# TOPOLOGY MANAGEMENT PROTOCOLS IN AD HOC WIRELESS SENSOR NETWORKS

A Dissertation

by

HO GIL KIM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2007

Major Subject: Computer Science

# TOPOLOGY MANAGEMENT PROTOCOLS IN AD HOC WIRELESS SENSOR NETWORKS

A Dissertation

by

HO GIL KIM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

| | |
|---|---|
| Chair of Committee, | Eun Jung Kim |
| Committee Members, | Valerie E. Taylor |
| | Rabi N. Mahapatra |
| | Deepa Kundur |
| Head of Department, | Valerie E. Taylor |

December 2007

Major Subject: Computer Science

# ABSTRACT

Topology Management Protocols in Ad Hoc Wireless Sensor Networks.

(December 2007)

Ho Gil Kim, B.S., Korea Military Academy, Korea;

M.S., Younsei University, Korea

Chair of Advisory Committee: Dr. Eun Jung Kim

A wireless sensor network (WSN) is comprised of a few hundred or thousand autonomous sensor nodes spatially distributed over a particular region. Each sensor node is equipped with a wireless communication device, a small microprocessor, and a battery-powered energy source. Typically, the applications of WSNs such as habitat monitoring, fire detection, and military surveillance, require data collection, processing, and transmission among the sensor nodes. Due to their energy constraints and hostile environments, the main challenge in the research of WSN lies in prolonging the lifetime of WSNs.

In this dissertation, we present four different topology management protocols for $K$-coverage and load balancing to prolong the lifetime of WSNs.

First, we present a Randomly Ordered Activation and Layering (ROAL) protocol for $K$-coverage in a stationary WSN. The ROAL suggests a new model of layer coverage that can construct a $K$-covered WSN using the layer information received from its previously activated nodes in the sensing distance. Second, we enhance the fault tolerance of layer coverage through a Circulation-ROAL (C-ROAL) protocol. Using the layer number, the C-ROAL can activate each node in a round-robin fashion during a predefined period while conserving reconfiguration energy. Next, Mobility Resilient Coverage Control (MRCC) is presented to assure $K$-coverage in the presence

of mobility, in which a more practical and reliable model for $K$-coverage with nodal mobility is introduced. Finally, we present a Multiple-Connected Dominating Set (MCDS) protocol that can balance the network traffic using an on-demand routing protocol. The MCDS protocol constructs and manages multiple backbone networks, each of which is constructed with a connected dominating set (CDS) to ensure a connected backbone network. We describe each protocol, and compare the performance of our protocols with Dynamic Source Routing (DSR) and/or existing $K$-coverage algorithms through extensive simulations.

The simulation results obtained by the ROAL protocol show that $K$-coverage can be guaranteed with more than 95% coverage ratio, and significantly extend network lifetime against a given WSN. We also observe that the C-ROAL protocol provides a better reconfiguration method, which consumes only less than 1% of the reconfiguration energy in the ROAL protocol, with a greatly reduced packet latency. The MRCC protocol, considering the mobility, achieves better coverage by 1.4% with 22% fewer active sensors than that of an existing coverage protocol for the mobility. The results on the MCDS protocol show that the energy depletion ratio of nodes is decreased consequently, while the network throughput is improved by 35%.

To my soul mate Eunju and my family for their love and encouragement.

# ACKNOWLEDGMENTS

Dr. Eun Jung Kim, my academic advisor, taught me the lesson I needed most to learn. I am grateful to her stimulating suggestions and encouragement in all the times of research for and writing of this dissertation. I could learn her boundless energy dedicated to research and our lab students. The high standards she has imparted to me will guide my research efforts my whole life long. I would like to thank Dr. Valerie E. Taylor, my dissertation committee member, for helpful discussion on the way of data acquisition and presentation. I improved the results of this dissertation via her great idea. Dr. Rabi N. Mahapatra, my other committee member, has offered sound advice over the years. I have been incredibly fortunate to have had a creative discussion with him. I also wish to thank Dr. Deepa Kundur who is my other committee member. She has given and confirmed the novelty of this research and encouraged me to go ahead with my idea. I have furthermore to thank Dr. Yoon Suk Choi for his helpful advice and support for this dissertation during the presentation of this dissertation.

I am especially grateful for the friendship of Yuho Jin, Manhee Lee and Heungki Lee who are the first members of our lab. They always used to abide with me whenever I need their help. I also thank all our other lab members, Minseon, Inchoon, Sungho, Baik Song, Jay, and James for their helpful work and talk for my research. My former and current colleagues from the Department of Computer Science supported me in the way of technical discussion, advice, and friendship. I profited immensely from my colleagues of Korea Military Academy as well as the worship members from Vision-Mission Church during my study. They have not hesitated to share my pleasure as a brother and friend. I would like to appreciate all of them with my great honor.

TABLE OF CONTENTS

CHAPTER Page

# LIST OF TABLES

TABLE Page

# LIST OF FIGURES

FIGURE Page

# CHAPTER I

## INTRODUCTION

A wireless sensor network is a wireless network that is comprised of numerous small sensor nodes, each of which is equipped with a radio transceiver, a small microprocessor, a data memory, a group of sensors, and a battery. With respect to low cost and tiny sensor nodes, recent advances in embedded computing systems have developed a single tiny sensor node, called mote, within a size of 1-inch x 1.5-inch [1], or even a thumb-sized device [2].

Usually, a hundred or a thousand of the tiny sensor nodes are spatially distributed over a remote target area and operate autonomously. The distributed sensor nodes cooperatively collect and transmit sensed data, such as heat, pressure, light, vibrations, etc., and a new data regenerated from the raw data, through neighbor nodes. The task of data collection requires a reliable sensing ability, and the data transmission should be guaranteed between any pair of source and destination. Meanwhile, due to the energy constraints and unmanned control, the main challenge in the research of WSN is to prolong the lifetime of WSNs for the reliable data collection and transmission. Since the network lifetime of WSN is restricted by the small capacity of a battery, it is a critical issue to minimize the energy consumption of each node without loss of the functionality of WSN including a robust fault tolerating method.

In this dissertation, we are interested in the topic of topology management to provide energy efficient and robust WSNs by applying the well-known $K$-coverage and the load balancing problems. $K$-coverage is the study to fulfill the reliability of coverage while turning off any redundant node for the sake of energy conservation.

---

The journal model is *IEEE Transactions on Automatic Control.*

Load balancing, on the other hand, studies a technique to balance network traffic evenly so that an early energy depletion, along with any hot spot path, will not cause the network to disconnect. Most of the strategies concerning energy conservation fall into the above two categories.

### A. $K$-coverage Problem

The $K$-coverage of WSNs studies a methodology to ensure that every point in a target area is covered by at least $K$ different working nodes. The set of redundant nodes, then, can sleep until one of the working nodes fails. As a result, the $K$-covered network can extend the network lifetime without loss of sensing reliability. Hence, the trade-off between the quality of coverage and the number of nodes selected for maintaining the coverage is a critical issue to implement $K$-coverage algorithm.

A node can determine its eligibility as a working node by calculating the overlap of its covering area between its working neighbor nodes based on the geographic coordinate and the distance of sensing radius, which is a general approach in most $K$-coverage algorithms. This approach can provide almost perfect $K$-coverage; however, it will cause a significant computation and message overhead in a high density network. Furthermore, it can be costly to obtain an exact geographic coordinate using either the Global Positioning System (GPS) or a virtual coordinate retrieval algorithm [3, 4].

We propose Randomly Ordered Activation and Layering (ROAL) protocol for the $K$-coverage, which can select a set of working nodes locally without any coordinate system. Our decision algorithm in ROAL runs in $O(K)$ computation time and with $O(K)$ message overhead at each node, where $K$ is the degree of coverage. In our extended work of C-ROAL protocol, a more robust and energy-efficient fault-tolerant

$K$-coverage algorithm will be introduced.

In the point of nodal mobility, the ROAL and C-ROAL protocols express no concrete idea. For this reason, we study a $K$-coverage model under the mobile situation. A mobility resilient coverage control (MRCC) $K$-coverage model is designed to guarantee the $K$-coverage in the presence of mobility. In this work, we also present the impact of wear-out failures of sensor nodes to the obtained $K$-coverage.

## B. Load Balancing Problem

Load balancing, as explained before, refers to distributing network traffic evenly across a network so that no single path is overwhelmed and the network can remain connected as long as possible.

The problem of load balancing in WSNs has been studied through the multipath or distributed routing protocol. The multipath routing protocol aims to discover multiple disjoint paths between any pair of source and destination. The multiple paths discovered are maintained in a memory, i.e., route cache, and can be used either as a backup route for a broken path or to balance network traffic. However, the decision procedure to find an optimal path to balance network traffic using the multiple paths requires additional computational overhead, including managing the current link state such as a residual energy or the amount of usage [5].

On the other hand, a distributed routing scheme is developed for coordinate-based routing protocols, such as GPSR [6], to provide the load balancing. The geographic coordinate in GPSR is used for each forwarding node to select the next hop node closer to the direction of destination. Although GPSR can provide the distributed shortest path routing, it has the same problem related to using the GPS system and could cause a hot spot around an obstacle, i.e., a region unoccupied by

sensor nodes. Variant new virtual coordinate systems, instead of the geographic coordinate system, are proposed to avoid the hot spot problem in GPSR in many previous studies [3, 4, 7, 8].

In this dissertation, we suggest a MCDS protocol to balance the network traffic using dynamic source routing (DSR). The suggested protocol uses the concept of connected dominating set (CDS) to build a load-balanced topology unlike the coordinate-based protocols with which a big overhead to abstract a global topology could occur as the node density increases. In addition, our protocol needs no additional overhead to retrieve an optimal path among multiple paths. Instead, sensor nodes proactively construct multiple virtual layers that are composed of a connected dominating nodes, and the constructed multiple layers provide a distributed routing environment for DSR routing protocol.

In Chapter II, we propose the ROAL protocol that introduces a basic concept of layer coverage and a reconfiguration scheme as a method to balance the energy of each node. In Chapter III, we present a circulation method that can substitute each set of working nodes with a set of sleeping nodes instantly, which greatly reduces the energy consumed for the reconfiguration of the ROAL protocol. We also show the analytical model of the expected coverage and connectivity of our layer coverage in this chapter. In Chapter IV, we develop a $K$-coverage model in the presence of mobility. We derive a model of moving-in and moving-out probability to correctly model the dynamic changes of network with the respect of $K$-coverage. We also combine a wear-out failure of each node to properly control the working period of individual nodes. In Chapter V, we discuss the MCDS protocol. An heuristic algorithm to construct multiple CDSs is introduced, and the feature of load balancing and network performance with the MCDS protocol is studied against the case where only DSR is used without MCDS. Finally, we conclude our research in Chapter VI by reviewing

the suggested protocols with our contribution.

# CHAPTER II

## ROAL: A RANDOMLY ORDERED ACTIVATION AND LAYERING PROTOCOL FOR $K$-COVERAGE

In this chapter, we propose a Randomly Ordered Activation and Layering (ROAL) protocol. Each node under the ROAL protocol can decide its eligibility regarding a given coverage degree $K$ at randomly generated activation time using only the coverage status informed from its neighbor nodes located within its sensing region. A new concept of layer coverage also provides a simple and effective reconfiguration method for energy balancing. The simulation results show that the ROAL protocol can guarantee $K$-coverage with more than 95% coverage ratio, which almost closes to the coverage ratio that is achieved using the geographic coordinate. A significantly extended network lifetime is also observed against the original topology of a given network.

The rest of the chapter is organized as follows. In Section A, we discuss the related work and we provide the details of the ROAL protocol in Section B. In Section C, we present our simulation results to analyze the performance of the proposed algorithm, while we conclude our work in Section D.

### A. Background and Related Work

The $K$-coverage problem is the study on the decision for selecting a set of working nodes such that, with $K$-covered sensor network, any point in an interesting area is monitored by at least $K$ different sensor nodes. The final goal of the $K$-coverage is, hence, to prolong the network lifetime using a limited energy budget of sensor nodes without losing the sensing quality.

Previous $K$-coverage algorithms can be classified into two different categories,

deterministic and probabilistic algorithms. The algorithms in the first category [9, 10, 11] aim to monitor as many points as possible with $K$ different active nodes simultaneously. The other approaches [12, 13, 14, 15] provide the required $K$-coverage based on the expected number of observations for each point, moving target during a given time interval, or the whole duration of movements.

As a highly related approach, we show three studies here. Ye et al. [10] proposed the probing environment and adaptive sleeping (PEAS) protocol that can cover and connect a sensor network by activating only one node within a probing radius of a node. They provided an heuristic way to provide a certain degree of coverage, decided by the number of distributed sensors. An integrated analytical model for multi-coverage and connectivity was suggested by Xing et al. [11], where a sensor network is $K$-covered if and only if all the points within the intersection area formed by all neighboring nodes are covered by $K$ nodes. The main problem with this approach is the time complexity of $O(N^3)$, where $N$ is the number of neighboring nodes. On the other hand, Set $K$-Cover problem [15] uses a similar concept as our layering algorithm. However, in these studies, the focus is to make $K$ subsets using all deployed nodes such that each subset covers all area or can take a $K$-coverage effect by the iterative activation of each subset in a round-robin fashion. In this scheme, each node belongs to one subset, and then each subset is activated one by one. To select nodes efficiently in terms of accuracy, they also use the geographic information. The ROAL protocol suggested here selects only $K$ subsets and the purpose is to guarantee 1-coverage for each layer without using the geographic information.

We apply the probing scheme proposed in PEAS [10] and sentry selection protocol [16], where each node sends a hello message to check out any other active nodes within its sensing area. If there is a reply from an active node, the probing node will sleep until the next probing time arrives. We enhance this approach to validate if any

$K$ different nodes are working within the sensing range using one message per each node. The running time of our algorithm is bounded by a small constant value of $t$, a given interval of a phase. The main idea is to make $K$ layers such that each layer is composed of a set of nodes to provide 1-coverage and the $K$ different layers provide $K$-coverage together.

## B. The ROAL Protocol

### 1. Basic Idea

The basic idea is to build $K$ logical layers[1] for requested $K$-coverage, where each layer consists of a disjoint set of working nodes that provide 1-coverage for the whole target sensing region, as shown in Fig. 1. In addition, we assume that the nodes remain in their original position in this work. From the set $S$ of all sensor nodes,

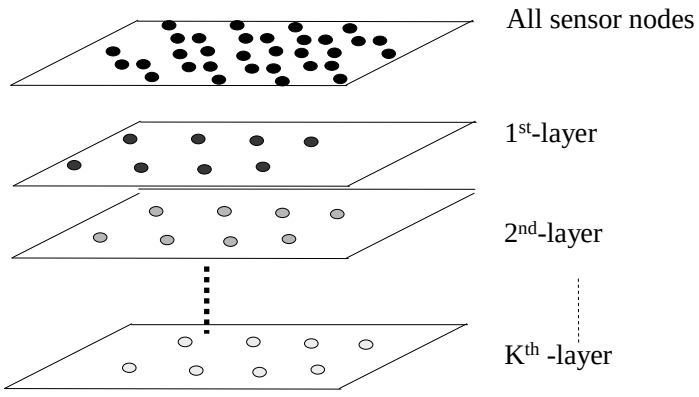



Fig. 1. $K$-Layer Coverage

we select only a small number of nodes to form 1-coverage and repeat this process $K$ times to form $K$-coverage. A set $S_i$, which is $i^{th}$ subset (or layer) of $S$, is composed

[1]A layer represents a virtual plane that includes a subset of working nodes.

of selected nodes, and $S_i \cap S_j = \phi$, if $i \neq j$ and $1 \leq i, j \leq K$. Also, $\cup_{i=1}^{K} S_i \subseteq S$ and $\sum_{i=1}^{K} |S_i| \leq |S|$.

All these selected nodes remain working to provide $K$-coverage for a predetermined period, while the other nodes go to sleep to save their residual energy. After the period, this process can be repeated to evenly distribute the energy consumption among the sensor nodes in the WSN.

Using this idea, we can easily change the degree of coverage during the network running time if a user wants to increase the degree of coverage for more accurate data or to reduce the degree for energy conservation. Unlike all the previous studies that did not consider dynamic real-time reconfiguration on the degree of coverage seriously, our approach can easily cope with such demands.

## 2. Randomly Ordered Activation (ROA) Algorithm

Randomly Ordered Activation is a stochastic and greedy algorithm that selects $K$ sets of working nodes for $K$-coverage at a randomly generated activation time. Before its activation time expires, a node running the ROA algorithm maintains a list of layer numbers (LIDs) sent by its neighbor nodes within its sensing circle area. The eligibility as a working node is decided when the activation time expires.

**Pseudo Code of ROA Algorithm**

Algorithm ROA($K$, $T_A$)

1. $t \leftarrow 0$
2. $LID \leftarrow 0$
3. $H \leftarrow \emptyset$

4. $T_a \leftarrow rand(0, T_A)$
5. **while** $t < T_A$
6. **if** ACTIVE message arrives from neighbor node
7. $H(ACTIVE.LID) \leftarrow true$
8. **if** $t = T_a$
9. $i \leftarrow 1$
10. **while** $i \leq K$
11. **if** $H(i) = false$
12. $LID \leftarrow i$
13. send ACTIVE.LID
14. i++
15. **if** $LID = 0$
16. sleep
17. **else**
18. set active

A field of boolean array $H$ indexed by the LID that is carried in the ACTIVE message of a neighbor node is set to true. A node will work if it finds its LID less than $K$ or it will go to a sleep mode otherwise. The ROA algorithm, therefore, can run in $O(K)$ time with $O(K)$ number of message exchanges at each node.

## 3. Detailed ROAL Protocol

In this section, we complete the design of the ROAL protocol that can maintain the $K$-coverage in a round-robin fashion for the purpose of energy balancing among all distributed nodes. Each round consists of three phases: Initialization Phase (IP), Activation Phase (AP), and Working Phase (WP). The duration of each phase is determined by the condition of the network such as the density of sensor nodes or the tasks of the applications. For simplicity, let three parameters, $T_I$, $T_A$, and $T_W$, be the durations of the IP, the AP, and the WP, respectively. In addition, let $T_a$ and $T_n$ be randomly generated activation and notification times, respectively, and they are used to avoid collisions in the wireless channel. Note that $0 < T_n < T_I$ since $T_n$ is used during the IP, and $0 < T_a < T_A$ since $T_a$ is used during the AP.

### a. Initialization Phase (IP)

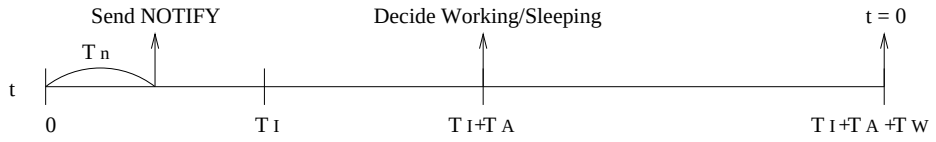
Each round starts with setting the local timer to 0, and then the IP begins. At the beginning of the IP, all sleeping sensor nodes wake up and participate in the decision (for working or sleeping) process with the working nodes in the previous round. Let $S_W$ and $S_S$ be the sets of working nodes and sleeping nodes in the previous round, respectively. Also, $R_n$ indicates the $n^{th}$ round. Then there are two cases depending on the round number.

**Case 1**: The first round ($R_1$)

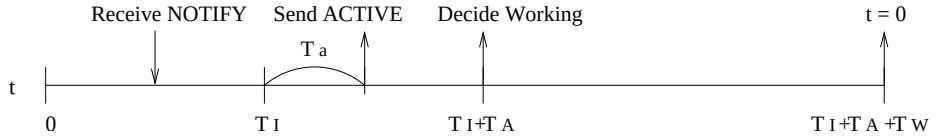
When sensor nodes are initially deployed over an area, all $K$ layers should be constructed. In this case, all nodes generate the activation time $T_a$ and wait for the starting of the AP.

**Case 2**: The second round or later ($R_n, n \geq 2$)

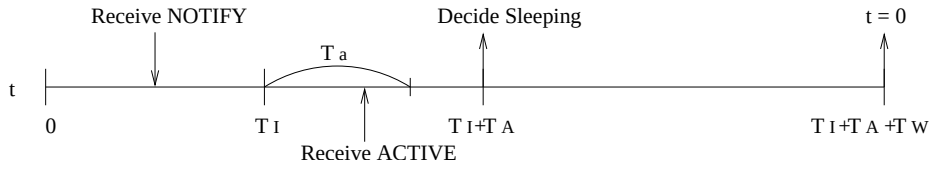
When the second round starts, we have a set of working nodes and a set of




(a) A Node in $S_W$ Decides Working/Sleeping.




(b) A Node in $S_S$ Sends ACTIVE and Becomes Working.




(c) A Node in $S_S$ Receives ACTIVE and Becomes Sleeping.

Fig. 2. Three Possible Scenarios in the ROAL Protocol

sleeping nodes. Each working node that belongs to $S_W$ has to increase its LID and all sleeping nodes wake up. Depending on the new $K$ and the previous $K$ values, there are three cases for this increase.

- Option 1: If there is no request to change the degree of coverage, the LID of each working node is increased by one. After increasing its LID, each working node will decide the next state for itself by comparing its LID with $K$. If the increased LID is greater than $K$, the working node will sleep for the next round.

- Option 2: If there is a request for a new increased $K$, each working node needs to increase its LID by the difference between the new increased $K$ and the previous $K$. This process will make more layers than one.

- Option 3: If there is a request for a decreased $K$, each working node increases its LID by one, like in Option 1, and if its LID is greater than the new $K$, the node goes to the sleep state for the next round.

After the increment of LID, each working node generates $T_n$ to decide the time when it sends a NOTIFY message to its neighbors. When $T_n$ expires, it broadcasts a NOTIFY message containing its LID and the new $K$, as shown in Fig. 2 (a). By receiving the message, newly awakened nodes can determine which layers have already been formed by currently working nodes and how many new layers should be built by themselves. In addition, each awakened node generates its Random Activation Time ($T_a$).

b. Activation Phase (AP)

All newly awakened nodes try to be working during the AP by sending out ACTIVE messages to their neighbors. While waiting for the Random Activation Time ($T_a$), each awakened node maintains a list of layers already composed by its neighboring

nodes using the LIDs, which are included in the NOTIFY messages from working nodes in the previous round or in the ACTIVE messages from other awakened nodes. When its $T_a$ expires, the node checks the list of layers that are already constructed. If it finds out a layer that is not made yet, the node sets its LID as the layer number and sends out its ACTIVE message with the LID as shown in Fig. 2 (b). After a node broadcasts its ACTIVE message, it will work as a working node during the WP. A node will go to sleep during the WP if all layers are already constructed before its $T_a$ expires, as shown in Fig. 2 (c).

The decision on the coverage is made by the reception of an ACTIVE message within the distance of sensing radius $r_s$ at each node. Through this approach, we can obtain a good approximation on 1-coverage for each layer. A more accurate analytical model will be studied further as a future work.

c. Working Phase (WP)

A node with its LID between 1 and $K$ works as a working node during the WP. All the other nodes go to sleep during the WP in the current round. During the WP, if a request for a new $K$ that is less than the current $K$ is received, each working node compares the new $K$ with its LID and goes to the sleeping state instantly if its LID is greater than the new $K$, as shown in Fig. 3, which is the state transition diagram for each node during one round. This is another benefit of our protocol for dynamic reconfiguration. If the new $K$ is larger than the current $K$, reconfiguration occurs in the next round. In Fig. 3, a transition occurs when the local timer ($t$) of a node indicates the start of the next phase and/or a certain condition is met.

