\ W & 0 & 0 & 0 & 1-\frac{1}{T_w} & \frac{1}{T_w} \\ S & \frac{1}{T_s} & 0 & 0 & 0 & 1-\frac{1}{T_s} \end{array} \tag{11}$$

When the state probability of a sensor node at time t is denoted as $\mathbf{p}(t)$, the state probability at time $t+1$ is given by,

$$\mathbf{p}(t+1) = \mathbf{p}(t)[P_E\mathbf{M} + P_I\bar{\mathbf{M}}], \tag{12}$$

where $\mathbf{p}(t) = [p_L(t), p_J(t), p_A(t), p_W(t), p_S(t)]$. $p_L(t)$, $p_J(t)$, $p_A(t)$, $p_W(t)$, and $p_S(t)$ are the state probability of the LISTEN, JOIN, ACTIVE, WITHDRAW, and SLEEP states, respectively. P_E and P_I are probability that a sensor node is eligible and ineligible, respectively, and $P_E + P_I = 1$ holds. Then, the steady state probability $\mathbf{p^s} = [p_L, p_J, p_A, p_W, p_S]$ can be derived by solving the following equations.

$$\mathbf{p^s} = \mathbf{p^s}[P_E\mathbf{M} + P_I\bar{\mathbf{M}}] \tag{13}$$

$$1 = \mathbf{p^s} \cdot \mathbf{1} \tag{14}$$

Now consider P_I. P_I is the probability that the sensing area of a sensor node is fully covered by the sensing area of more than k sensor nodes in the ACTIVE state. It can further be interpreted as the probability that "for each of points in the sensing area of a sensor node, there are k and more sensor nodes in the ACTIVE state in a circle of radius r centered at the point". When N nodes are randomly deployed in the region of size S, the node density D is N/S and the average number of sensor nodes in the sensing area of radius r is $D\pi r^2$. Therefore, the probability $p(n)$ that there are n nodes in a circle of radius r can be derived as,

$$p(n) = \frac{(D\pi r^2)^n}{n!} \exp(-D\pi r^2). \tag{15}$$

From this, the probability $q(k)$ that there are k or more nodes in a circle of radius r centered at a certain point can be derived as,

$$q(k) = \sum_{n=k}^{N-1} p(n) \sum_{i=k}^{n} {p_A}^i (1-p_A)^{n-i}, \tag{16}$$

where p_A is the probability that a sensor node is in the ACTIVE state.

The lifetime of a WSN of sensor nodes with the identical battery capacity can be derived as follows. First, for the initial amount of residual energy $E(0)$, timers have the initial value from Eqs.(6) through (9), where $T_s(i,0) = T_s(0)$, $T_w(i,0) = T_w(0)$, $T_j(i,0) = T_j(0)$, and $T_l(i,0) = aT_{hello}$. From this, the Markov probability transition matrices M and $\bar{M}$ are derived. Then, P_E and P_I are obtained from an arbitrary steady state probability $\mathbf{p^s}(0)$. By solving Eqs.(14), we can derive the steady state probability $\mathbf{p^s}(1)$, where 1 stands for the number of iteration. From $\mathbf{p^s}(1)$, P_E and P_I are calculated again. And then, Eqs.(14) gives next steady state probability $\mathbf{p^s}(2)$. By repeating this until $\mathbf{p^s}(n)$ converges, where $|\mathbf{p^s}(n-1) - \mathbf{p^s}(n)| < \epsilon$, we can finally have the steady state probability at time 0. Assuming that the amount of energy consumed in each state as λ_L, λ_J, λ_A, λ_W, and λ_S, respectively, the amount of energy $E(i,0)$ that sensor node i spends in the first time step can be derived as,

$$E(i,0) = \lambda_L p_L + \lambda_J p_J + \lambda_A p_A + \lambda_W p_W + \lambda_S p_S. \tag{17}$$

Therefore, the amount of residual energy of sensor node i at time $t = 1$ is $E(0) - E(i,0)$. Since the amount of residual energy is always the same among sensor nodes under the homogeneous condition, the lifetime of a WSN can be derived as,

$$Lifetime = \frac{E(0)}{E(1,0)}. \tag{18}$$

Therefore, under the homogeneous condition, the lifetime depends on the density of sensor nodes and the initial setting of timers.

If we can mathematically derive an appropriate set of timer setting as a function of the node density ρ like,

$$\mathbf{T}(\rho) = [T_s(0,\rho), T_l(0,\rho), T_j(0,\rho), T_w(0,\rho)], \tag{19}$$

where $T_{\{s|l|j|w\}}(0,\rho)$ are initial values depending on the density ρ, we can have an density-dependant and energy-efficient sleep scheduling mechanism. In the mechanism, sensor node i conjectures the node density ρ_i in its vicinity by receiving messages from neighboring sensor nodes in the range of radio signals. Then, by assuming that the whole region is of the same density, sensor node i determines the timers $\mathbf{T}(\rho_i)$ by using Eq. 19, which leads to the maximum lifetime of a sensor network of the density ρ_i. It may fall into the local optimal, but it should make the lifetime longer than a mechanism using the identical setting among sensor nodes. In addition, this adaptive setting mechanism can avoid the problem of selection of the initial values and adapt to dynamic changes in a WSN by addition, removal, and movement of sensor nodes.