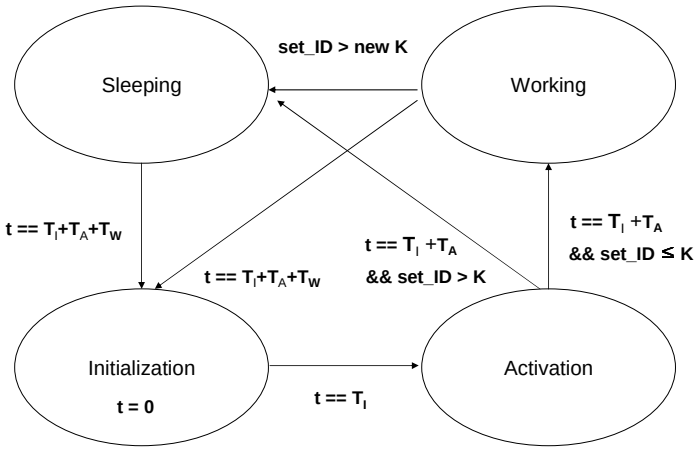


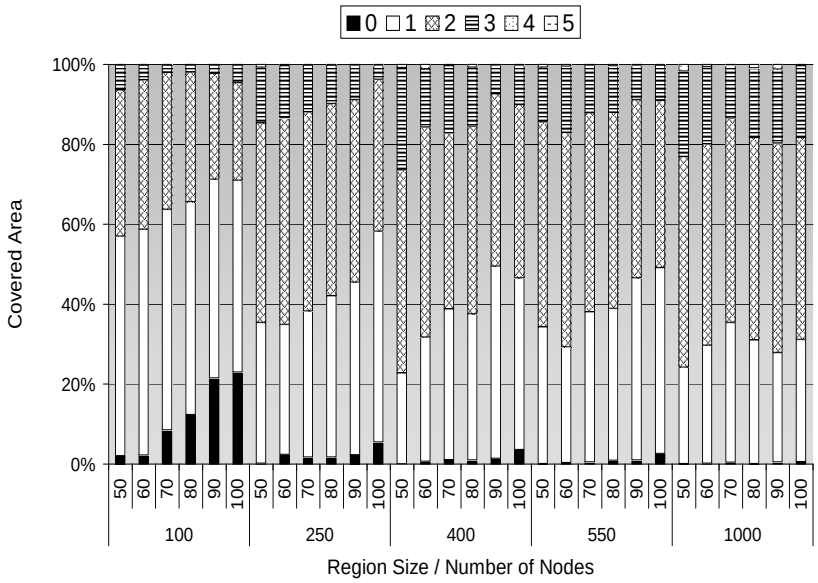

Fig. 3. State Transition Diagram for Each Node During One Round

## C. Experimental Results

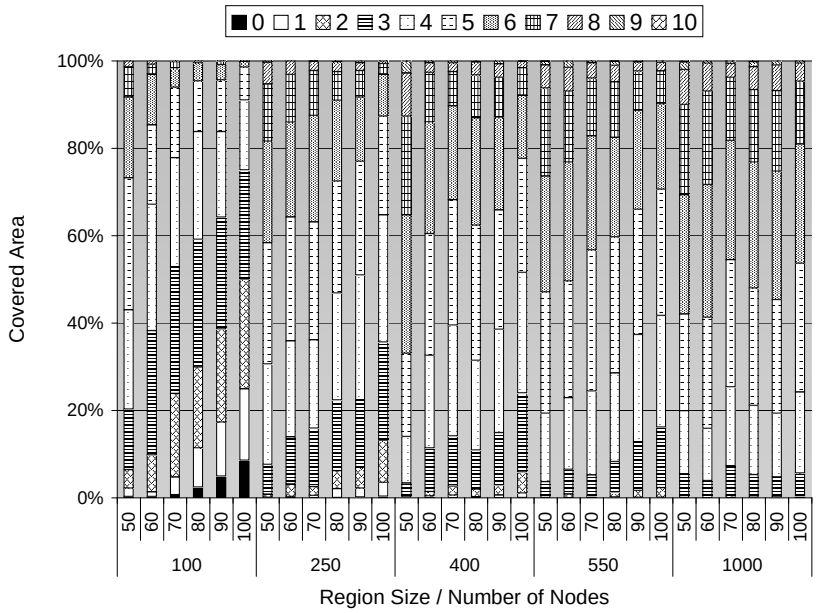
In this section, we show the results on coverage and network performance with the ROAL protocol.

### 1. Coverage Evaluation

To measure the coverage, the entire sensing region is divided into $1m \times 1m$ grids. Each point is considered to be covered if the point is located within the sensing range of a working node. The sensing range is $10m$, while the communication range is $30m$. Fig. 4 (a) and (b) show that the percentages of the covered areas for 1- and 3-coverage networks with the ROAL protocol, respectively. Each bar represents the ratios of resulting coverages for a specific region size/number of nodes. The upper row of the X-axis indicates the size of the region where sensor nodes are deployed. For example, 50 implies a $50m \times 50m$ region. The lower row of the X-axis indicates the number of sensor nodes deployed in the region. The ratio of the uncovered area with 1-coverage in Fig. 4 (a) reaches up to 24% when the density is 0.01 (100 nodes/10,000$m^2$), which is the worst case. If the density exceeds 0.025 (250 nodes/10,000$m^2$), the ratio of




(a) ROAL Protocol with 1-coverage

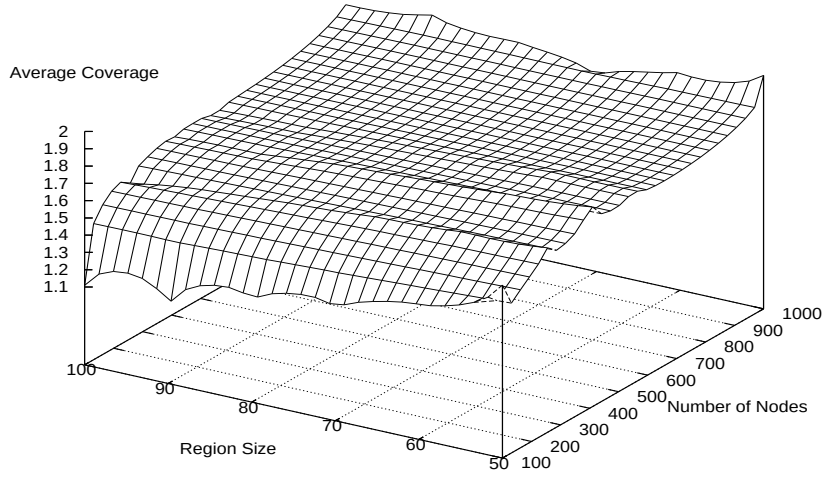



(b) ROAL Protocol with 3-coverage

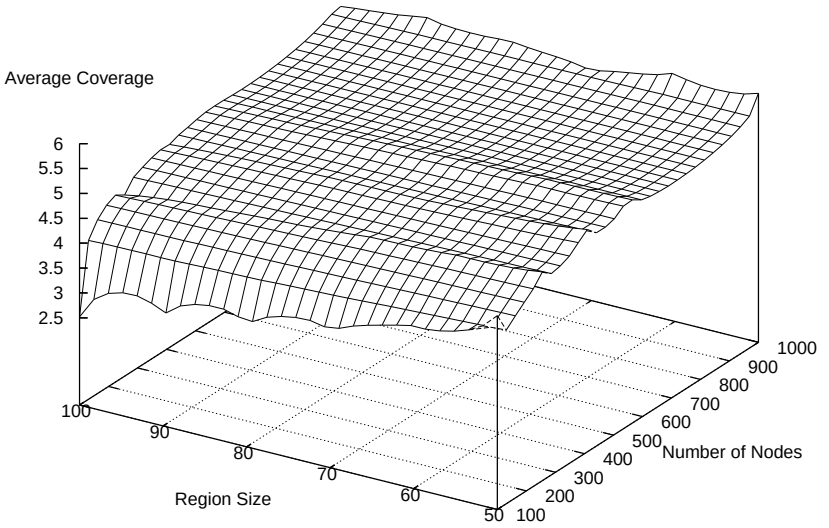
Fig. 4. Ratios of Covered Areas

the uncovered area decreases to below 5%. The ratio of the uncovered area with 3-coverage (0-, 1-, and 2-coverages) in Fig. 4 (b) reaches up to 50% when the density is 0.01. However, as the density increases, this ratio also becomes small. According to our observation, there still exists around 8% (for 1-coverage) and 23% (for 3-coverage) uncovered area with all sensor nodes working with the same number of nodes and the network size. Hence, the uncovered area incurred by the ROAL protocol is very small, less than 2% of the total region.

Fig. 5 (a) and (b) show the average degrees of 1- and 3-coverage networks with the ROAL protocol. The average degrees of 1-coverage range from 1.1 to 2 with different densities. This implies that the ROAL protocol can efficiently manage the quality of the required degree of coverage using a reasonable number of working nodes. The average degrees of the 3-coverage network also range from 2.5 to 6. Fig. 6 shows the number of working nodes with the region size of $50m \times 50m$ for 1-coverage and 3-coverage, respectively. The actual number of working nodes grows very slowly, while the number of the sensor nodes increases steeply. Compared to the results obtained using the geographic information in the CCP [11], the ROAL protocol can provide very competitive results without using any geographic information. The results on the average degree for 1-coverage and the number of working nodes for 1 and 3-coverage are close to each other. Moreover, since the ROAL protocol requires much lower running overhead compared to the approaches that use the geographic information, it really improves the energy performance of the sensor network. In addition, our protocol can support the desired degree of coverage, which is not provided in PEAS protocol.

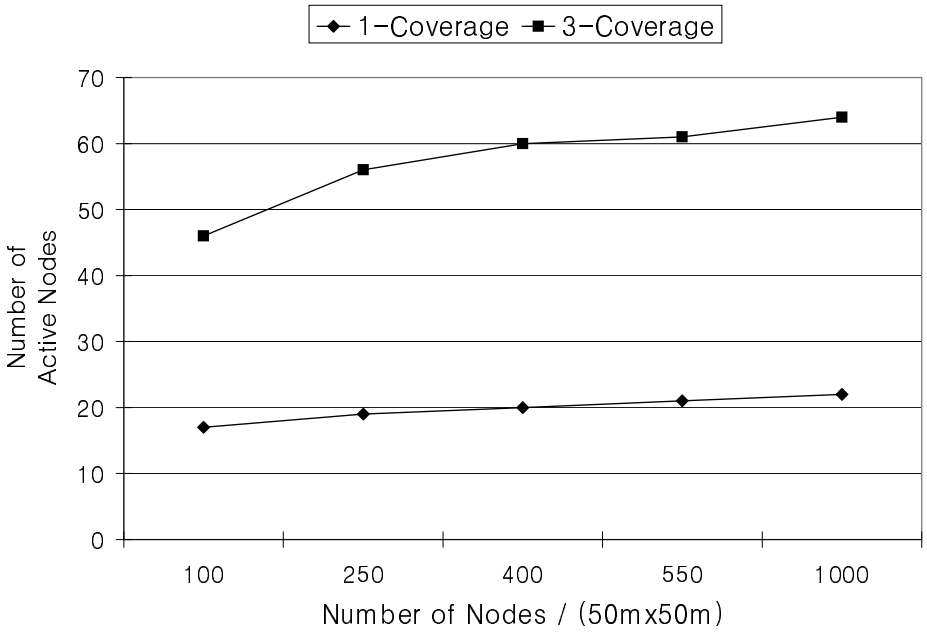



(a) Average Coverage Degrees with 1-coverage




(b) Average Coverage Degrees with 3-coverage

Fig. 5. Average Coverage Degree




(a) Number of Working Nodes

Fig. 6. Number of Working Nodes

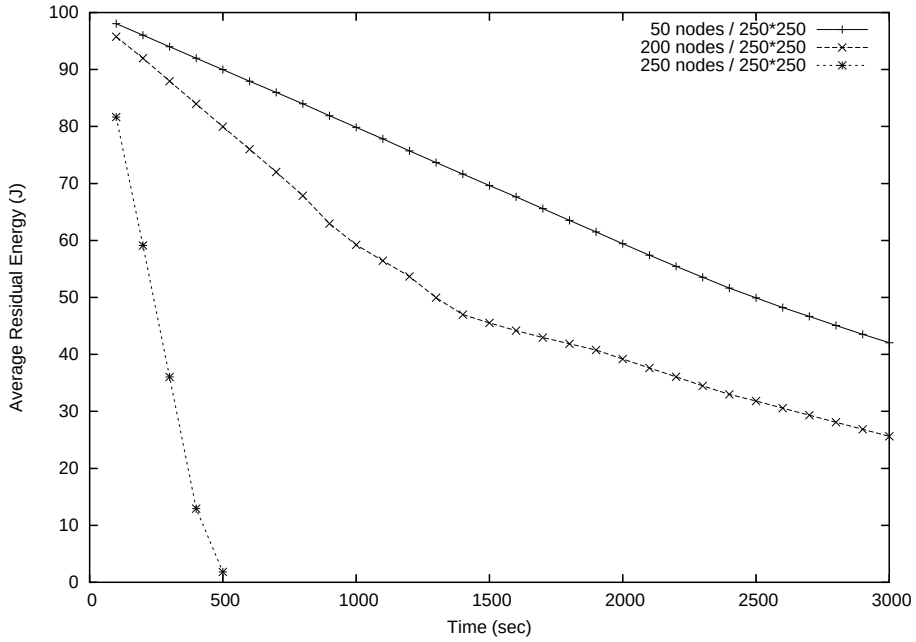



Fig. 7. Average Residual Energy with DSR Only

### 2. Network Performance

In this section, we evaluate the coverage lifetime and the packet delivery ratio, along with the residual energy of the network using the ns-2 simulator. We use the DSR routing protocol [17] to evaluate the ROAL protocol because it provides an on-demand source routing that does not need any location information and it is the basic routing scheme for other on-demand routing protocols.

For this simulation, $30m$ is set for the sensing radius and $75m$ for the communication radius of each node. We use $250m \times 250m$ 2-dimensional square for a target sensing region. In addition, there are 10 event points distributed randomly around the upper bound of the sensing area and each point generates 5 events per second. When working nodes around the event points sense the generated events, they send a 512 byte packet per one event to the sink node that is located at the right bottom of the sensing region. The average coverage is measured by counting the number of

neighboring nodes that detect the event. All results shown in this section are obtained using 1,000 second round time (5 seconds for $T_I$ and $T_A$ each, and 990 seconds for $T_W$), and simulation data are collected every 100 seconds. Also, each sensor node is given 100 Jules of initial energy.

Fig. 7 shows the average residual energy with the DSR protocol only (i.e., all nodes are working) for 50 nodes, 200 nodes, and 250 nodes, respectively. It is clear that, without any energy-saving scheme, the network with a small number of nodes has more residual energy than the one with a larger number of nodes. This implies that excessively redundant nodes cause more energy consumption with the DSR routing protocol that uses a broadcasting scheme. Fig. 8 (a) and (b) show the average residual energy and the minimum residual energy of the network for different coverage degrees ($K = 1$ and 2). With the ROAL protocol, the network can reserve more energy than with only the DSR protocol.

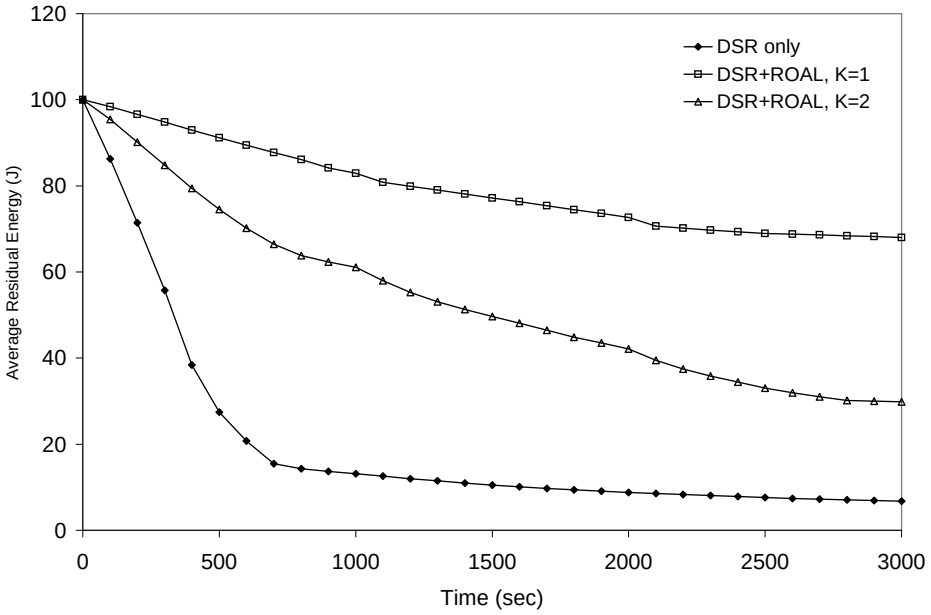
The average packet delivery ratio is shown in Fig. 9 (a). More than 95% of packets are dropped after 800 seconds with the DSR only. With the ROAL protocol, almost 100% packets are delivered up to 2,900 seconds when $K$ is 2, and up to 3,000 seconds when $K$ is 1. When $K$ is 2, the delivery ratio drops to 0 after 2,900 seconds because some intermediate nodes between the sources and the sink node completely depletes their energy. Some temporal drops are caused by packet losses during the reconfiguration period. In Fig. 9 (b), the average degree of coverage is shown with 380 sensor nodes. The average degree of 2-coverage remains around 2.0, while 1-coverage shows the average degree over 1 during the whole simulation time. Without the ROAL protocol, the average degree of coverage is around 5 at the beginning of the simulation, but it rapidly drops to around 1 after 300 seconds, since sensor nodes around the event points have died together for energy depletion except about one working node. Therefore, the network with the ROAL protocol can capture the

events for a longer time since it uses a small number of different working nodes in each round. In addition, the results also prove that the ROAL protocol can provide the required degree of coverage efficiently.

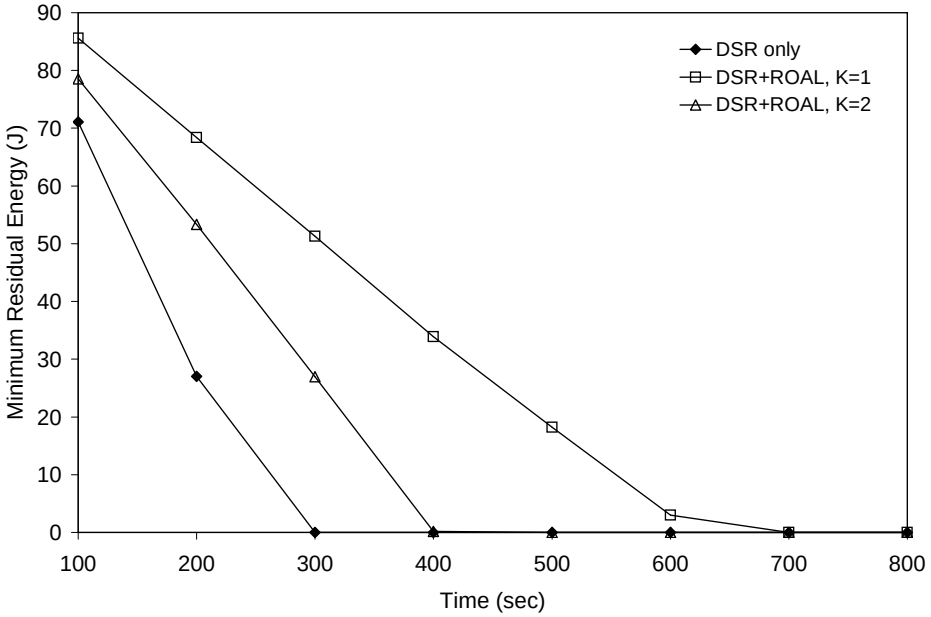
## D. Conclusion

In this chapter, we proposed a fast and efficient $K$-coverage algorithm, called the ROAL protocol, to solve the problem of providing a certain degree of coverage in WSNs. The main idea of the ROAL protocol is to ensure $K$-coverage using $K$ subsets of working nodes using the layering concept, where each subset guarantees 1-coverage. The ROAL protocol efficiently constructs $K$-coverage network with low message overheads and guaranteed packet delivery with the advantages of energy-savings in the network. Simulation results also support our claim.

In future work, we will suggest more useful schemes to select the working node sets regarding energy burdens in each node and may study on the measurement scheme for the duration of each phase regarding both maximal and a given desired network lifetime. Also, a proper reaction for the event of faulty nodes is an important issue for further study.

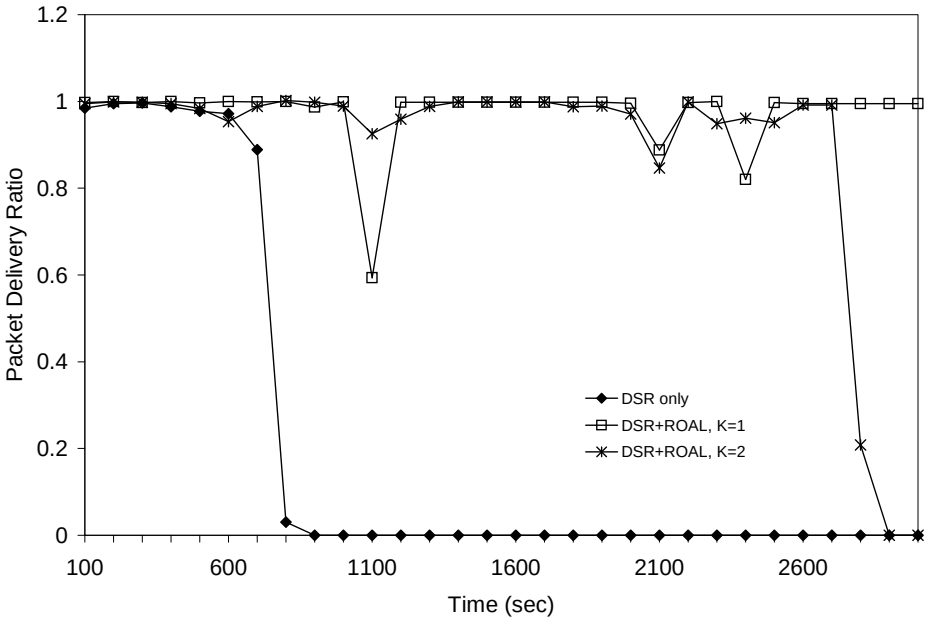



(a) Average Residual Energy

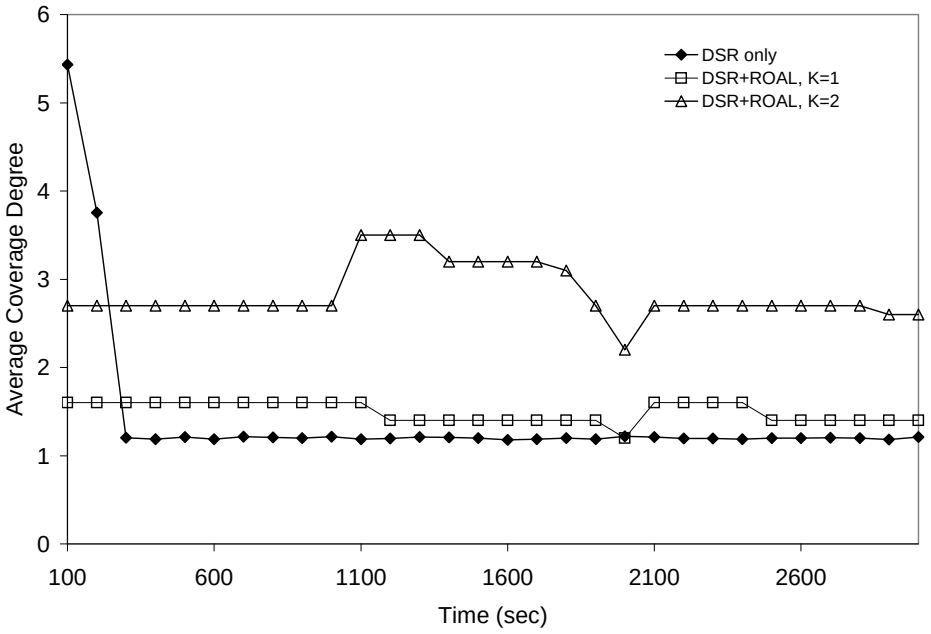



(b) Minimum Residual Energy

Fig. 8. Residual Energy with the ROAL Protocol




(a) Packet Delivery Ratio




(b) Average Coverage Degree

Fig. 9. Packet Delivery Ratio and Average Coverage Degree

# CHAPTER III

## C-ROAL: CIRCULATION-ROAL ENHANCED WITH FAULT-TOLERANCE FOR $K$-COVERAGE

We previously proposed a randomly ordered activation and layering (ROAL) protocol [18] for $K$-coverage. Although the $K$-coverage almost surely can be achieved by the ROAL protocol in a small constant time without the help of the GPS or a virtual coordinate algorithm, it requires a periodic reconfiguration for a fault tolerance and energy balance. In this chapter, we propose a C-ROAL protocol using a new circulation scheme to reconfigure the set of working nodes in an autonomous way, where the reconfiguration can be performed with a small and almost constant energy consumption. We also provide the model of the expected coverage and connectivity for the layer coverage and show a proper range in which only one node can be activated with regard to a node density and the sensing radius of a node.

The experimental results on the energy consumption show that the fraction of total reconfiguration energy becomes less than 1% of the energy consumed in the ROAL protocol with a high frequency of reconfiguration. Also, we obtain a greatly reduced packet latency, which corresponds to only 5% of the delay that occurred in the ROAL protocol.

The rest of this chapter is organized as follows. The problem related to the fault tolerance of previous studies is discussed in Section A. In Section B, we provide the details of a new C-ROAL protocol. We discuss the analytical model of coverage and connectivity, and show our experimental results on the performance of circulation in Section C. We conclude our work in Section D.

## A. Background and Related Work

In the PEAS protocol [10], where a probing scheme selects one node per unit probing area, the probing rate of each sleeping node is adjusted to the desired probing rate specified by the application to keep the rate under the existence of any faulty nodes. Since the frequency of wake-up will increase as the probing rate decreases, the energy overhead for the fault-tolerance becomes high according to the fault rate and the number of deployed nodes. Another problem in PEAS may occur in unbalanced energy burdens. Each node is only deactivated when it depletes all residual energy, which in turn can cause a locally uncovered region.

The CCP protocol is a deterministic approach and has a high time complexity of $O(N^3)$, where $N$ is the number of neighboring nodes. In CCP, each working node sends a periodic beacon message, while each sleeping node wakes up periodically to hear the beacon messages. The expected overhead of energy for fault-tolerance will also increase as the number of deployed nodes increases.

Set $K$-Cover problem[1] [19, 15], which is a probabilistic approach, makes $K$ subsets using all deployed nodes such that the frequency of covering any point in the interesting area is maximized as many as at least $K$ times, while each subset is activated in a round-robin fashion. This approach introduced a round-robin coverage that is able to balance the energy consumption with a fault-tolerance. Our circulation method suggested in this study uses the same concept with the round-robin approach. However, our protocol can guarantee $K$-coverage while circulating one layer at each round.

In our previous study [18], we suggested a randomly ordered activation and layer-

[1]Circulation suggested in this study uses the same round-robin approach. However, our scheme builds as many layers as possible and activates only $K$ layers at a time.

ing (ROAL) protocol, which provides the $K$-coverage using a new layering scheme. We proved that the layer coverage makes it simple to build and maintain the $K$-coverage. However, the accumulative energy consumption spent for a periodic reconfiguration will increase in proportion to the fault rate or the number of deployed nodes.

In this work, we propose a new protocol that can improve the performance of fault-tolerance and energy-balance for $K$-coverage based on our layer coverage. With our proposed circulation method, we recursively activate a different set of nodes for the fault-tolerance and the energy-balance among all deployed nodes without the repetitive layering procedure of the ROAL protocol. Also, we present the analytical model of the coverage and the connectivity of our layer coverage to verify the effect of the unit size of layering radius to the resultant coverage and connectivity.

## B. Circulation-ROAL Protocol

In Chapter II, we introduced a primitive reconfiguration scheme where all nodes wake up and generate a random and uniformly distributed activation time $T_a$. When its $T_a$ expires, a node looks up any available layer that has not yet been selected by its neighbors, and will be activated by sending an ACTIVE message if one of $K$ layers is missing. Since the ROAL protocol repeats the layering procedure at every working period, the potential cost of energy will be increased as the frequency of layering is raised for an increased fault ratio. The reason is because the ROA algorithm needs to build $K$ layers at every round. We resolve this problem by modifying the layering procedure such that all possible layers are built at the first period and we circulate them at every period, removing the repetitive layering procedure.

## 1. Model of Circulation-ROAL (C-ROAL)

In this section, we explain the details on the C-ROAL protocol that circulates the layers without repeating the layering procedure at every working round $T_w$ as the ROAL protocol needs. First, we define by $L(l)$ a set of nodes that have a layer_ID (LID) $l$.

**A**lgorithm C-ROA ($K$, $T_A$)

1. $t \leftarrow 0$
2. $LID \leftarrow 0$
3. $H \leftarrow \emptyset$
4. $T_a \leftarrow rand(0, T_A)$
5. **while** $t < T_A$
6. **if** $t = T_a$
7. $LID \leftarrow minl|H(l) = false$
8. send ACTIVE.LID
9. **else if** ACTIVE.LID arrives
10. $H(LID) \leftarrow true$
11. **if** $h \leftarrow max[H]$ is less than or equal to $K$
12. $T_w = \infty$
13. **else if** my $l$ is less than or equal to $K$

14. calculate $T_w$ and go to work

15. **else**

16. calculate $T_s$ and go to sleep

Every node obtains its LID $l$ when its $T_a$ expires. In the original ROAL protocol, only $K$ layers are formed by the ROA algorithm. However, the C-ROA algorithm assigns layer id to all nodes. Then, each node will act following the circulation rule either setting a sleeping period ($T_s$) or a working period ($T_w$) based on its LID. The calculation of $T_w$ and $T_s$ is explained in the next section. Based on its LID, every node calculates its working or sleeping period.

### 2. Circulation Rule

A circulation rule is developed for the purpose of a low-power reconfiguration of working nodes while providing a robust and energy-balanced $K$-coverage. Usually, a sleeping node needs to wake up with a certain frequency to monitor if some healthy nodes are working within its sensing range for assuring a reliable quality of $K$-coverage. Hence, the frequency of the fault detection messages will be increased as a fault ratio increases, which deteriorates the energy credit in WSNs. A different way is implemented in this study using the circulation scheme that substitutes a set of working nodes with a different set of sleeping nodes, meanwhile guaranteeing a constant $K$-coverage. This property differentiates our approach from the Set $K$-cover study in that the degree of coverage can be changed whenever a different set is activated [19, 15].

Fig. 10 shows the state transition diagram of circulation. Using a LID obtained by the C-ROA algorithm, each node calculates $T_w$ or $T_s$. Whenever $T_w$ of working nodes or $T_s$ of sleeping nodes expires, each node transits its state, as shown in Fig. 10.

Hence, the transition between the working and sleeping state happens autonomously, which can reduce the energy consumed for a repetitive layering at every round as in the ROAL protocol.

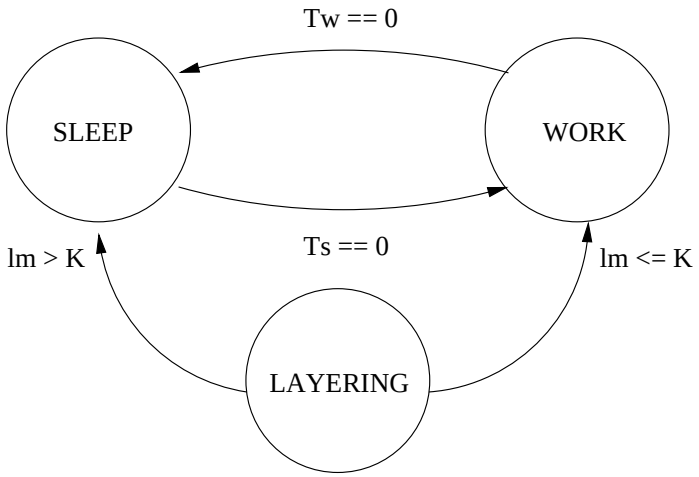



Fig. 10. State Transition with Circulation

We implement a deterministic circulation, which uses a fixed and constant reconfiguration time $T_r$. Each node calculates its $T_s$ or $T_w$ using the predefined $T_r$. Each node knows the highest LID $h$ by hearing the packets exchanged during the layering procedure. With a given degree of coverage $K$, its LID $l$, the highest layer id $h$ and $T_r$, a node determines both $T_s$ and $T_w$ as follows.

A. Initial calculation for $T_w$ and $T_s$

Initially, we need to activate all $K$ layers for the first round. Each node that has a LID $\leq K$ decides to work and calculates its $T_w$. $T_w$ will be different according to its LID $l$ since only nodes in the bottom layer, i.e., layer 1, will sleep after this first round. For this reason, nodes that have LID $l$ less than or equal to $K$ will work during $T_w$ that is calculated by:

$$T_w(l) = \begin{cases} T_r \times l, & \text{if } l \leq K \text{ and } h > K \\ \infty, & \text{if } l \leq K \text{ and } h \leq K. \end{cases}$$

If the highest LID $h$ of a node $v$ is smaller than $K$, $v$ will set its $T_w = \infty$. If its LID is greater than $K$, $v$ will go to sleep during $T_s$ calculated as follows:

$$T_s(l) = (l - K) \times T_r, \text{ if } l \geq K.$$

We circulate one layer in a sequence of $C(1)$, $C(2)$, $C(3)$, $\ldots, C(h)$ at every $T_r$, while sustaining $K$ layers at every time instance. The term of circulation $C(l)$ defines that working nodes in layer $l$ go to sleep and sleeping nodes in layer $m$ wake up to work, where $m = (K + l) \bmod h$, if $m > 0$, or $m = h$ if not.

B. After initial calculation

Once the first period of working or sleeping mode finishes, the way of calculation changes as follows.

$$T_w(l) = T_r \times K$$
$$T_s(l) = (h - K) \times T_r.$$

Note that a current working node will have a new $T_s$ and a newly wake-up node will calculate a new $T_w$. Because each node can decide its $T_w$ and $T_s$ using $h$ and its LID with the given values of $T_r$ and $K$, a local difference of density will not affect the the overall coverage.

### 3. Discussion on Fault Tolerance

Using the circulation scheme, we can provide an energy efficient fault tolerance while maintaining $K$-coverage. First of all, the $K$-coverage itself is a scheme to provide a high probability of detection unless $K$ nodes within the unit sensing area have fault. In addition, the circulation scheme can provide a more robust environment by circulating a faulty layer with a new healthy layer. If we can decrease the interval of circulation, the recovery time for the faulty region can be minimized. We concern the cost of reconfiguration in terms of both the energy efficiency and the network performance. As shown in the experiment of an energy and delay in Section 3, the circulation proves itself as an energy-efficient fault-tolerant scheme for $K$-coverage in that the energy consumed for the reconfiguration remains almost constant and small even if the frequency of the reconfiguration increases dramatically. As well as the fault-tolerance, the circulation scheme is a good method for the energy-balance because a set of nodes belonging to a layer will be activated in a round-robin fashion. For the summary, we can say that the C-ROAL protocol provides a robust and energy-efficient fault-tolerance with energy-balance.

## C. Analysis on Layer Coverage

In this section, we analyze the probability of the coverage and connectivity of our layer coverage. The coverage and connectivity depends on the layering radius of $r_l$, the sensing radius of $r_s$, the communication range $r_c$, and a node density. We assume that the positions of nodes follow the Poisson point process of constant and finite density $\lambda$ in area $\mathbb{R}^2$. Here, we assume that there will be no collision during the layering procedure for the simplicity of proof.

A random variable $i$ that is the expected number of nodes in a circle area of

$\pi r^2$ centered at a random point follows the Poisson distribution such as $P(i,r) = e^{-\lambda\pi r^2}(\lambda\pi r^2)^i/i!$. At a random position, each node decides its layer by running ROA or C-ROA algorithm. Assume that the layering radius is $r_l$ where a node sends an ACTIVE message to its neighbors. As shown in Fig. 11, the minimum distance between any nodes in the same layer will be $r_l/2$, and there is only one node within the area of $\pi(r_l/2)^2$ as depicted with the smallest ball in Fig. 11.

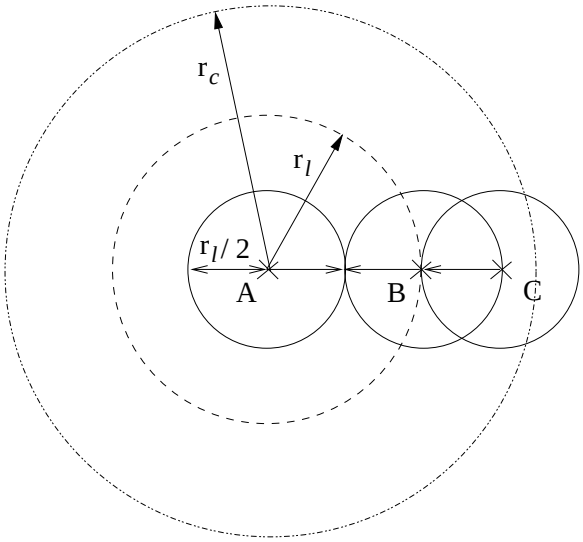



Fig. 11. Minimum Distance between Two Working Nodes

### 1. Connectivity

The connectivity will be guaranteed in our layer coverage if any on-duty node working for a certain layer is connected. We mean by connected that any two nodes can reach to each other in multiple hops. We follow a similar procedure, as shown in the study [20] to prove the connectivity, but provide a different model using the Poisson point process model.

**Lemma 1**. Assume that $n$ nodes are distributed in $\mathbb{R} = [0,l]^2$ according to the

Poisson point process model, Then, there is almost surely at least 1 node per unit area of $d^2$ when $l$ goes to infinity if the density of nodes satisfies $\lambda = \frac{k \ln l^2}{d^2}$ for $k > 1$.

*Proof.* We divide the region $\mathbb{R}$ into $N = \frac{l^2}{d^2}$ squares of size $d \times d$. Let $\mu_0(n, N)$ be the random variable denoting the number of empty squares of size $d \times d$, where $n$ is the number of nodes in area $\mathbb{R}$, and $p_0$ is the probability of empty nodes in one square. Then, the expected number of empty cell $E[\mu_0(n, N)]$ will be:

$$E[\mu_0(n, N)] = N \cdot p_0 = N \cdot e^{-\lambda d^2} = N e^{-nN^{-1}}.$$

Here, we want to find $E[\mu_0(n, N)]$ when $l \to \infty$. From the above equation, we obtain:

$$\ln E[\mu_0(n, N)] = \ln \tfrac{l^2}{d^2} - \tfrac{nd^2}{l^2}.$$

If we assume $nd^2 = kl^2 \ln l^2$, the above equation becomes:

$$\ln E[\mu_0(n, N)] = \ln \tfrac{1}{d^2 l^{2k-2}}.$$

If $k > 1$, then

$$\lim_{n,N \to \infty} lnE[\mu_0(n, N)] = -\infty.$$

Hence, $\lim_{n,N \to \infty} E[\mu_0(n, N)] = 0$ and there almost surely is at least 1 node in each square. The density of nodes will be:

$$\lambda = \tfrac{n}{l^2} = \tfrac{k \ln l^2}{d^2}, \text{ where } k > 1.$$

The connectivity will be satisfied if the communication range is greater than or equal to the expected maximum distance between two working nodes as proved in Theorem 1.

**Theorem 1**. A maximum distance between two active nodes that belong to the same layer is less than or equal to $(1 + \sqrt{5}) \times r_l$ if the density of nodes is satisfied as in Lemma 1.

*Proof.* Based on the condition of density derived in Lemma 1, we follow the complete proof of Lemma 3.1 and Theorem 3.1 in [10]. We assume that each square in Fig. 11 has only one node considering the worst case of Lemma 1 where $d = r_l$. Because there will be only one node per each square, node B is the furtherest node to be activated away from A in the gray square if A is activated in the dark square as shown in Fig. 12. However, if node C is activated earlier than B within the layering radius of $r_l$ from B, node B will sleep. The distance between node A and node C is the maximum distance where any two working nodes can be apart from each other. Hence, the communication range $r_c = (1 + \sqrt{5})r_l$ is a minimum range to guarantee the connectivity. The details can be found in [10].

In addition, the connectivity for one layer of Theorem 1 is a sufficient condition for the case of $K$-coverage.

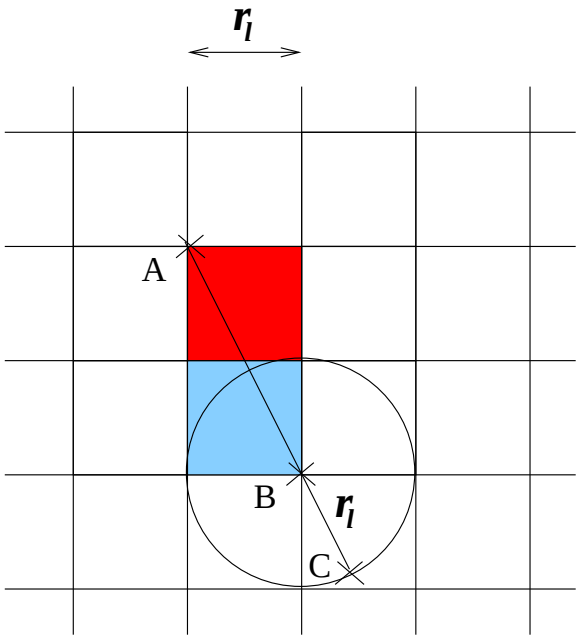



Fig. 12. Maximum Distance of Two Closest Working Nodes

### 2. Coverage

If a given sensor network has the density $\lambda$, we can scale down the density to $\lambda_l(1)$ for 1-coverage obtained by the ROA or C-ROA algorithm. Because ROA and C-ROA selects one node per area of $\pi(r_l/2)^2$, the scaled-down density will be $\lambda_l(1) = \frac{1}{\pi(r_l/2)^2}\lambda = \frac{4\lambda}{\pi r_l^2}$. According to the Poisson point process model, the coverage is the probability of empty node within the sensing radius of $r_s$. Hence, the percentage of 1-coverage will be:

$$\begin{aligned} R_c &= 1 - e^{-\lambda_l(1)\pi r_s^2} \\ &= 1 - e^{-4\lambda\alpha^{-2}}, \alpha = r_l/r_s. \end{aligned} \tag{3.1}$$

From (3.1), a minimum required density for $R_c$ almost surely is calculated by

$$\lambda = \frac{-ln(1-R_c)}{4} \cdot (\frac{r_l}{r_s})^2. \tag{3.2}$$

For $K$-coverage, one can easily calculate a required density by $\lambda_l(K) = \frac{K}{\pi(r_l/2)^2} \cdot \lambda$.

### 3. Simulation

Using 300 nodes deployed over a two-dimensional area of 500m×500m, we obtain the total energy consumed for the layering procedure and the average delay incurred for data packet delivery using the ns-2 simulator. The other options are 25m for $r_s$, 70m for $r_c$, and $K = 3$. Dynamic source routing (DSR) is used as a routing protocol. In Fig. 13, we compare the energy consumption for the reconfiguration. The C-ROAL protocol consumes a constant small energy for the layering procedure even though the reconfiguration periods vary from 50 to 1,000 seconds. However, the total energy consumption increases greatly as the reconfiguration period becomes shorter in the ROAL protocol. We also show the average delay of a data packet incurred by the

reconfiguration in Fig. 14. We can see that the packets are delivered much faster with the C-ROAL protocol, while the delay is increasing as the number of reconfiguration is increased in ROAL. The reason is because C-ROAL protocol never deters the packet delivery due to the autonomous circulation. We also expect that the difference of the total reconfiguration energy will be increased as the number of nodes increases. From the above results, we can say that the C-ROAL protocol can greatly improve the energy efficiency, while providing $K$-coverage with both the energy-balance and the fault-tolerance together for WSNs.