4 Conclusion

In this thesis, to tackle the coverage problem, we first proposed three placement methods for achieving the required coverage with the small number of additional sensor nodes. Through simulation experiments, we showed that the void-based method was the best for 1-coverage, but the weighted coverage-based method required the minimum number of additional sensor nodes for more than 2-coverage requirement. In the second part of the thesis, we proposed an energy-efficient sleep scheduling mechanism to prolong the lifetime of a WSN by setting timers in accordance with the amount of residual energy. By simulation experiments, we verified that the energy-dependent timer setting led to the longer lifetime of a WSN. Then, to investigate the relationship among timer setting, the node density, and the lifetime, we next built a Markov model and showed the way to derive the lifetime from given timer setting and the node density.

As future research works, we will further consider an energy-efficient sleep scheduling mechanism based on the analytical results, in which a sensor node can appropriately determine timer setting from the local observation. In addition, we plan to implement the proposed mechanism using off-the-shelf sensor nodes to verify the practicality and applicability of the proposal.

Acknowledgement

First, I thank my supervisor Professor Masayuki Murata of Osaka University, for his continuous support through my studies and preparation of this thesis.

And also I would express my sincere appreciation to Professor Makoto Imase of Osaka University, for his guide and comments in the meeting.

Special thanks go to my co-advisor, Associate Professor Naoki Wakamiya of Osaka University. I sincerely appreciate him to greatly improve my work and help me in writing the thesis. I also would like to thank to Kenji Leibnitz, Specially Appointed Associate Professor of Osaka University. He gave me invaluable helps on my research experiments.

I am also grateful to Professors Koso Murakami, Teruo Higashino, Hirotaka Nakano, and Tetsuji Satoh of Osaka University, for their appropriate guidance and invaluable first-hand advice.

Finally, I would like to express my sincere thanks to all members of the Department of Information Networking, Graduate School of Information Science and Technology of Osaka University for their support. Our conversations and cooperative works have greatly influenced this thesis.

References

[1] M. Cardei and J. Wu, *Handbook of Sensor Networks*, ch. Coverage in Wireless Sensor Networks. CRC Press, 2004.

[2] S. Meguerdichian, F. oushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings of IEEE INFOCOM 2001*, pp. 1380–1387, April 2001.

[3] H. Durrant-Whyte, "Data fusion in sensor networks," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, p. 2, April 2005.

[4] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann Pub, 2004.

[5] L. Wang and Y. Xiao, "A survey of energy-efficient scheduling mechanisms in sensor networks," *Mobile Networks and Applications*, vol. 11, pp. 723–740, October 2006.

[6] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of ACM 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pp. 51–62, November 2003.

[7] M. Ye, E. Chan, G. Chen, and J. Wu, "Energy efficient fractional coverage schemes for low cost wilress snesor networks," in *Proceedings of the 26th IEEE Internatioal Conference Workshop on Distributed Computing Systems (ICDCSW'06)*, pp. 79–84, July 2006.

[8] H. O. Sanli and H. Cam, "Energy efficient differentiable coverge service protocols for wireless sensor networks," in *Proceedings of Third IEEE International Conference on Pervasive Computing and Communications Workshop (PERCOMW'05)*, pp. 406–410, March 2005.

[9] M. Inanc, M. Magdon-Ismail, and B. Yener, "Power optimal connectivity and coverage in wireless sensor networks," Tech. Rep. TR03-06, RPI Computer Science Technical Report, July 2003.

[10] R. Zheng, G. He, and X. Liu, "Location-free coverage maintenance in wireless sensor networks," Tech. Rep. UH-CS-05-15, Department of Computer Sience, University of Houston, July 2005.

[11] B. Liu and D. Towsley, "On the coverage and detectability of large-scale wireless sensor networks," in *Proceedings of the Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks Conference (WiOpt 2003)*, March 2003.

[12] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of ACM Wireless Sensor Network and Application Workshop 2002*, pp. 32–41, September 2002.

[13] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of ACM 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pp. 28–39, November 2003.

[14] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, pp. 36–72, August 2005.

[15] G.-L. Sun and W. Guo, "Comparison of distributed localization algorithms for sensor networks with a mobile beacon," in *Proceedings of IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 536–540, March 2004.

[16] D. Tian and N. D. Georganas, "Connectivity maintenance and coverage preservation in wireless sensor networks," in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 1097–1100, May 2004.

[17] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," in *Proceedings of NSF International Workshop on Theoretical and*

Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, February 2004.

[18] J. Polastre, R. Szewczyk, C. Sharp, and D. Culler, "The mote revolution: Low power wireless sensor network devices," in *Proceedings of Hot Chips 16: A Symposium on High Performance Chips*, August 2004.

[19] M. Magdon-Ismail, F. Sivrikaya, and B. Yener, "Joint problem of power optimal connectivity and coverage in wireless sensor networks," *Wireless Networks, Springer*, June 2006.