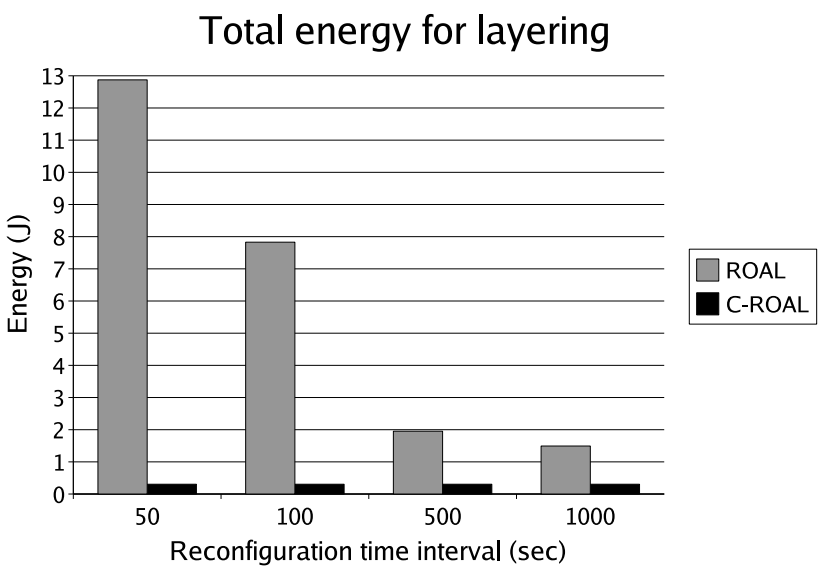



Fig. 13. Comparison of Energy Consumption for Reconfiguration

## D. Conclusion

In this chapter, we developed the C-ROAL protocol that can improve the fault-tolerance and energy-balance for $K$-coverage using a new circulation and C-ROA

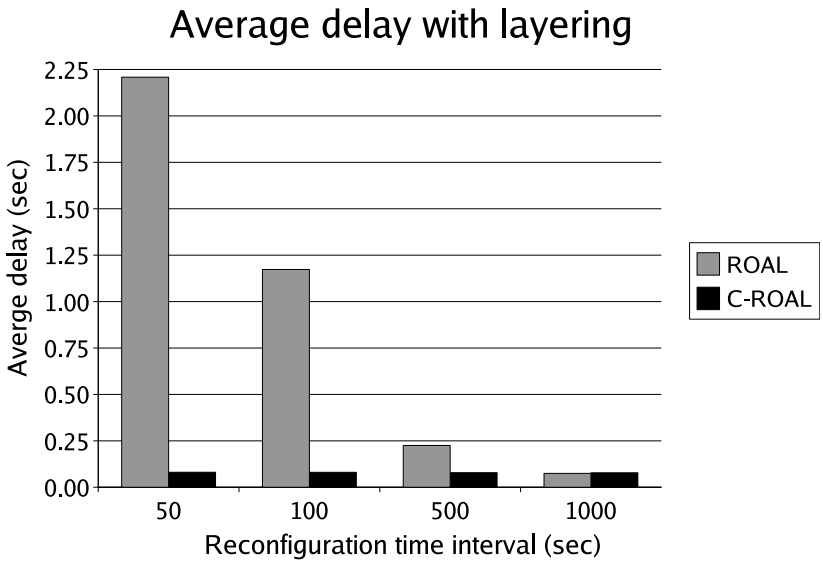



Fig. 14. Average Delay Incurred by Reconfiguration

scheme. We also proved that the C-ROAL protocol can guarantee the connectivity and coverage if a certain minimum density is satisfied regarding the sensing and the layering radius. The C-ROAL protocol can reconfigure the sets of working nodes with a greatly decreased energy consumption compared to the ROAL protocol. This property enhances the energy balance and fault tolerance for WSNs.

In future work, we will implement a more concrete strategy that can replace a fault node in a certain layer with a healthy node in other layers to stabilize the QoS during the network lifetime.

# CHAPTER IV

## MRCC: MOBILITY RESILIENT COVERAGE CONTROL FOR $K$-COVERAGE

In this chapter, we consider more realistic WSN environments where the sensor nodes are moving around, which can disappear due to wear-out failures. By enhancing a variant of the Random WayPoint (RWP) model [21], we propose Mobility Resilient Coverage Control (MRCC) to assure $K$-coverage in the presence of mobility. Our basic goals are 1) to elaborate the probability of breaking $K$-coverage with moving-in and moving-out probabilities, and 2) to issue wake-up calls to sleeping sensors to meet user requirement of $K$-coverage even in the presence of mobility. Furthermore, to show the impact of wear-out failures on the coverage achieved, we adopt a lognormal distribution to depict the conditional probability of failures and observe the influence of reduced numbers of active nodes on coverage. Our experiments with ns-2 show that MRCC achieves better coverage by 1.4% with 22% fewer active sensors than that of the existing Coverage Configuration Protocol (CCP). By taking the reliability of nodes into account, the performance drop with respect to coverage is 3.7% (for coverage $>$ 1), while the reduction in the number of sensor nodes is 18.19% when compared with pure MRCC. Comparing CCP and MRCC with reliability, we observe a 3.4% reduction in coverage for the average probabilistic case and 5.78% for the individual probabilistic case, while achieving a 12.82% and 28.2% reduction in number of nodes, respectively.

The rest of this chapter is organized as follows. Section A discusses the existing works and Section B introduces the basic concept and corollaries of $K$-coverage. In addition, we reformulate the average and individual probabilities in the mobility model. We will explain the experiments conducted with NS2 for our scheme in Section C. Finally, the conclusion and future work are mentioned in Section D.

## A. Background and Related Work

There are many obstacles to induce a fault in guaranteeing the coverage in real WSN environments. As an undermining factor of $K$-coverage requirement, we would like to focus on mobile sensor nodes with wear-out failures. Mobility with wear-out failures in a WSN are one of the sources that make solutions of the above problems harder [21, 22] and the same is true for Ad Hoc networks [23, 24, 25]. In solving the problem of mobility with failures, the biggest concern is that of maintaining a connected and covered network, while minimizing the power consumption so that the sensed data are safely delivered even in the presence of breaking the confidence of coverage and connectivity.

Span [26] was designed to adaptively elect coordinators among all the nodes in the network. Its goals are to ensure that sufficient coordinators are elected so that every node is within the radio range of at least one coordinator and to rotate the coordinators through the *withdrawal* mechanism in order to ensure that all nodes share the task of providing global connectivity. Based on Span, Coverage Control Protocol (CCP) [11] was devised to provide a specific coverage degree requested by an application with a decentralized protocol that only depends on local states of sensing neighbors. In contrast to stationary WSNs stated above, we consider mobility in guaranteeing $K$-coverage.

To properly model the mobility of a sensor or a vehicle, [27] suggested a scheme where mobile objects are uniformly distributed over a cell. Each sensor chooses a direction $\theta$ and speed $v$, uniformly at random in intervals $[0,2\pi)$ and $[0,V_{max}]$ respectively. With the optional operation of *thinking time*, the Random WayPoint (RWP) model similar to [27] has been a commonly used synthetic model for mobility in Mobile Ad hoc Networks (MANETS) [25, 28]. However, this model fails to provide a steady

state in that the average nodal speed consistently decreases over time, and therefore should not be directly used for simulation. So [29] suggested a modified Random WayPoint model to be able to reach a steady state. For the coverage problem, [22] chose a direction $\theta \in [0,2\pi)$ and a speed $v \in [0,V_{max}]$ according to distribution density functions, $f_\theta(\theta)$ and $f_V(v)$, respectively. A mobility model was used by [21] for choosing a WayPoint uniformly like RWP to build a robust connectivity topology-Minimum Spanning Tree. The approach of [22] is that a sensor sweeping a field can give better coverage. The difference between [22] and ours is that [22] considered sweeping an area with lack of sensors, while our scheme guarantees $K$-coverage all the time even if some sensors leave their duty area. In [30], authors suggested several algorithms that identify and minimize existing coverage holes based on Voronoi diagram and then compute the desired target positions where sensors capable of movement should move, while sensors in our scheme are unable to specify their destinations.

Certain work that has been proposed in literature with respect to mobility of nodes can be found in [31]. Here, *Random Walk, Random WayPoint, Random Direction, Gauss-Markov, and Probabilistic Random Walk* have been explained in reasonable detail, while [32] deals mainly with achieving steady state of movement with the help of a Random WayPoint model. [21, 22] suggest improvement in coverage achieved with mobility, and we further reinforce this argument with the help of theoretical formulations as well as simulations, by showing the same in the presence of both mobility and reliability considerations.

## B. Mobility Model and MRCC Protocol

Based on the general $K$-coverage used in the previous chapter, we want to state some corollaries giving the surplus number of sensors to assure the degree of coverage by

issuing wake-up calls to sleeping sensors for this work.

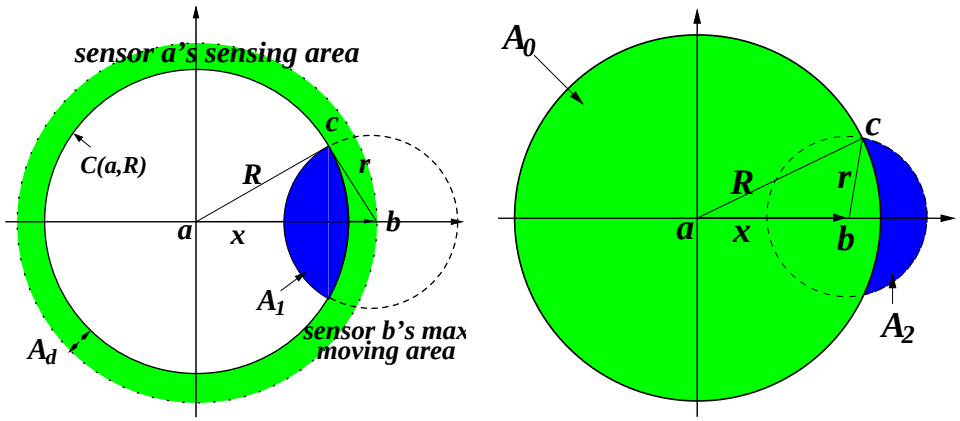
COROLLARY 1 *If a given topology of a WSN is assured by an optimal algorithm for $K$-coverage regardless of the distribution of a set of sensors, the active sensor node has to keep at least $K-1$ neighbors in its sensing range $R_s$ because the point of active sensor needs to be covered by $K-1$ neighbors and itself.*

COROLLARY 2 *In a given topology assured by an optimal algorithm for $K$-coverage, when a sleeping sensor initiates its sensing activity within its sensing range, it should have at least $K$ neighbors on already active duty in its sensing range by the definition of $K$-coverage.*

Based on the sufficient condition of $K$-coverage expressed in these corollaries, we plan to devise a mobility-resilient topology control with a modified Random Way-Point model for mobility in the following sections. To enforce the requirement of $K$-coverage, the above corollaries specify the number of active sensors required to prevent frailty of $K$-coverage.

### 1. Probabilities in Mobility Model

Like RWP [25], but unlike the Brownian- or RWP-like mobility model in [21], we consider two probabilities of sensors moving-in/out, as average and individual. Furthermore, we reformulate the moving-in probability with the location area $A_d$ of outside sensors which deemed to move in while [21] used $A_0$ in the conditional part. We compare the difference in calculating the moving-in probability with [21] and derive the average and individual probabilities with the following assumption of our mobility model; 1) All nodes are randomly distributed within a circle of area $A_0$ with sensing radius $R$ and the total number of nodes $N$ is known, 2) for a short-term interval of length $t$, each node moves independently toward a random direction in




(a) Moving In (b) Moving Out

Fig. 15. Probability of Moving In vs. Moving Out

$[0, 2\pi)$, with a constant speed $v$ that is uniformly distributed in $[0, v_{max}]$ and it may stay still for a while, and 3) the locations of sensors are known using an expensive GPS or a cheap method of trilateration.

Under these assumptions, within a range interval $r$, which is a function of time $t$, we can calculate two probabilities 1) that a new neighbor sensor $b$ moves into the detection range of node $a$ in Fig. 15 (a), and 2) that an existing neighbor $b$ moves out of the detection range $C(a, R)$ in Fig. 15 (b), $\overline{P}_{m.i}$ and $\overline{P}_{m.o}$, respectively. $C(a, R)$ is denoted as the circle of radius $R$ centered at point $a$ of node (a.k.a. point) $a$.

a. Probability of **M**oving **I**n $C(a, R)$, $\overline{P}_{m.i}$

Suppose node $a$ is located at point $a$ with its neighbor $b$ at point $b$, as shown in Fig. 15 (a). The maximum detection range of node $a$ is $R$ and the distance between node $a$ and $b$ is $x$, where $x$ is equal to or larger than $R$. Also, let point $c$ be an intersection of a circle made by point $a$ with radius $R$ and a circle made by maximum movement of point $b$ with velocity $v_{max}$ and time $t$. Then $\overline{bc}$ becomes $r = v_{max} \cdot t$ and the

probability that node $b$ moves into the detection range of node $a$ within time $t$ is the probability that node $b$ moves into the circle $C(a, R)$, which is exactly the shaded area between two circles as shown in Fig. 15 (a). This probability can be calculated in terms of the following two cases.

**Case I:** $0 < r < 2R$

$$\overline{P}_{m.i} = \int_R^{R+r} \frac{2\pi x}{A_d} \frac{A_1}{\pi r^2} dx = \int_R^{R+r} \frac{2A_1 x}{A_d r^2} dx, \tag{4.1}$$

where $A_1 = \alpha_1 R^2 + \alpha_2 r^2 - xR\sin\alpha_1, A_d = \pi((R+r)^2 - R^2) = \pi(2rR + r^2), \alpha_1 = \angle cab = \arccos\frac{x^2+R^2-r^2}{2xR}$, and $\alpha_2 = \angle cba = \arccos\frac{x^2+r^2-R^2}{2xr}$.

**Case II:** $r \geq 2R$

$$\begin{aligned}
\overline{P}_{m.i} &= \int_R^{r-R} \frac{2\pi x}{A_d} \frac{\pi R^2}{\pi r^2} dx + \int_{r-R}^{r+R} \frac{2\pi x}{A_d} \frac{A_1}{\pi r^2} dx \\
&= \int_R^{r-R} \frac{2\pi R^2 x}{A_d r^2} dx + \int_{r-R}^{r+R} \frac{2A_1 x}{A_d r^2} dx \\
&= \frac{R^2(r-2R)}{r^2(r+2R)} + \int_{r-R}^{r+R} \frac{2A_1 x}{A_d r^2} dx.
\end{aligned} \tag{4.2}$$

The first fraction in (4.1) explains the conditional probability about the existence of a sensor at point $x$ and the second fraction is the ratio of area $A_1$ to total area of node $b$'s movement. Unlike [21] in (4.1) and (4.2), we considered $A_d$ as a conditional probability because the probability of location of outside sensors is represented by $A_d$, not $A_0$. The first term of (4.2) considers the case that the movement circle is larger than the circle of sensor $a$ so that the former circle includes the latter. The second term in (4.2) represents a situation where there is an intersection between the movement circle and the sensing circle of sensor $a$.

b. Probability of **M**oving **O**ut $C(a,R)$, $\overline{P}_{m.o}$

The probability that one of the neighbors $C(b,R)$ moves out of the detection range of node $a$ within time $t$ is the probability that node $b$ moves out of circle $C(a,R)$, more specifically, which is the shadowed area outside of the detection circle made by node $a$, as shown in Fig. 15 (b). There are three possibilities because the range of the node either 1) intersects, 2) includes, or 3) intersects and includes the given node in the following manner,

**Case I:** $0 < r < R$

$$\overline{P}_{m.o}=\int_0^R \frac{2\pi x}{A_0}\frac{A_2}{\pi r^2}dx = \int_{R-r}^R \frac{2A_2 x}{A_0 r^2}dx, \tag{4.3}$$

where $A_2 = (\pi-\alpha_2)r^2 - \alpha_1 R^2 + xR\sin\alpha_1$.

**Case II:** $R \le r < 2R$

$$\begin{aligned}\overline{P}_{m.o}&=\int_0^{r-R} \frac{2\pi x}{A_0}\frac{\pi(r^2-R^2)}{\pi r^2}dx + \int_{r-R}^R \frac{2\pi x}{A_0}\frac{A_2}{\pi r^2}dx \\ &=\frac{\pi(R+r)}{A_0 r^2}(r-R)^3 + \int_{r-R}^R \frac{2A_2 x}{A_0 r^2}dx.\end{aligned} \tag{4.4}$$

**Case III:** $r \ge 2R$

$$\overline{P}_{m.o} \;=\; \int_0^R \frac{2\pi x}{A_0}\frac{\pi(r^2-R^2)}{\pi r^2}dx = \frac{\pi(r^2-R^2)R^2}{A_0 r^2}. \tag{4.5}$$

In (4.4), the first term shows that the center of the moving circle (larger than $C(a,R)$) ranges from 0 to $r-R$ resulting in the moving circle encompassing the circle $C(a,R)$, while the second term accounts for the intersection between the moving circle and circle $C(a,R)$.

Fig. 16 depicts the functions, $\overline{P}_{m.i}$ and $\overline{P}_{m.o}$, considering (4.1) to (4.5) according to $r$ for $R_s = 10$ whose value decides the stiffness of these functions, not the monotonic

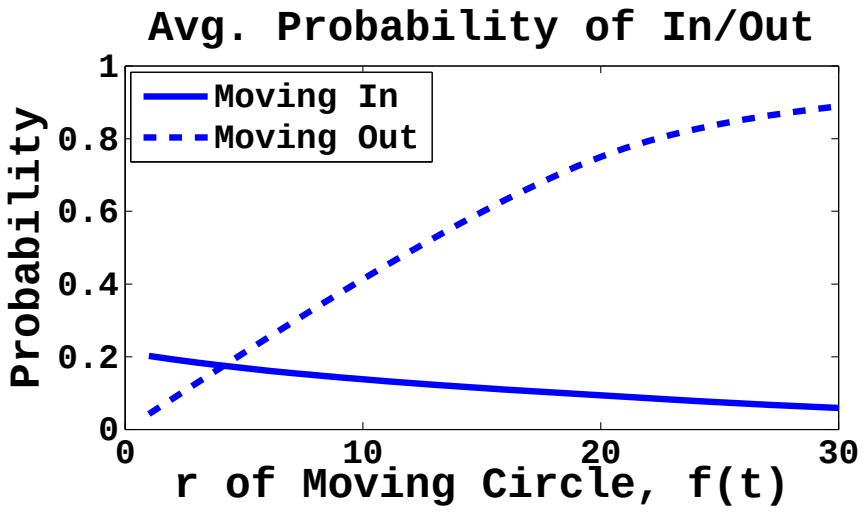



Fig. 16. $\overline{P}_{m.i}$ vs. $\overline{P}_{m.o}$

increase or decrease. While $\overline{P}_{m.i}$ starts at 0.2 and decreases gradually, $\overline{P}_{m.o}$ increases expeditiously. (4.1) through (4.5) have been devised for average probability, meaning that regardless of the distance given by variable $X$, from the area of interest, every sensor has the same probability. But, intuitively, at given time $t$, sensors outside or inside the rim have a larger probability of moving in or out, respectively. Therefore, if we specify this individual probability of moving in and out, each sensor can make a more accurate decision. This insight can be formalized in the following equations,

$$P_{m.i|X=x} = \begin{cases} \dfrac{A_1}{\pi r^2} & 0 < r < 2R, R \leq x < R + r \\ \dfrac{R^2}{r^2} & r \geq 2R, R \leq x < r - R \\ \dfrac{A_1}{\pi r^2} & r \geq 2R, r - R \leq x \leq r + R \end{cases} \tag{4.6}$$

and

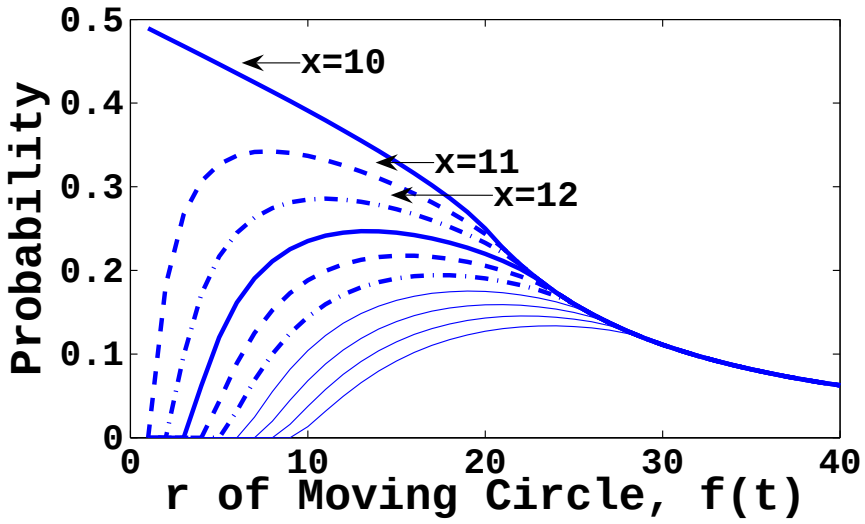
$$P_{m.o|X=x} = \begin{cases} \dfrac{A_2}{\pi r^2} & 0 < r < R, R - r \leq x < R \\ \dfrac{(r^2 - R^2)}{r^2} & R \leq r < 2R, 0 \leq x < r - R \\ \dfrac{A_2}{\pi r^2} & R \leq r < 2R, r - R \leq x \leq R \\ \dfrac{(r^2 - R^2)}{r^2} & r \geq 2R, r - R \leq x \leq R. \end{cases} \tag{4.7}$$

Fig. 17 shows the individual probabilities, $P_{m.i}$ and $P_{m.o}$ with $R = 10$, plotted against with $r$. Depending on the distance to a particular sleeping sensor, $P_{m.i}$ has peak value, while $P_{m.o}$ shows only a gradual increase regardless of $r$.

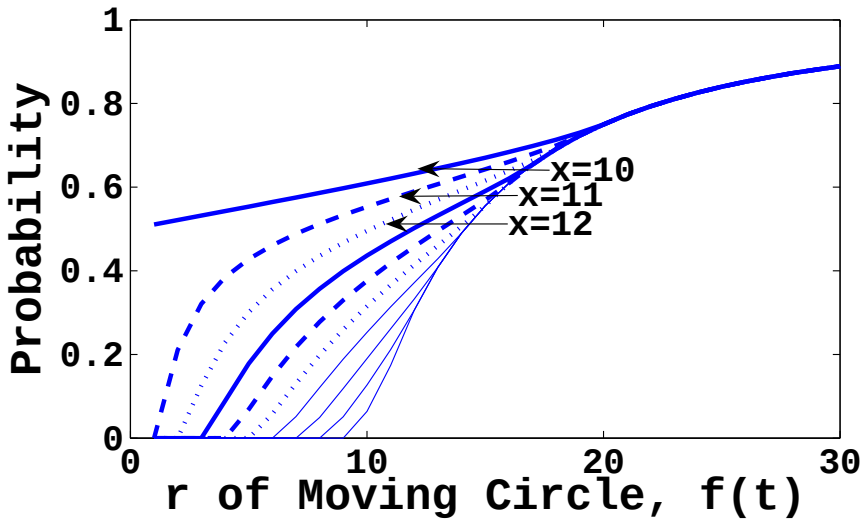
c. Deciding Wake-up Call

After the probing operation for deciding the duty sensors for $K$-coverage, each sensor's destination is determined whether it is active or asleep. *Corollary* 1 and 2 require each sleeping sensor to be $K$-covered and, therefore, it can observe the number of active **in**side sensors $N_{in}$ in its radius $R_s$. For $N_{ot}$, we can know the number of **o**u**t**side active sensors, $N_{ot}$, according to *Theorem* 1 of [33] because to be connected $R_c \geq 2R_s$ must be satisfied and the distance between sensors can be measured based on the transmission signal during $K$-coverage configuration.

Based on the equations about average probabilities, (4.1) through (4.5), of sensors moving in and out, now we can formulate the probability of breaking the $K$-coverage at the point of a sleeping sensor, given $N_{in}$ and $N_{ot}$. Suppose there are random variables, $N_{m.o}$ and $N_{m.i}$, about sensors **m**oving-**o**ut from inside and sensors **m**oving-**i**n from outside, respectively. Then, the probability of breaking $K$-coverage $\overline{P}_B$ considering two random variables, $N_{in}$ and $N_{ot}$, gives us the following two cases, one of which is for $N_{in} = K$ and the other for $N_{in} = K + \alpha$,




(a) Moving In




(b) Moving Out

Fig. 17. Probability of Moving In/Out According to Distance of Active Sensor

$$\overline{P}_B = \begin{cases} \sum_{l=1}^{N_{in}} P(N_{m.i} < l \mid N_{m.o} = l) P(N_{m.o} = l) \\ \sum_{l=\alpha+1}^{N_{in}} P(N_{m.i} < l - \alpha \mid N_{m.o} = l) P(N_{m.o} = l), \end{cases} \tag{4.8}$$

where $P(N_{m.i} < l \mid N_{m.o} = l) = \sum_{j=0}^{l-1} \binom{N_{ot}}{j} \overline{P}_{m.i}^{j} (1 - \overline{P}_{m.i})^{N_{ot}-j}$ and $P(N_{m.o} = l) = \binom{N_{in}}{l} \overline{P}_{m.o}^{l} (1 - \overline{P}_{m.o})^{N_{in}-l}$.

The range interval $r$ which is calculated by $v_{max} \cdot t$, determines the probability $\overline{P}_{m.i}$ and $\overline{P}_{m.o}$. Therefore, the maximum probability with information about both, constant $R$ and an active sensor's distance from a sensor deemed to be sleeping. Now (4.8) can be expanded considering the individual probabilities, i.e., (4.6) and (4.7) by removing $\binom{N_{ot}}{j}$ and $\binom{N_{in}}{l}$.

As expected from Fig. 16 and summation in (4.8), the function in (4.8) is monotonically decreasing. Therefore, we set some value of adjustable threshold to awaken the sleeping sensors.

2. MRCC Protocol Mechanism

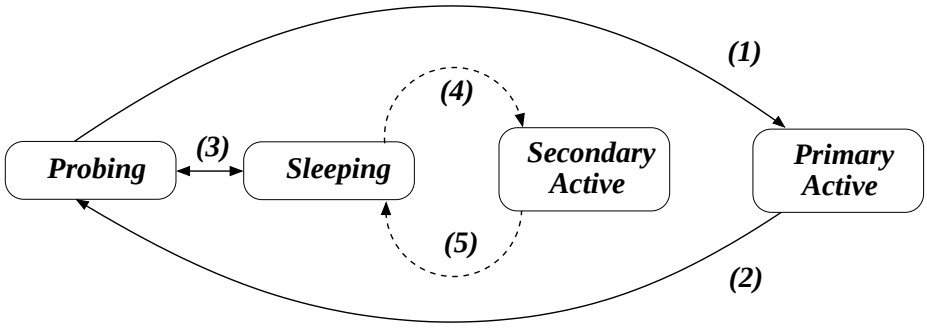



Fig. 18. State Transition Diagram for Each Sensor in MRCC

Usually a topology control mechanism resides and operates between the routing

layer and the MAC layer so that the routing layer uses information of rigid infrastructure of a WSN secured by the topology control such as [33, 26]. Besides the basic mechanism of electing active sensors implemented in [33, 26], we need an additional ruling of the sensor state transition from sleeping to active, as shown in Fig. 18.

As suggested in [33, 26], three solid lines with (1), (2), and (3) in Fig. 18 are the transitions suggesting election of *primary active* sensors to provide guaranteed coverage over the whole field. *Probing* sensors that want to be *primary active* sensors need to be desynchronized to figure out their roles; otherwise, a group of sensors are blindly set to wake up to cover the same coverage hole at the same time. Once the role of each sensor is decided, we need to choose some of the *sleeping* sensors that have their breaking probability calculated by (4.8) to be lower than a predefined threshold value. Two dotted lines of (4) and (5) explain these situations where each of the sleeping sensors asynchronously figures out the probabilities to become a *secondary* sensor like (1), (2), and (3).

### 3. MRCC Protocol with Wear-out Failures

In addition to the mobility of sensor nodes, the associated reliability measures in the form of wear-out failures should be also considered. A lognormal distribution has been found to be a more effective model than an exponential distribution for degradation processes common to semiconductor failures because of the multiplicative degradation argument [34]. Moreover, the memoryless property of an exponential distribution makes it unsuitable for modeling semiconductor reliability [35].

To obtain the lifetime distribution for the processor as a whole, combining the effects of the individual lognormal distributions across all the mechanisms and structures is necessary. The standard lognormal distribution for some $\mu$ and $\sigma$ can be given as

$$f(t) = exp(-(\ln t - \mu)^2/(2\sigma^2))/(t\sigma\sqrt{2\pi}), \tag{4.9}$$

where $\mu$ is the mean lifetime of a component and $\sigma$ is the shape parameter. (4.9) represents the reliability of each component, such as the embedded processor and RF transceiver circuit of a sensor node.

Given the nature of operation of WSNs, a sensor node is not supposed to run all the way until its battery power is exhausted. Factors such as the overhead of wake-up have led us to consider a posteriori probability and a cumulative operation time, $t_i$, meaning that in off-duty the timer for reliability calculation of a sensor stops and the timer is restarted once the sensor is selected for the next duty cycle as a coordinator. Therefore, the probability of a failure in the time interval $(T_1, T_2)$, given that the sensor has survived up to the time $T_1$, can be expressed as

$$F_{T_1,T_2} = \frac{\int_{T_1}^{T_2} f(t)dt}{R(T_1)}. \tag{4.10}$$

A conditional probability of survival, i.e., reliability, in the interval $t'$ and $t$ is $R(t \mid t') = R_{t,t'} = 1 - F_{t',t}$.

As the sensor nodes are independently configured for $K$-coverage, the reliability $R(t)$ of a certain sensing area is calculated as a product of reliability of the active sensor nodes involved in coverage. Since the system has lived up to $t'$, the conditional reliability $R(t \mid t')$ of the system, to live until $t$ can be given by

$$
\begin{aligned}
R(t \mid t') &= R(T > t \mid t') \\
&= R(t_1 > t, t_2 > t, \cdots, t_K > t \mid t') \\
&= R(t_1 > t \mid t'_1) R(t_2 > t \mid t'_2) \cdots R(t_K > t \mid t'_K) \\
&= R_1(t_1 \mid t'_1) R_2(t_2 \mid t'_2) \cdots R_K(t_K \mid t'_K) \\
&= \prod_{i=1}^{K} R_i(t_i \mid t'_i), \qquad (4.11)
\end{aligned}
$$

where $t_i = t + \epsilon_i$ and $t'_i = t' + \varepsilon'_i$. $\epsilon_i$ and $\varepsilon'_i$ are scaling factors for cumulative spent time in each sensor $i$ except the off-duty time because each sensor has different spent time and, therefore, different next survival time. If the number of active sensors for $K$-coverage is greater than $K$, we can expand (4.11) by considering the combination of $K'$ sensors taken $K$ at a time.

The rationale of using conditional reliability in (4.10) for $K$-coverage is that, at the beginning of the selection for the next duty cycle, each sensor alive feeds its information including cumulative lifetime to its neighbors, notifying that it has survived. The cumulative reliability is given by (4.10), which is the product of conditional reliabilities of active sensors contributing to $K$-coverage of the WSN. A drawback of this scheme is the overhead of broadcasting packets containing related information.

## C. Experimental Results

In this section, we use ns-2 implementation of Span [26] and CCP [33] as an optimal algorithm to assure $K$-coverage in a variety of coverage degree cases.

Fig. 19 shows the benefit of using average and individual probabilities in the situation of breaking 3-coverage. In the figure, first two curve lines explain the mathematical plots for (4.8), while the third line is made from the actual breaking probability of ns-2. As the figure reveals, there is a gap between average and individual

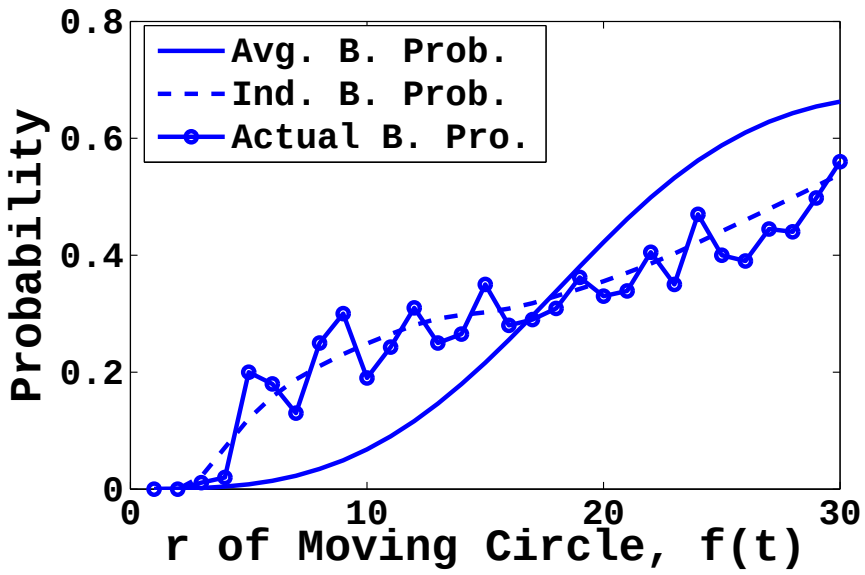


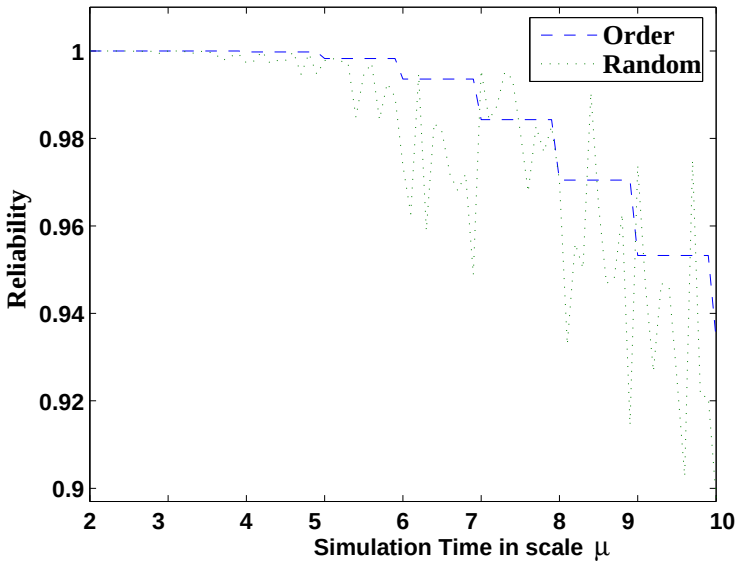

Fig. 19. Comparison of Three Breaking Probability in 200 Runs

probabilities. Furthermore, the dotted line shows the same behavior with individual probability.

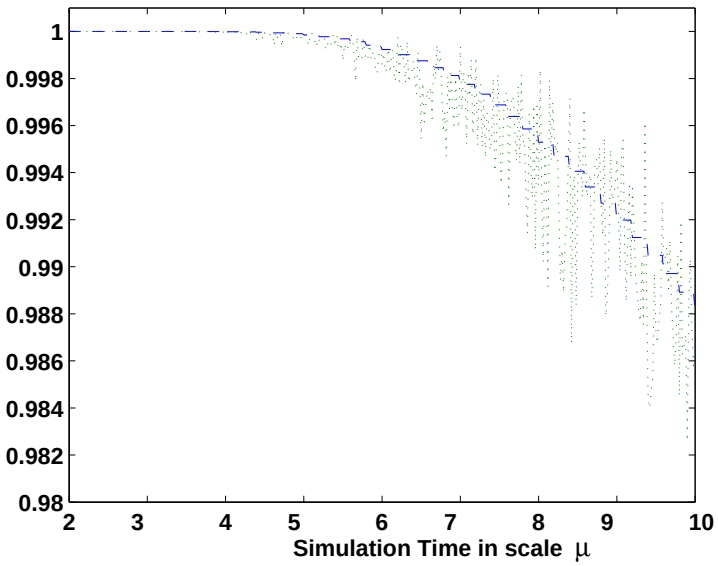
The conditional reliability defined in Section 3 assumes that a sensor has survived up to the time $T_1$ in selecting the working sensors for the next duty cycle. Due to the conditional probability, the value of $R_{T_1,T_2} = R(T_2 \mid T_1)$ is relatively large at the beginning, but gradually decreases as the duty cycle goes by.

The experiments we conducted in Fig. 20 is to justify the choice of shorter duty cycles to guarantee the reliability at each duty, at the cost of packet overhead for probing. At each different duty cycle, the selection of active sensors for $K$-coverage is based on the spent lifetime marked as "Order" or regardless of spent lifetime marked as "Random," where "Random" in $0.25\mu$ has the worst variation of reliability. The impression given by Fig. 20 is that the "Random" case shows larger fluctuations in measured reliability during operation.

The simulation results in Table I validate the miscalculation of [21], as well as




(a) Duty Cycle: $0.25\mu$




(b) Duty Cycle: $0.05\mu$

Fig. 20. Comparison of Large ($0.25\mu$) vs. Small ($0.05\mu$) Duty Cycle in the Long Run

Table I. Comparisons of $A_d$ vs. $A_0$ and Avg. vs. Ind. Prob.

| **Prob. Scheme:$A_0$vs.$A_d$** | **Coverage Achieved(%)** | **No. A. N.** † |
|---|---|---|
| MRCC w/ $A_0$ | 90.6*/94.7** | 67/89 |
| MRCC w/ $A_d$ | 96.6/97.3 | 65/84 |
| **Coverage Scheme** | **Coverage Achieved(%)** | **No. A. N.** † |
| CCP | 93.8*/92.7** | 87/90 |
| MRCC w/ Avg. P. | 94.7/93.2 | 77/88 |
| MRCC w/ Ind. P. | 94.4/94.1 | 71/84 |
| CCP w/ Rel. | 96.015*/92.94** | 76/79 |
| MRCC w/ Avg. P. and Rel. | 97.883/89.75 | 63/72 |
| MRCC w/ Ind. P. and Rel. | 96.36/87.56 | 53/58 |

* 1-Coverage ** 2-Coverage † Number of Active Nodes among 120

the probability behaviors of Fig. 16, 17, and 19. For this experiment, the sensor field has an area of $400m \times 400m$. It has also been observed during the simulation of $112s$ that, after a certain period for enough given energy, the average number of active nodes remains almost constant with a standard deviation of a constant factor. This is due to the fact that the cycle of selection and withdrawal of sensors reaches approximate synchrony and, hence, almost an equal number of nodes are added and withdrawn. The first half of Table I validates the wrong conditional equation in [21] by the difference in the number of active sensors and achieved $K$-coverage. Therefore, moving-in probability needs to use the area of outside where moving-in sensors are

located. The second half of Table I shows that, although the difference in coverage achieved in the three cases is marginal, the number of active sensors for 1- and 2-coverage show significant difference, supporting our claim that individual probabilities of moving-in/out is reasonably good in selecting an optimal number of sensors for an energy-efficient coverage scheme.

Finally, the third half of Table I shows the coverage achieved in all three cases when the reliability of nodes is taken into consideration. The results show that the number of active nodes selected for coverage has reduced considerably. This is a consequence of our scheme, which selects only the required number of sensors for $K$-coverage and considering duty cycle of operation of nodes. This translates to immense savings in terms of average power consumed, which, in turn, results in the longevity of the lifetime of the network. Note that the number of wear-out failures over a period of time $t$, reduce due to an increase in the overall lifetime of the network. However, there is a slight degradation in the coverage performance when higher coverage is required.

## D. Conclusion and Future Work

In this chapter, we have addressed the problem of achieving $K$-coverage in the presence of mobile sensors with wear-out failures. Mobility of sensors is one of the major hurdles in achieving $K$-coverage due to the uncertainty of the location of the sensor nodes for the coverage.

We have proposed the Mobility Resilient Coverage Control (MRCC) protocol that reduces the number of active sensors significantly, while providing a higher achieved coverage than previous approaches. We've also considered wear-out failures of the nodes along with mobility to select the number of sensors to be active for $K$-coverage,

and have shown the improvement in reducing the number of active nodes with a slight trade-off in terms of coverage achieved. Therefore, with less numbers of sensors for the same coverage, we believe that a significant amount of power can be saved, leading to the longevity of lifespan of a WSN. Our experiments with NS2 show that MRCC achieves better coverage by 1.4% with 22% fewer active sensors than that of the existing Coverage Configuration Protocol (CCP). By taking wear-out failures of sensor nodes into account, we observe that MRCC with reliability uses 28.2% less active sensor nodes than CCP with reliability.

We plan to measure power savings obtained by considering battery depletion, as well as how to optimally decide the wake up time based on residual power of a sleeping sensor.

# CHAPTER V

## MULTIPLE-CDS TOPOLOGY FOR WIRELESS SENSOR NETWORKS

In this chapter, we present a Multiple-connected dominating set (MCDS) topology protocol, which is designed to provide a load balancing specifically for the on-demand routing protocols. We define the problem of multiple connected dominating sets (MCDSs) and propose greedy and heuristic algorithms that form the MCDSs in three stages. Our protocol runs in $O(\Delta^2)$ computation time with $O(\Delta)$ message overheads, where $\Delta$ is the maximum degree of the network graph.

The properties on load balancing and network performance are evaluated through ns-2 simulator. From the results, we show that the energy depletion ratio of nodes is significantly decreased, while conserving more residual energy. The network throughput is improved by 35% and the routing overhead is decreased by 71% as the density of nodes increases. We also observed the increment in packet delivery ratio more than 2 times when the degree of nodes is increased.

The rest of this chapter is organized as follows. Section A discusses the related existing works that are studied to provide the load balancing to WSNs. In Section C, we introduce the concept of MCDS topology with the technique used to construct it. Section D deals with our suggested algorithm for the MCDS topology. Finally, we provide our simulation results on the energy and network performance in Section E, and conclude our work in Section F.

### A. Background and Related Work

Generally, routing protocols can be categorized into two main categories such as proactive and reactive protocol [36]. Proactive protocol characterizes all possible connectivity to all existing nodes in the network in advance. On the other hand,

reactive, also called on-demand, protocol discovers a route when a source node needs to send its packet by broadcasting a route discovery packet. While this routing protocol has the purpose of finding a reachable path between any two nodes, the ultimate goal is to minimize path length, route traffic, and packet loss, etc. On wireless sensor networks, however, we need to consider a balanced network traffic to prolong the network lifetime under the energy constraints of WSN.

## B. Related Work

Proactive approach to the load balancing is proposed in the studies of GLIDER [7] and MAP [8]. The motivation of both works arose from the geographic forwarding protocol. In a geographic forwarding scheme such as GPSR [6], the next hop is a neighbor node that is closest to a destination node. Hence, the geographic greedy routing can cause a hot spot around the boundary of a void region (obstacle). Hence, a main idea of both protocol is to build a virtual coordinate system that can reflect the shape of a given network topology through their own topology abstraction. The abstraction method of GLIDER is to subdivide the original topology into several voronoi cells of randomly selected landmarks. After then, each node is assigned with a new virtual coordinate that is calculated to represent a relative distance from the corresponding landmark. MAP abstracts a medial axis composed of nodes that forms a center line within the network topology. Hence, MAP needs to find all boundary nodes first and assigns a virtual coordinate to each node using a relative distance from the axis and a relative position along the axis, i.e., x and y coordinates in a two-dimensional region. Each node, then, can deliver packets based on the retrieved virtual coordinate in a geographic forwarding strategy as in GPSR. They have proved that the network traffic can be distributed over multiple paths around the boundary

of the void region. The problem is that the abstraction of the global topology with a new coordinate could cause a high volume of messages with the increment of the density and size of the network.

AOMDV [37] is a multipath on-demand routing protocol, which is extended from the AODV protocol [38] for establishing multiple disjoint paths between any pair of source and destination. Multiple paths are constructed by one-time flooding of a route discovery packet and multiple route reply messages from the destination. All intermediate nodes on the reverse paths of the route reply messages retrieve any possible disjoint and loop free path. Hence, each node will retain multiple next hop nodes for each destination and source. Each node, then, forwards a data packet through one path at a time until all paths are broken. The cached route information can reduce the frequency of route discovery for the same pair of sources and destinations. Authors claimed that AOMDV can be used for load-balancing when all multiple paths are used at the same time. However, packet reordering at a destination and the amount of traffic splits for each path are issued as a future work. A probabilistic forwarding for load-balancing is studied in [39]. In their Gossip-based routing, each node forwards packets with some probability. This approach can reduce the message overhead on route discovery. The problem is that network throughput and connectivity can be degraded without the choice of a proper forwarding probability as the density of node changes.

Recently, many approaches, with the purpose of energy conservation, have suggested to construct and use a connected dominating set (CDS) as a backbone network for the reactive routing protocol to conserve the energy consumptions used for the broadcast. A non-CDS node is covered (within the transmission range) by at least one CDS node. Nodes in CDS are responsible for forwarding all packets. Many studies have researched the algorithms to construct a minimum CDS to derive the smallest

communication subgraph of a given network so that the network can conserve more energy [40, 41, 42].

A necessity for a backup link in CDS-based topology construction has been found in the study of [42]. The study suggested $k$-connected $k$-dominating set algorithms for the construction of more robust CDS-based backbone network. The study suggested three different algorithms, but the best algorithm needs $O(k\Delta^4)$ computation time, where $\Delta$ is the maximal node degree [42].

Our approach is unique in that we can build multiple CDSs and use them as distributed route backbones. A source node can forward its packet through one CDS to which it belongs, while other CDSs only can receive the packet. Intuitively, the network traffic will be uniformly distributed over all CDSs as long as all nodes are uniformly distributed over the constructed CDSs. In terms of the balanced network traffic, the MCDS protocol can provide load balancing in addition to the effect of energy conservation.

## C. Protocol Overview

### 1. Route Discovery in DSR Protocol

Dynamic source routing (DSR) [17] is a well-known, on-demand multi-hop routing protocol that uses source routing that places a complete path information in the header of a data packet. The path information is the sequence of hops that the packet should follow to its destination. In on-demand routing protocols, a source node initiates a route discovery procedure only when a source node wishes to send a packet to a destination and does not know a route to the destination. Normally, the source route is searched in a route cache of routes previously learned or a source node will initiate the route discovery procedure to find a new route only when no route is

found in its cache.

To initiate the route discovery, a node transmits a ROUTE REQUEST (RREQ) message, which is a single local broadcast packet, to its neighbor nodes. Each RREQ packet is identified with a unique *request id* (RID) that is determined by the source node of the RREQ, and contains the identifiers of the source and destination node. During the route discovery procedure, the address of each intermediate node through which a particular RREQ message has traveled is recorded in the route record of the RREQ message. A ROUTE REPLY (RREP) message is returned to the source node by another node that receives the RREQ message if the receiving node is the destination of the RREQ message. The accumulated route record of the RREQ is copied to the RREP message and cached in the route cache of the source node. The destination node can either discover a new route to the source node or simply reverse the sequence of hops in the route record to send its RREP message. In receiving the RREQ message, any intermediate node searches its own route cache for a route to the target before forwarding the RREQ. If there is a route to the destination of the RREQ, the node returns a RREP to the source node rather than forwarding the RREQ.

During the route discovery procedure, DSR can cache multiple paths per route discovery replied by multiple intermediate nodes, which could result in reply storm, thereby wasting bandwidth and increasing collisions. To prevent the route reply storm, a node delays sending its own RREP for a random period while receiving and looking for data packets from the source node of the route discovery. If it receives such a data packet that has a route length less than or equal to its own route length from the source to destination, the node discards its own RREP.

The route cache can incur a hot-spot route that may be heavily used to a certain destination. Without using the route cache, instead, the performance of DSR will be

hurt by the overhead related to the route discovery procedure.

## 2. CDS and CDS Layers

As we reviewed the DSR protocol, there is no fundamental method to distribute the energy burden in terms of load balancing over the sensor nodes. We study a novel method that can balance the network load of DSR protocol using the concept of connected dominating set (CDS).

The construction of MCDS topology is highly related to the CDS problem. Assume that a unit disk graph $G(V, E)$, where an edge exists between two nodes if and only if the two nodes locate within the communication range $r_c$. A CDS is a subset $S \subseteq V$ such that $S$ is a dominating set of $G$ and the subgraph of $G$ induced by $S$ is connected. In other words, a node not in $S$ has at least one corresponding edge in $E$ to a node in $S$, and $S$ is a connected subgraph of $G$.

To obtain a minimum CDS is known to be NP-hard [43]. Because of the hardness of the problem, many approximation algorithms have been proposed to construct a small CDS [44, 45, 42, 40]. However, these protocols are hard to apply to achieve multiple CDSs for the reasons of a relatively high cost of computation time, centralized approach, and message overhead.

We propose a greedy dominating-and-connecting (DAC) heuristic. The construction procedure of CDS using the DAC method is shown in Fig. 21. Assume that every node in the figure is ordered by a randomly generated decision time from $A$ to $F$ in ascending order. To form a CDS, node $A$ elects itself as a dominating node because no other node has been elected as a dominating node within its transmission region ($r_c$). Next, node $B$ elects itself as a dominating node. Among the neighbor nodes within the transmission range of $A$, node $C$ and $D$ are elected as a connecting node for $A$. Next, node $E$ also elects itself as a dominating node for a different CDS because it

already has both dominating node $A$ and connecting node $D$. Node $F$ should be a connecting node for both $A$ and $B$. Here, nodes $A$, $B$, $C$, $D$, and $F$ form one CDS and the other nodes will form another CDS. In this study, we call one CDS among the constructed multiple CDSs a layer. For a practical reason, we construct MCDS topology by allowing a node to be shared by multiple CDSs.

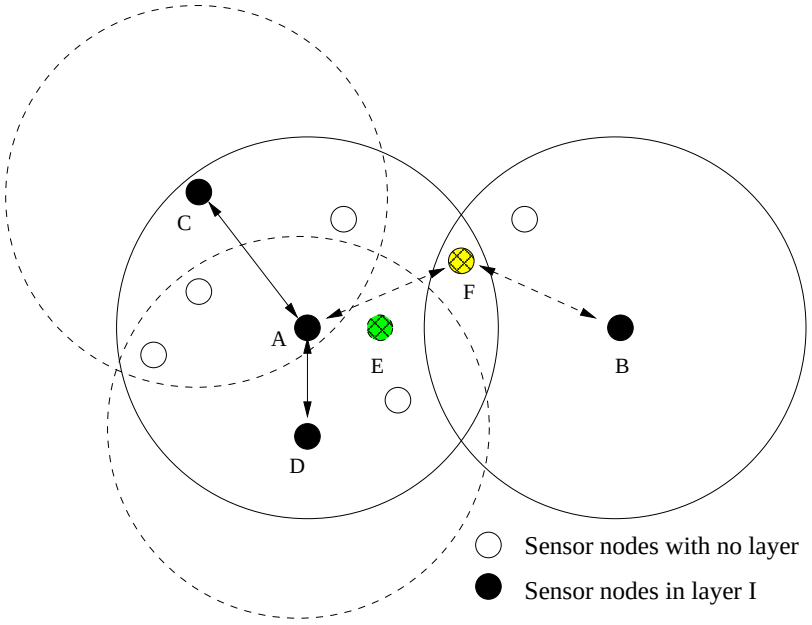



Fig. 21. The Concept of Dominating-And-Connecting (DAC)

### 3. MCDS Topology for DSR Protocol

In Fig. 22, assume there are two CDSLs and the black and white nodes consist of each different CDSL, respectively. The white nodes in the figure form one CDS as layer A and the black nodes form another CDS as layer B. A RREQ packet generated by node B2 will be propagated through only the black nodes. If node A1 is targeted by node B2, it can listen to the RREQ packet of B2 even though the layer where it belongs is different from layer B. When A1 listens to the RREQ of B2, it replies to the RREQ by sending RREP along with a reverse path of the RREQ, which is mainly

comprised of nodes in layer B. On the other hand, a route discovered by a node in layer A will have a data path comprised of nodes in layer A. Hence, we can expect that resultant routes discovered by the route discovery procedure can be distributed evenly over the multiple CDSLs if source nodes are evenly located over all CDSLs.

Each layer is differentiated by a layer identification number (LID) that is integer value corresponding to the level of each layer in a hierarchical view. Every node is assigned with one or more LIDs and forwards a RREQ packet if and only if the RREQ has the same LID with one of its LIDs. When a source node sends its RREQ, it includes its default LID to identify one layer through which the RREQ is forwarded.

A novel feature of MCDS topology is placed on the effect of saving the total energy of network in addition to the load balancing. Each constructed CDSL takes a role of network-wide backbone that can reduce the number of forwarding nodes, which can bring out the effect of energy conservation.

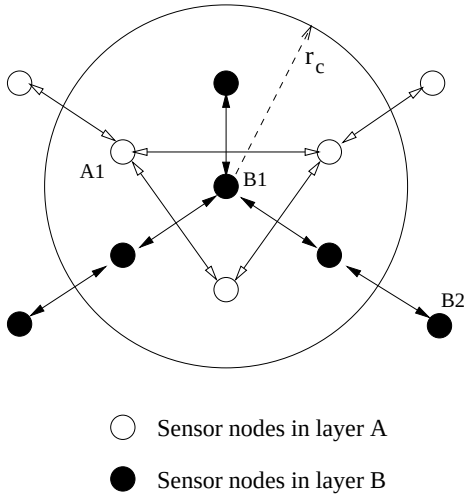



Fig. 22. Concept of Layer Communication

## D. MCDS Protocol

We make the following assumptions. First, every node is synchronized as explained in the previous section. Second, every node has a unique identifier (UID). Note that our algorithm assigns a layer identifier (LID), in addition to the UID, to every node to distinguish each layer.

The MCDS protocol constructs multiple CDSs via a layering procedure, and provides a fault management scheme that restores a broken connectivity among the CDSs. We implement three different algorithms such as primitive layering (PL), intermediate layering (IL), and runtime-layering (RL) algorithms, each of which run in a corresponding construction stage. Each node proactively builds a primitive state of CDSs during the first two stages and the boundary of layer or disconnected region in each layer will be connected using a normal route discovery procedure during the last stage.

Each node decides its LID when a randomly generated decision time (DT), which is generated in every node between 0 and $T_L$, expires. The period of $T_L$ can be easily calculated if we assume that the DT follows a uniform distribution. To calculate a proper value of $T_L$, we also assume that the position of dispersed nodes follows a two-dimensional Poisson process having density $\lambda$. The expected number of nodes ($N$) within the circular region of transmission range $r_c$ is $\lambda\pi r_c^2$. Therefore, the length of $T_L$ should be greater than $\frac{r_c}{c} \times 4N$, where $c$ is the propagation speed of wireless signal. In this formula, $4N$ is the number of nodes in the circular region of radius $2r_c$ that is a maximum distance within which a random position could receive two signals from two different nodes at a time.

To maintain the state of layer construction of its neighbor nodes located within its communication range, every node maintains a Layer Table (LT). Each record of

LT contains the entries related to the layering procedure, such as a list of UIDs of connected neighbor nodes ($CN$), a list of UIDs of disconnected neighbor nodes ($DN$), a flag of $forwarding$, a flag of $boundary$, and an integer field of $uid$. The role of each entry will be explained later. After now, we will remark a record indexed by $m$ as $DN(m)$.

### 1. Stage 1: Primitive-Layering (PL) Procedure

When algorithm Primitive-Layering (PL) starts, DT is generated randomly between 0 and $T_L$ at each node. Before its DT expires, if $v$ receives an announcement packet (AP) from its neighbor nodes, it updates its $CN[m]$ and $DN[m]$ for the layer $m$ of AP. AP is a 1-hop broadcast packet that carries LID, UID, and the list of layer neighbor ($LN$) of a sending node.

**Stage 1. Algorithm Primitive-Layering (LT, $T\_L$)**

1. $LT \leftarrow 0$
2. $t \leftarrow 0$
3. $DT \leftarrow rand(0, T_L)$
4. **while** $t < T_L$
5. **if** AP is received, Update(AP.LID, AP)
6. **else if** $t = DT$
7. $lid \leftarrow$ Decision-Rule-I()
8. Update($lid$, NIL)

9. $forwarding[lid] \leftarrow true$

10. send AP with $LN \leftarrow CN[lid]$

11. **else if** $t = T_L$ goto Stage 2

**Algorithm Update(lid, AP)**

**At Sending Node** $s$

1. move all nodes in $DN[lid]$ to $CN[lid]$

2. $CN[lid] \leftarrow CN[lid] \cup my_U ID$

**At Receiving Node** $r$

1. **if** $N \leftarrow \{n \mid n \in DN[lid] and n \in DN[lid] \cap AP.LN\} \neq \emptyset$

2. move $N$ from $DN[lid]$ to $CN[lid]$

3. **else if** $CN[lid] \cap AP.LN \neq \emptyset$ **then** $CN[lid] \leftarrow CN[lid] \cup AP.UID$

4. **else** $DN[lid] \leftarrow DN[lid] \cup AP.UID$

**Algorithm Decision-Rule-I()**

1. $lid \leftarrow NIL$

2. $m \leftarrow 1$

3. **while** $lid = NIL$

4. **if** $DN[m] \neq \emptyset$, $lid \leftarrow m$

5. **else if** $CN[m] = \emptyset$, $lid \leftarrow m$

6. **else** $m \leftarrow m + 1$

7. return lid

Assume that a node $s$ decides its LID at DT. To decide its LID, $s$ scans its LT from the lowest layer. If node $s$ finds a non-empty $DN[m]$ for layer $m$, it elects itself as a connecting node for layer $m$. In this case, all nodes in $DN[m]$ are moved to $CN[m]$ including $s$ itself, and $s$ sends AP with $LN \leftarrow CN[m]$. If there is no disconnected node for layer $m$, $s$ looks for the $CN[m]$ whether it is empty or not. The empty $CN[m]$, together with the empty $DN[m]$, means that there is no node elected for layer $m$ currently. In this case, $s$ elects itself for layer $m$ as a dominating node. Otherwise, $s$ repeats the decision procedure for the next higher layer because the layer $m$ is already completed as a connected dominating set within the disk of radius $r_c$ centered at $s$, D(s). When node $r$ receives AP, it compares the $LN$ delivered in AP with its $DN$ indexed by the LID $m$ in AP. Node $r$ moves all matched entries in $DN(m)$ to $CN_u(m)$. If no match happens, only the UID in AP, which is the UID of the sending node, is added to $DN[m]$ as a disconnected layer-neighbor node.

The example of the PL procedure is depicted in Table II with relation to Fig. 21. Here, the DT is given to each node from $A$ to $F$, and hence, the decision procedure follows in the ascending order of UID. At time 1, node $A$ finds no previous layer formed by its neighbors and elects itself as layer 1 while sending its AP with no $LN_A(1)$. The receiving nodes $C, D, E, F$ add the UID $A$ to their $DN$ lists. Node $B$ will also find no previous layer, and hence, sends its AP with no $LN_B(1)$. Node $C$ finds a disconnected neighbor $A$ at its DT and sends AP with $LN_C(1) = A, C$. When node $A$ hears this message, it can move its own UID to its $CN_A(1)$. Node $C$ adds UID $A$ and $C$ to its $CN_C(1)$. Node $D$ will follow the same procedure as shown in Table II. In the case of node $E$, the LID will be chosen among the next higher layers since all $LN_E(1)$ are already connected. At last, node $F$ elects itself for layer

Table II. Primitive-Layering Procedure

| ***Node*** | CN/DN | $t = 1$ | $t = 2$ | $t = 3$ | $t = 4$ | $t = 5$ | $t = 6$ |
|---|---|---|---|---|---|---|---|
| A | CN | | | A,C | A,C,D | A,C,D | A,C,D,F |
| | DN | A | | | | | |
| B | CN | | | | | | B,F |
| | DN | | B | B | B | B | |
| C | CN | | | A,C | A,C | A,C | A,C |
| | DN | A | A | | | | |
| D | CN | | | | A,D | A,D | A,D |
| | DN | A | A | A | | | |
| E | CN | | | | A,D | A,D | A,D |
| | DN | A | A | A | | | |
| F | CN | | | | | | A,B,F |
| | DN | A | A,B | A,B | A,B | A,B | |

1 connecting the node $B$.

At the end of this stage, every node will have its one default LID and a corresponding $forwarding[lid]$ is set to $true$. This default LID is used as a $layer\_id$ that is included into a RREQ of a sending node.

Clearly, the computation cost of PL is upper bounded by $\Delta$ for decision making and $\Delta^2$ for receiving, where $H$ is the maximum number of layers constructed, and $\Delta$ is a maximum degree of a given graph $G$. When a node decides its LID, it needs to count the number of neighbors in each of $CN$ and $DN$ for constructed layers. For receiving, each node needs to compare its neighbors with LID $m$ and the listed nodes in a received $LN$ with LID $m$ of AP, which needs at most $O(\Delta^2)$ computation time. Hence, the total time complexity is $O(\Delta^2)$. The message overheads at each node is

$O(\Delta)$.

### 2. Stage 2. Inter-Layering (IL) Procedure

The Inter-Layering algorithm runs in the same way with PL except for the decision rule. By running the Inter-Layering (IL) algorithm, each node $v$ looks for a layer with a boundary condition and selects its second LID. The boundary refers to a border of a layer, and we can decide whether a node locates in the boundary of a certain layer by looking up the relationship between the disconnected neighbors and connected neighbors for each layer. When node $v$ finds that there are more than two disconnected neighbor nodes in a layer $m$, $v$ knows layer $m$ is disconnected locally within D(v). More conditions are explained later. Hence, in this case, $v$ needs to connect the disconnected neighbors by electing itself for layer $m$. Of course, there can be more than one layer that needs to be connected in D(v). However, the number of layers that have the boundary condition will be different at each node, so we can not decide the number of layering procedure at this moment. The rest of the layers with boundary condtions will be handled during the next stage.

**Stage 2. Algorithm Inter-Layering (LT, $T\_L$)**

1. $t \leftarrow 0$
2. $DT \leftarrow rand(0, T_L)$
3. **while** $t < T_L$
4. **if** AP arrives, Update(AP.LID, AP)
5. **else** if $t = DT$
6. lid $\leftarrow$ Decision-Rule-II()

7. **if** $lid > 0$
8. Update(lid, NIL)
9. $forwarding[lid] \leftarrow true$
10. send AP with $LN \leftarrow CN(lid)$
11. **if** $t = T_L$
12. run Boundary(0)
13. goto Stage 3

**Algorithm Decision-Rule-II()**

1. $m \leftarrow 1$
2. **while** $m \leq size(LT)$
3. **if** Boundary($m$)
4. return $m$
5. $m \leftarrow m + 1$
6. return 0

**Algorithm Boundary($m$)**

1. **if** $t < T_L$
2. **if** ($|DN[m]| > 1$) or ($|DN[m] = 1|$ and $|CN[m]| > 0$)
3. return $true$
4. **else**

5. **while** $m \leq size(LT)$

6. **if** $|DN_v(m)| > 0$

7. $boundary[m] \leftarrow true$

8. $m \leftarrow m + 1$

Based on $DN[m]$ and $CN[m]$, $v$ can decide whether a layer $m$ has a boundary condition or not. There are two boundary conditions: 1) $|DN[m] = 1|$ and $|CN[m]| > 0$ 2) $|DN[m]| > 1$. The former condition represents the existence of disconnected layer neighbors in that there are one disconnected node and one or more connected nodes. If there is a layer that satisfies the boundary condition while scanning from the lowest layer, a node will choose the layer as its second layer. The default LID and second LID are marked as true at the corresponding $forwarding[m]$ field.

To prepare for the next stage, each node performs a procedure of second boundary detection by running the procedure $Boundary(0)$ at the end of this stage. This procedure finds and marks all boundary conditions of $|DN_v(m)| > 0$. This condition includes all of the previous conditions, and also includes a case of isolated boundary, which means that there is one disconnected node in a layer $m$ and the node has no layer-neighbor node. A corresponding $boundary[m]$ is set to false for the layer with the boundary condition.

Phase 2 has a $O(\Delta)$ computation cost for both the decision making and boundary detection. The computation cost for receiving and message overhead are the same as those of Phase 1.

### 3. Stage 3: Runtime-Layering (RL) Procedure

The Runtime-Layeing procedure performs two functions in that it decides whether a node that receives a RREQ forwards the RREQ or not, and extends a layer if either the receiving node has either a *boundary* set to true or the LID of RREQ is greater than the maximum LID in the LT of the receiving node.

Actually, the MCDS protocol operates between the MAC layer and the routing layer in a communication protocol stack so that the routing layer can receive only filtered RREQ that have the same LID with the one obtained by the layering procedure. Therefore, the DSR protocol can work with a reduced number of RREQ with the MCDS protocol, which can reduce the computation overhead and resource requirement for the routing procedure.

When a RREQ with LID $m$ is received from node $s$, node $v$ first checks out whether the LID $m$ is greater than the maximum layer or not. If the LID of RREQ is greater than the maximum layer, which means that there is no node that can retransmit the RREQ within D(v) except $s$, $v$ runs the Decision-Rule-III to elect a next-hop node for node $s$ together with its neighbors that have also received the RREQ from $s$. Those neighbors will be the nodes that have its maximum layer less than $m$.

If LID $m$ of RREQ received is less than or equal to the maximum layer, a receiving node $v$ forwards the RREQ if its $forwarding[m]$ is true. If $forwarding[m]$ is set to false, $v$ checks its $boundary[m]$. If $boundary[m]$ is $false$, $v$ will discard the RREQ received because it is not in a corresponding layer $m$. Node $v$ will run Decision-Rule-III if its $boundary[m]$ is true since there is at least one disconnected node in layer $m$ within D(v). The nodes that have $boundary[m]$ set to false with $forwarding$ set to false, as in $v$, will also run Decision-Rule-III to elect next forwarding node.

The Decision-Rule-III works as follows. If a RREQ with LID $m$, which incurrs one of the two above situations, is received for the first time at a group of receiving nodes $P$, the nodes in $P$ store the Request ID (RID) of the RREQ and forward the RREQ without any decision at this moment. Note that only one node among the nodes in $P$ can send RREQ at any one time. Here, assume that node $v$ is the first forwarding node among the nodes in $P$. When $v$ sends the RREQ, another group of nodes $N$ within the transmission range of $v$ can receive the RREQ sent by $v$. As explained in Section C, nodes in $N$ will proceed with the first copy of RREQ from $v$, while discarding all next copies of RREQ from nodes in $P$. When the nodes in $N$ forward the RREQ to next-hop nodes, the nodes in $P$ can also hear this message. By hearing the RREQ sent by its next-hop, any node in $P$ checks a field of UID of previous hop $prev\_src$ in the header of RREQ, which was included by any sending node with its previous hop node. In this case, the previous hop node in $prev\_src$ is the UID of node $v$. This result indicates that node $v$ is selected among all nodes in $P$ that received the RREQ from nodes in $P$. As a result, $v$ sets $forwarding[m]$ to true and $boundary[m]$ to false because the boundary condition in layer $m$ is currently dissolved. The other nodes in $P$, however, will wait to check if any node in their next-hop may respond to it with RREQ having its UID in $prev\_src$ field. Meanwhile, if a new RREQ with a different RID from the previous one it sent is received, it sets both $forwading[m]$ and $boundary[m]$ to false because it could not find its next-hop that responded to it. As the RREQ with LID $m$ is propagested through the network, the layer $m$ will be extended through the network by newly elected nodes as explained above.

As you can see, the algorithm Decision-Rule-III runs in $O(\Delta)$ time for the worst case since the frequency of running the Decision-Rule-III is bounded by the maximum number of layers in a given network, which is bounded by the maximum degree of

the network, $\Delta$. Note that the layer extension is a one-time decision procedure for any disconnected layer. The RL procedure has no message overhead since the procedure uses a normal route request (RREQ) packet. Therefore, the total computation cost to complete the multiple-CDSs (MCDSs) construction is $O(\Delta^2)$ and the overall message overhead is $O(\Delta)$. In addition, the RL procedure for layer extension needs a field of *prev_src* in RREQ packet to identify the UID of 1-hop previous node. Any forwarding node will include the unique id (UID) of its 1-hop previous node under the MCDS protocol.

**Stage 3. Algorithm Runtime-Layering (LT, RREQ)**

1. $lid \leftarrow LID(RREQ)$
2. **if** $lid > size(LT)$
3. **if** Decision-Rule-III(LT, RREQ)
4. forward RREQ
5. **else**
6. discard RREQ
7. **else**
8. **if** $boundary[lid] = false$
9. **if** $forwarding[lid]$
10. forward RREQ
11. **else** $forwarding[lid]$

12. discard RREQ
13. **else**
14. **if** Decision-Rule-III(LT, RREQ)
15. forward RREQ
16. **else**
17. discard RREQ

**Algorithm Decision-Rule-III(LT, RREQ)**

1. $lid \leftarrow LID(RREQ)$
2. **if** the first time receiving of RREQ
3. $rid[lid] \leftarrow RID(RREQ)$
4. return true
5. **else if** $rid[lid] = RID(RREQ)$
6. if $prev\_src(RREQ) = my\_UID$
7. $boundary[lid] \leftarrow false$
8. $forward[lid] \leftarrow true$
9. return true
10. **else**
11. $boundary[lid] \leftarrow false$
12. $forward[lid] \leftarrow false$
13. return false

### 4. Fault Tolerance

We build a fault tolerance feature using the RL procedure. Any fault can be detected either by a periodic beacon message or by a feature of bi-directional communication. For the first case, if a node finds no reply to its beacon message from a certain layer neighbor node, it sends a Re-Layering Request packet. For the second case, a data packet can be used. When there is no acknowledgement from its layer neighbor, which is known during the route discovery procedure, the detecting node also sends a Re-Layering Request packet. All neighbor nodes that receive the Re-Layering Request set their *boundary* to true and will run the Decision-Rule-III whenever a RREQ packet arrives for the same layer.

## E. Simulation Environment and Results

### 1. Simulation Setup

In order to evaluate the features on load-balancing and network performance of the proposed MCDS protocol, we run simulation using the ns-2 [46] simulator. We implemented our MCDS protocol and Gossip-based routing protocol [39] based on basic DSR routing protocol [17]. The Gossip protocol is selected as a load balancing scheme for on-demand routing protocols in that network traffic can be distributed with a certain probability. To implement the Gossip protocol over the DSR routing protocol, we applied the GOSSIP3(.65,4,1) scheme in which a node broadcasts a route request packet (RREQ) if it locates within 4 hops from a source node or, if not, it forwards a RREQ with a probability of 65% at the first reception of the RREQ. If it initially got the RREQ and did not forward the RREQ but it did not get the same RREQ from at least one other node during a predefined timeout interval. We set the NODE_TRAVERSAL_TIME $40ms$. The parameters of GOSSIP3 used in our simu-

lation are chosen for the node degree of 10, which are used in [39]. In our simulation environment, we have the same situation when the transmission range is $50m$.

Table III shows the configuration parameters used in the simulation. Over a two-dimensional plane with size of $400m \times 400m$, we randomly generate 10 different network topologies, each of which is composed of randomly placed 350 sensor nodes. Based on the generated topologies, we vary the transmission range $r_c$ of sensor node among 40, 50, 60, and 70m to simulate different node degrees of the network. Each data value in our results represents an average of 100 runs per each transmission range using the 10 different topologies. Network traffic is generated by any node that locates within a sensing range of $40m$ from two random event points that placed in the opposite side of one destination node. Each event point constantly generates one event per second. The MAC layer scheme is the IEEE 802.11 MAC protocol. In this study, we assume a stationary network where the location of nodes is fixed at the original position. Each node is given with initial energy $50J$. In our simulation, we set 100 seconds for the layering procedure. Simulation time is set to $3,000$ seconds. During the simulation time, we sample the data on current energy and network traffic of each node at every 100 seconds. The original DSR protocol, MCDS protocol, and Gossip protocol are represented as DSR, MCDS, and Gossip, respectively.

## 2. Simulation Results

### a. Properties of MCDS Structure

Fig. 23 shows the number of layers constructed while varying the transmission range. The average number of layers constructed increases linearly as the transmission range is extended, which proves that the dominating-and-connecting (DAC) method fairly constructs each layer with a similar number of nodes even though the density of nodes

Table III. Simulation Environments and Scenarios

| ***Parameter*** | *Value* |
|---|---|
| Simulator | ns-2 (version 2.29) |
| Network Size | $400m \times 400m$ |
| Number of Nodes | 350 |
| Comm. Range | $40m, 50m, 60m, 70m$ |
| Sensing Range | $40m$ |
| Initial Energy | $50J$ |
| Simulation Time | $3000s$ |
| MAC | IEEE 802.11 |
| Data Packet Size | $512bytes$ |
| Bandwidth | $2Mbps$ |

in the transmission range increases. We will show the number of nodes in each layer later.

The difference between maximum and minimum number of layers indicates that the number of constructed layers varies over the network area. This is not surprising, since the number of nodes at the local region in the network is unlike the density of nodes. We will show that the Runtime-Layering procedure can guarantee a high probability of packet delivery through the heterogeneous levels of layers by extending each disconnected layer over a network-wide area.

In Fig. 24, we show the number of elected nodes in each layer. When the transmission range is $40m$, we can see a great difference among the constructed layers. Because of a relatively small transmission range, most of the nodes are elected for the first layer. However, the difference becomes moderate as the transmission range is increased, which means that the distributed sensor nodes can be elected uniformly

over the constructed layers if the density of nodes is high enough. In other words, this result shows that the DAC method can select a fairly constant number of nodes per each layer. Our simulation shows that the average degree of nodes in each layer is around 6. Based on the average degree, we can construct the MCDS that has a uniform distribution of elected nodes in each layer by spreading a proper number of nodes for a certain number of layers.

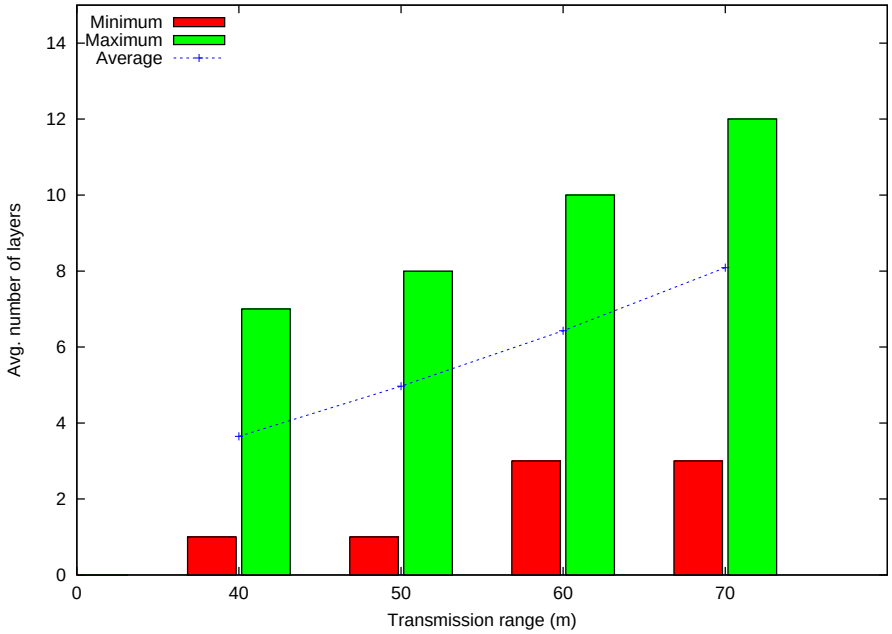



Fig. 23. Maximum, Minimum, and Average Number of Layers Constructed via DAC

b. Effect on Load Balancing

In the following two sections, we evaluate the network performance of the DSR with MCDS protocol (MCDS), DSR without MCDS (DSR), and DSR with Gossip (Gossip) regarding the load balancing and energy consumption. First, the effect of load balancing is observed by counting the number of nodes that depleted its whole given energy budget during network operation. If a node is highly burdened with a net-

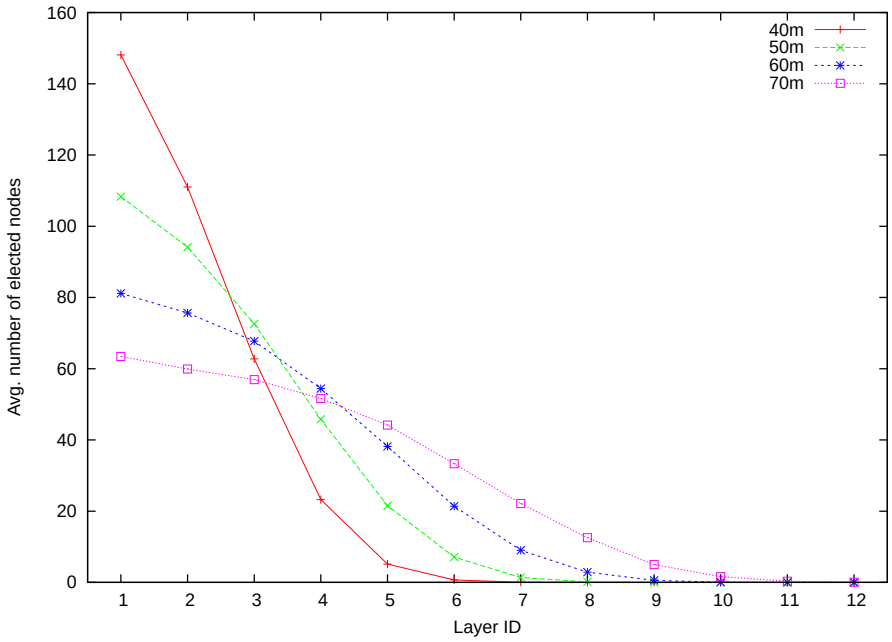



Fig. 24. Average Number of Elected Nodes in Each Layer via DAC

work traffic, the node will deplete its energy earlier than others. Figs. 25-28 show the average number of energy-depleted nodes sampled at every 100 seconds during the simulation time. As shown in the figures, the average number of energy-depleted nodes increases at any time instance as the transmission range becomes longer. This is because each node consumes more transmission energy when the transmission range is extended. By observing the number of energy-depleted nodes in each case, we can see that the MCDS case has more alive nodes compared to both the DSR case and the Gossip case during the whole simulation time. In addition, the time instance when the first node depletes its whole energy is shown last in the MCDS case across all the transmission ranges. This result proves that the MCDS protocol can distribute the network traffic more evenly than both the original DSR protocol and the Gossip protocol. If we refer to the delivery ratio shown in Fig. 37, all three different protocols show a similar packet delivery ratio during the simulation time.

Meanwhile, Fig. 25 shows that the MCDS protocol has more balanced network traffic in that the MCDS case shows the smallest number of energy-depleted nodes among all the compared protocols. The effect of the load balancing in MCDS is maximized against other protocols when the transmission range is relatively long since the network traffic could be distributed over more constructed layers. The saturated region of each graph indicates that there is no more packet delivery caused by network disconnection. We also found that the Gossip protocol could reduce the energy burden of each node only for a small period of network operation during which most nodes are alive. However, the network traffic of RREQ is increased greatly as shown in Fig. 35, which in turn increased the overall energy burden in each node. In the Gossip protocol, the RREQ packet is delayed in some fraction of nodes with a given forwarding probability during which a source node may send another RREQ packet. Hence, the overall packet overhead can be increased as the number of failed nodes increases unless the forwarding probability is dynamically changed according to the fraction of node failure. From this simulation, we can conclude that the MCDS protocol effectively balances the network traffic for the on-demand routing protocols among the compared protocols. In addition, the effect will be enhanced as the density of nodes increases.

In Figs. 29-32, we show the average residual energy of the entire network. We can see that the MCDS case conserves more energy than both the DSR and Gossip cases. Also, the difference in the amount of the residual energy gets bigger as the transmission range increases. This result shows that the MCDS protocol consumes less energy even though the amount of network traffic is similar among the three protocols. The effect of energy savings is the result of the reduced number of forwarding nodes. In the MCDS protocol, only the nodes in the same layer forward RREQ packets, which conserves a residual energy in the nodes in a different layer. On the other hand, most

of the nodes participated in broadcasting RREQ and wasted their residual energy in both the DSR and Gossip cases. The Gossip protocol has shown the largest energy consumption because the frequency of RREQ was increased, as shown in Fig. 35.

c. Effect on Network Performance

In order to investigate the effect on the packet propagation of the MCDS protocol, four metrics are studied. First, we compare the end-to-end delay of the three protocols. The end-to-end delay is a packet delivery time including all possible delays related to the route discovery, buffering in intermediate nodes, and propagation time between source and destination node. As shown in Fig. 33, the Gossip case shows the largest end-to-end delay since many nodes keep the route discovery packet before forwarding until the predefined timeout period. The MCDS case has shown a slightly increased delay against the DSR case for 40 and $50m$ since the route discovered has more hops than that of the DSR case. However, the number of hop count is reduced when the transmission range is extended. This is because the MCDS case has a reduced number of neighbors compared to the DSR case as the transmission range increases.

The second metric is the hop count of discovered route from source to destination. As shown in Fig. 34, the average hop counts are similar in all three protocols, and decrease as the transmission range is reduced. Specifically, the MCDS case shows the smallest number of hop count under a relatively large transmission range. With a large communication range, the MCDS case has a relatively less number of neighbors compared to other protocols and the overall hop counts are also reduced.

Third, we discuss the message overhead related to the number of route discovery packets and the number of data packets dropped in the MAC layer because of collisions. In Figs. 35-36, the normalized number of RREQ packets and data packets dropped to those of the DSR case are represented. The MCDS case used only 71%

RREQ packets with a $40m$ transmission range and about 40% with the higher transmission ranges compared to the DSR case during whole simulation time. However, the Gossip case forwarded more RREQ packets than the DSR and MCDS cases. This overhead directly increased the energy consumptions at each node as previously discussed. The reason for the reduced number of RREQ packets in the MCDS case is straight-forward since the forwarding of RREQ packets is restricted to only the same layer with the source node.

Meanwhile, the number of data packets dropped is increased in the MCDS and Gossip cases against the DSR case. In the MCDS case, the collision can happen more frequently than the DSR case because multiple close paths in MCDS may cause the collision with a higher probability than a single path in DSR. To solve this problem, we plan to implement a smart scheduler for the MAC layer with the MCDS protocol. The Gossip case also incurred a high ratio of packet drops against the DSR case. This is also caused by the multiple paths that are used simultaneously.

Lastly, we discuss the packet delivery ratio of each protocol. In Figs. 37-40, we draw the results of average delivery ratio varying the transmission range. As shown in Fig. 37, all protocols show a similar packet delivery ratio. The MCDS case has not improved the delivery ratio much, but has reserved more energy as previously shown. However, the delivery ratio of MCDS is much higher than any other protocols when the transmission range is increased. From the results, we can see that the MCDS case has a higher network throughput up to 35% against DSR and 40% against Gossip. Hence, we can also expect a prolonged network lifetime with the MCDS case if the same amount of network traffic is propagated. More specifically, we can conclude that the load balancing and energy efficiency with the MCDS protocol can be maximized as the density of nodes is increased. The unique and novel feature of the MCDS protocol regarding the load balancing and energy efficiency resides in this point. The

Figs 25-40 are shown at the end of this chapter.

## F. Conclusion

In this study, we proposed a topology management protocol, Virtual Layer Topology, that provides the load balancing for on-demand routing protocols. Our greedy Dominating-and-Connecting (DAC) method could construct multiple connected dominating sets (CDSs) in $O(\Delta^2)$ time. With the MCDS protocol, all nodes are distributed evenly over the multiple layers, each of which corresponds to one CDS, and the load balancing and energy efficiency of WSN is greatly improved. However, we also found that the number of packet drops was increased for the multiple paths that are used simultaneously. For future work, we will enhance the MCDS topology implementing a more efficient fault tolerating scheme that can reconfigure MCDS to increase the connectivity under a high rate of node failure. We will also study a methodology with which the MCDS protocol can be used under the presence of node mobility. The necessity of a smart scheduler that can send packets through different layers without having a high collision ratio is emerged through this work as another important future work.

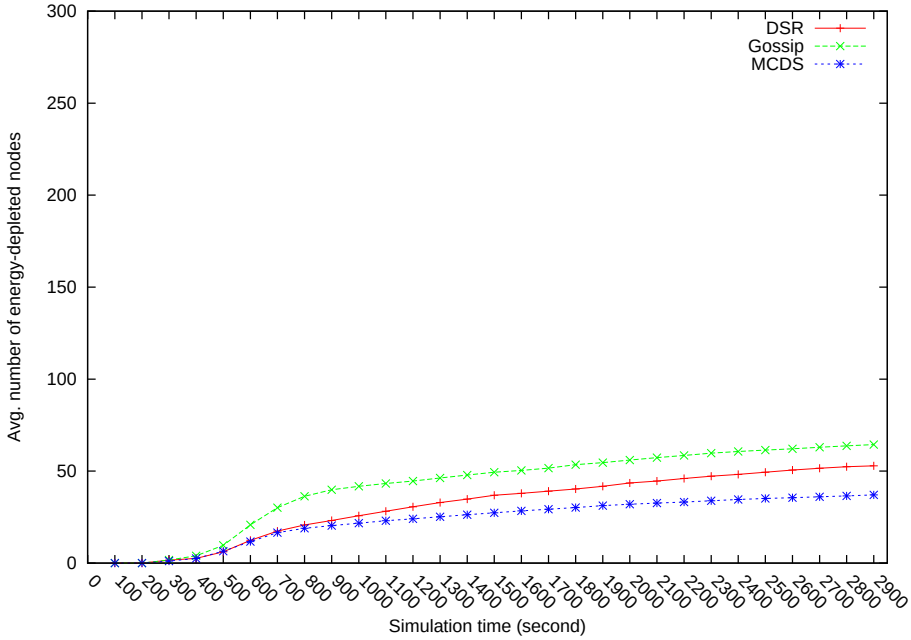



Fig. 25. Number of Energy-depleted Nodes with 40m Transmission Range

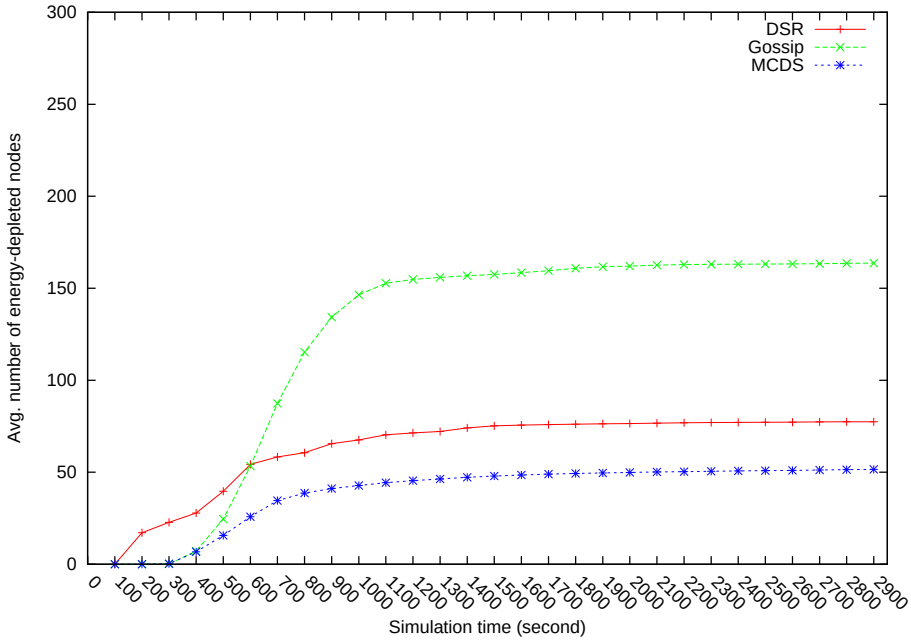



Fig. 26. Number of Energy-depleted Nodes with 50m Transmission Range

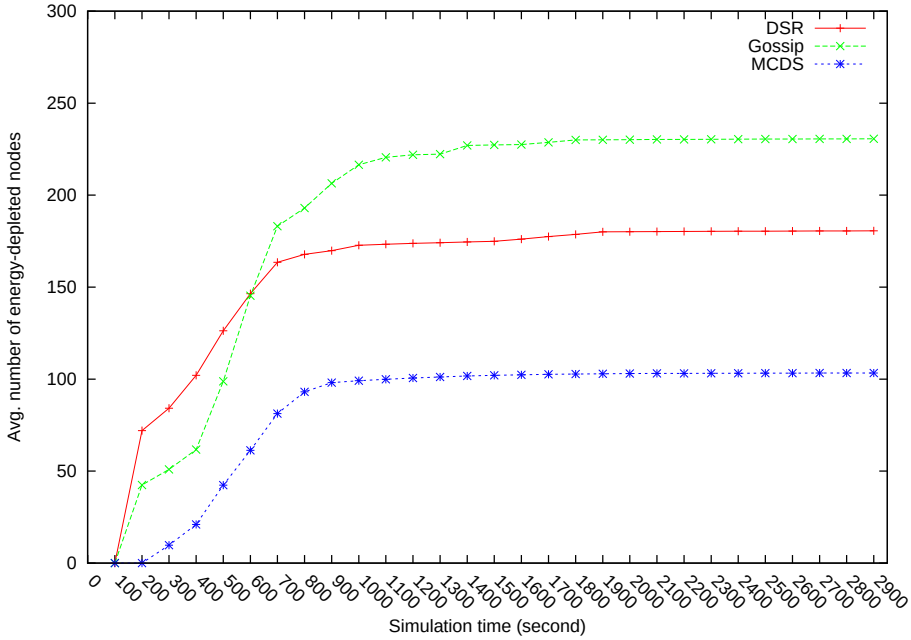



Fig. 27. Number of Energy-depleted Nodes with 60m Transmission Range

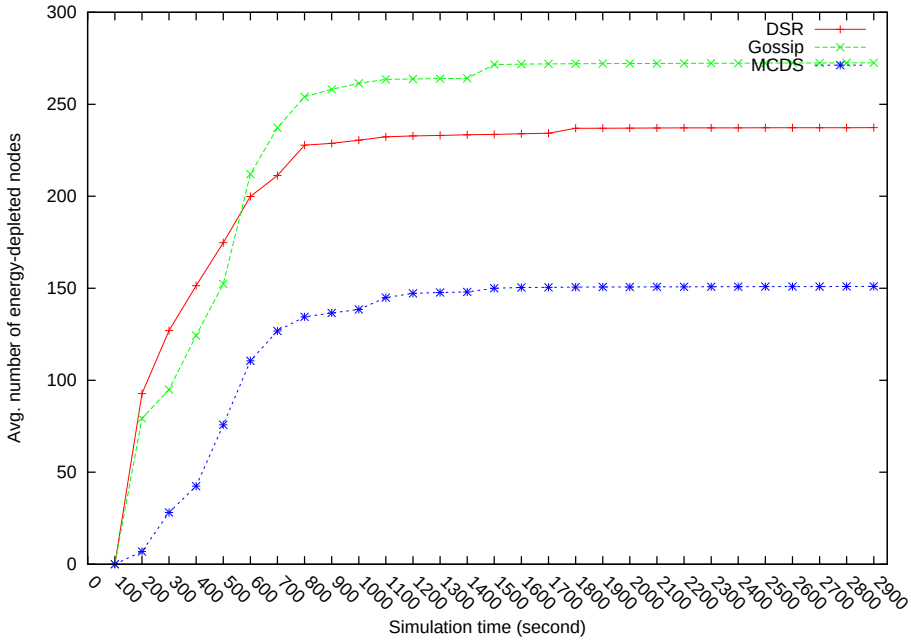



Fig. 28. Number of Energy-depleted Nodes with 70m Transmission Range

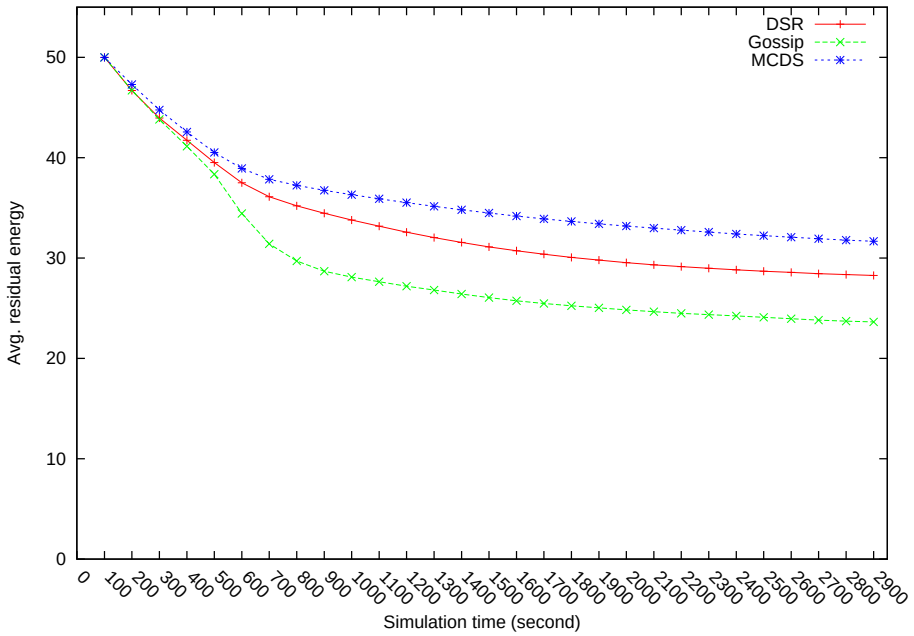



Fig. 29. Residual Energy with Transmission Range of 40m

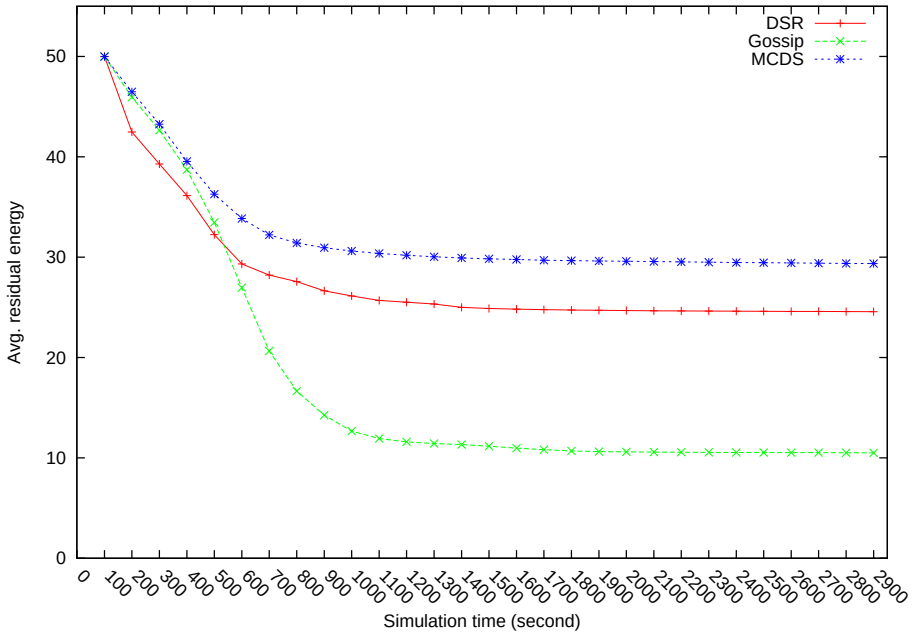



Fig. 30. Residual Energy with Transmission Range of 50m

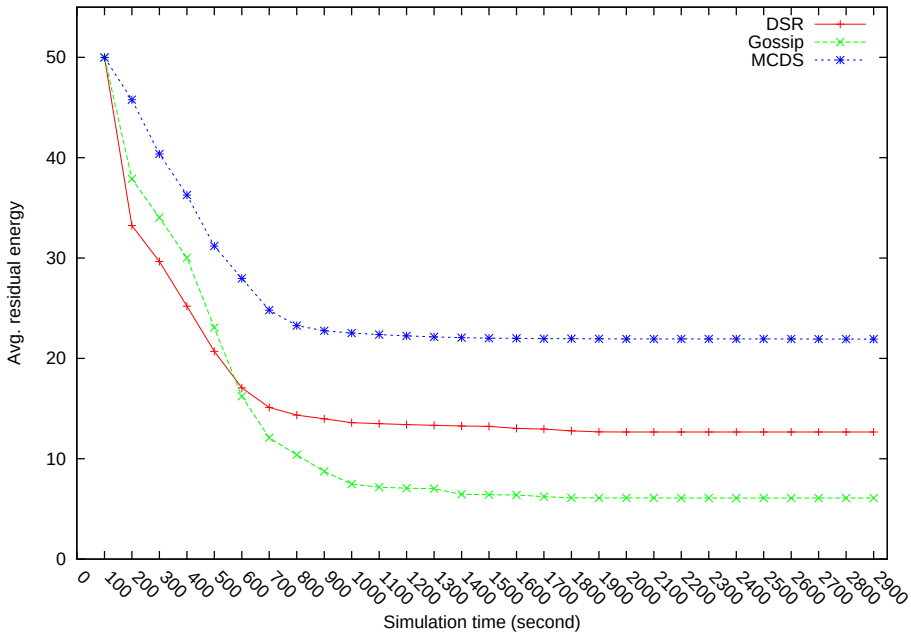



Fig. 31. Residual Energy with Transmission Range of 60m

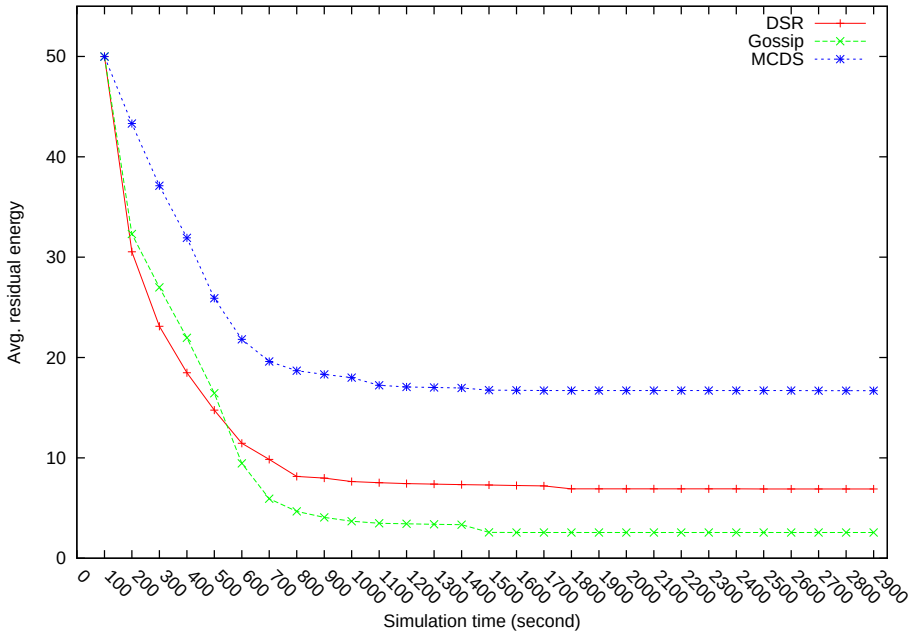



Fig. 32. Residual Energy with Transmission Range of 70m

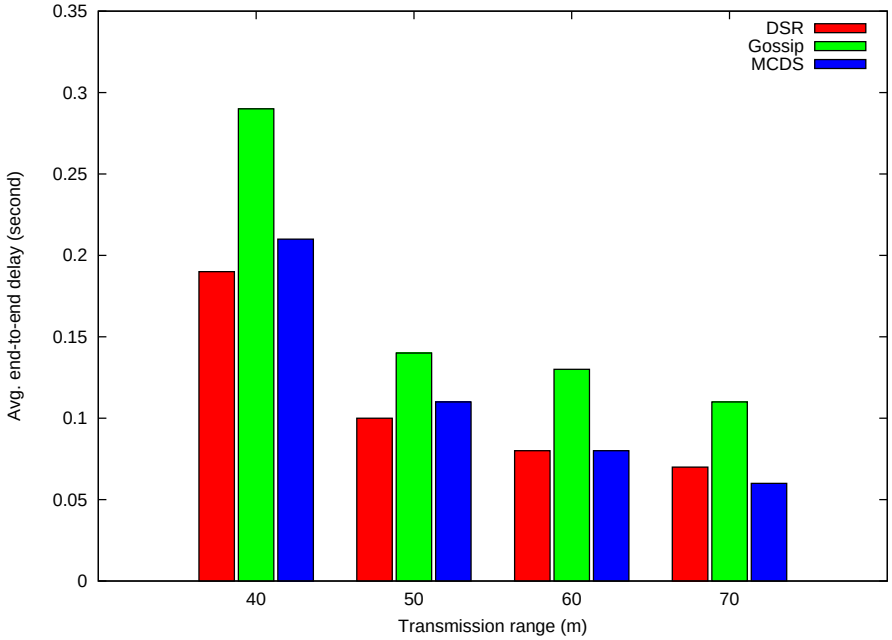



Fig. 33. Delay

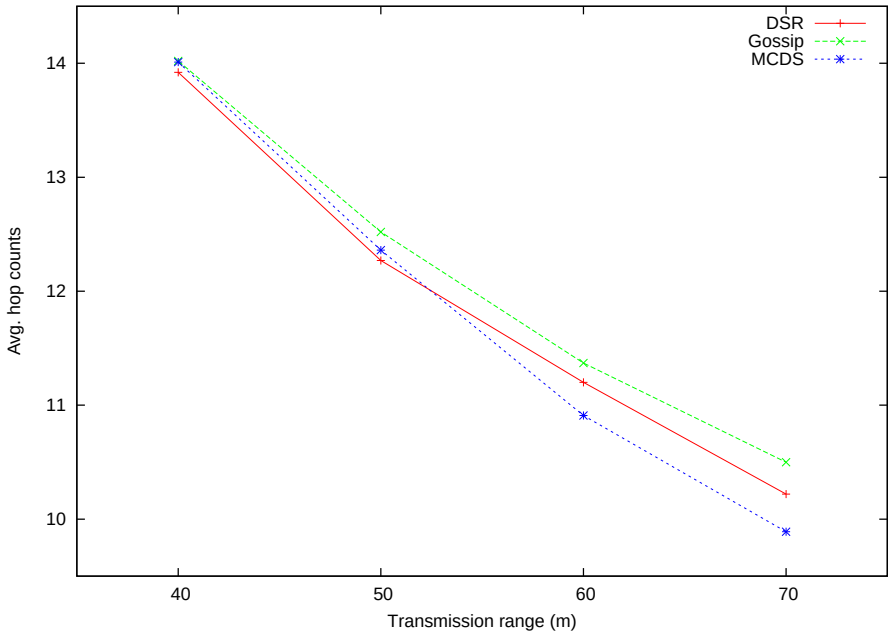



Fig. 34. Average Number of Hop Counts

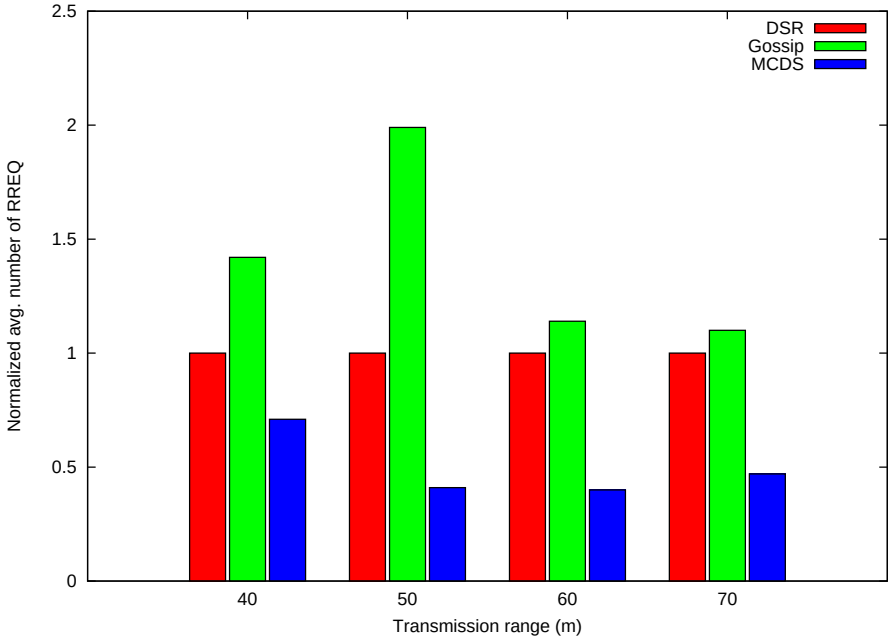



Fig. 35. Overhead of Route Request (RREQ) Packets

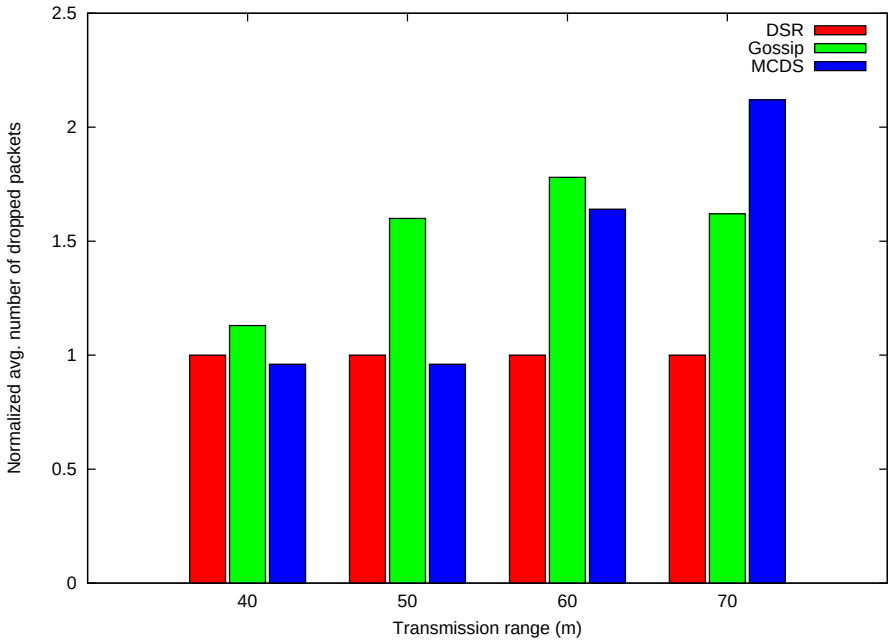



Fig. 36. Overhead of Data Packets

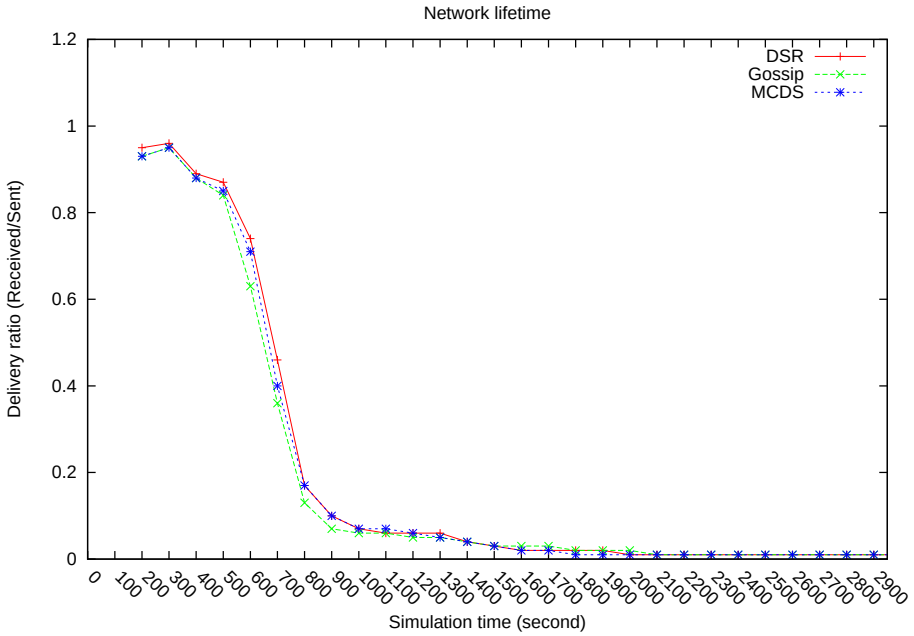



Fig. 37. Delivery Ratio with Transmission Range of 40m

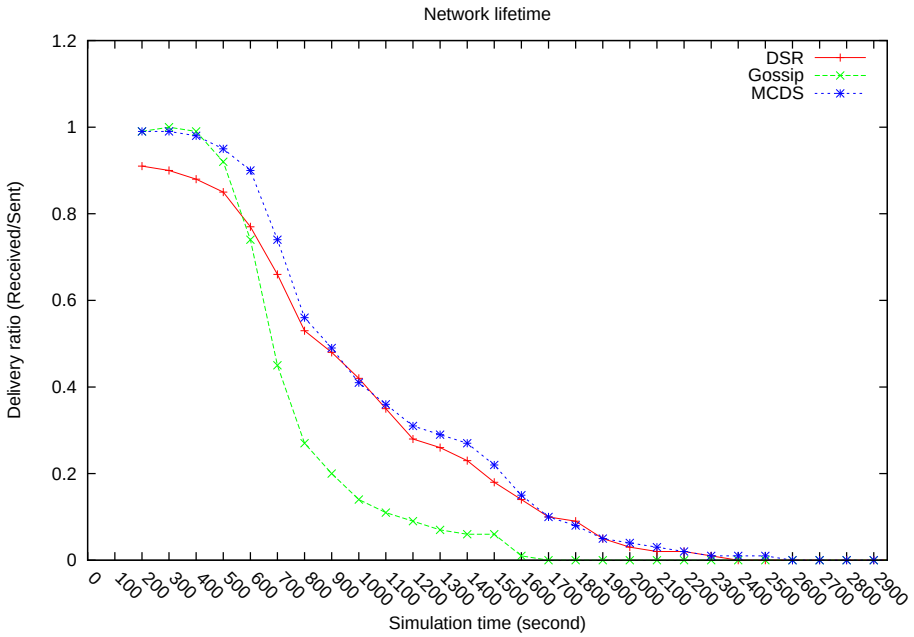



Fig. 38. Delivery Ratio with Transmission Range of 50m

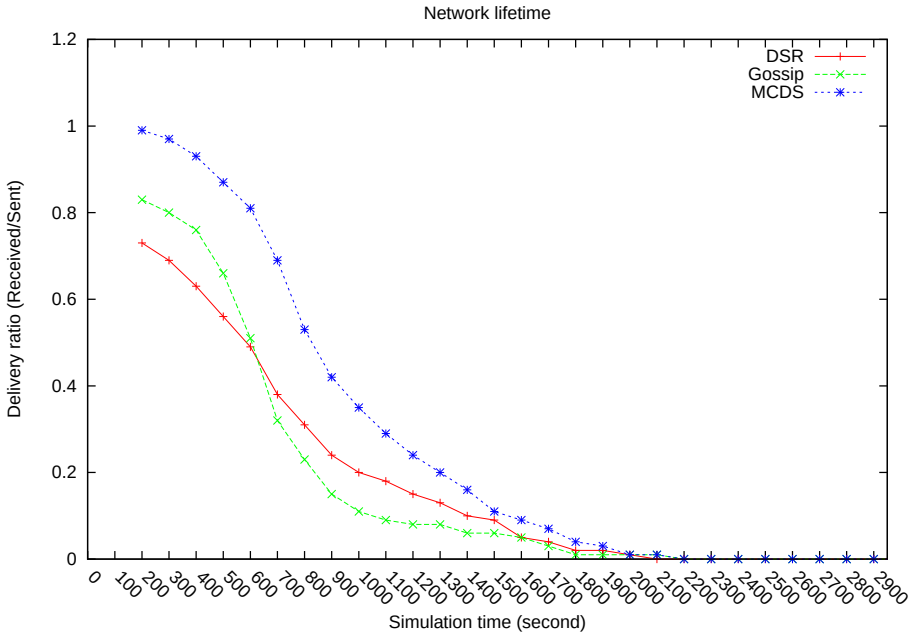



Fig. 39. Delivery Ratio with Transmission Range of 60m

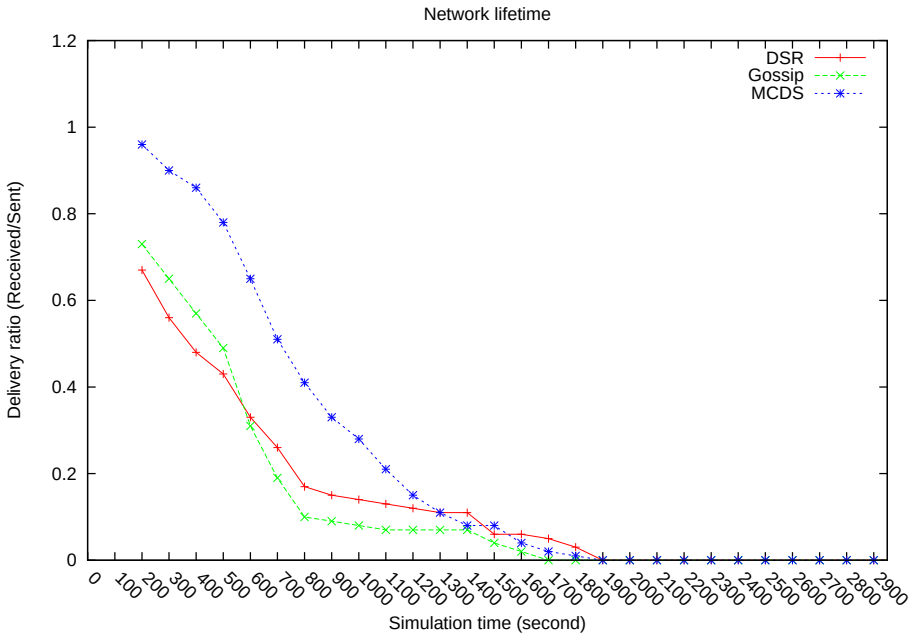



Fig. 40. Delivery Ratio with Transmission Range of 70m

# CHAPTER VI

## CONCLUSION

We have presented the energy-efficient topology management protocols that are implemented through four different protocols with regard to the $K$-coverage and the load balancing problems in Wireless Sensor Networks (WSNs).

To solve the $K$-coverage problem, we first proposed the Randomly Ordered Activation (ROAL) protocol. The ROAL protocol introduced a new model of layer coverage that can solve the $K$-coverage problem in $O(K)$, where $K$ is the coverage degree, without using the geographic coordinates. The layer coverage also provided a dynamic reconfiguration scheme that can evenly balance the energy consumption over all distributed nodes. We have shown that the ROAL protocol can achieve almost the same quality of coverage with the strategy that uses the geographic coordinates with a significantly reduced algorithm running time. We extended the ROAL protocol with an enhanced fault tolerant method in the Circulation-ROAL (C-ROAL) protocol. In this work, we also presented the mathematical analysis on the coverage and connectivity of the layer coverage scheme. The C-ROAL protocol greatly reduced the energy consumption spent during a repetitive layering procedure while guaranteeing the coverage. The unique feature of the circulation of layers is placed on the low cost fault tolerance. The autonomous and sequential activation using the circulation method can reconfigure the network topology providing the energy-balancing with no redundant energy consumption.

We proposed the Mobility Resilient Coverage Control (MRCC) protocol to ensure $K$-coverage in the presence of mobile sensors, along with the wear-out failures of nodes. The MRCC protocol has shown that it can reduce the number of working nodes with the account of moving-in and moving-out probability. Also, we applied

the wear-out failure to replace any working node before an expected reliable working duration elapses. This approach proved that the $K$-coverage can be satisfied with an acceptable probability using a less number of working nodes.

Lastly, we presented a load balancing protocol, called multiple-CDS Topology (MCDS), working with the Dynamic Source Routing (DSR) protocol. For this work, we defined a new concept of the Connected Dominating Set (CDS) and constructed the MCDS using the CDS concept using a Dominating-and-Connecting (DAC) heuristic. The MCDS protocol proved that the network traffic can be balanced over the sensor nodes by the multiple CDSs structure. Moreover, we have shown that the overall energy properties and network performance has been improved via the MCDS protocol.

# REFERENCES

[1] Deborah Estrin, David Culler, Kris Pister, and Gaurav Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.

[2] B. Warneke, M. Last, B. Liebowitz, and K.S.J. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Computer*, vol. 34, pp. 44–51, 2001.

[3] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Proc. 2nd International Workshop on Information Processing in Sensor Networks (IPSN03)*, Palo Alto, California, USA, April 2003, pp. 333–348.

[4] A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proc. 9th Annual International Conf. on Mobile Computing and Networking*, San Diego, California, USA, September 2003, pp. 96–108.

[5] V. Ponduru, B. Mukherjee, and D. Ghosal, "A distributed coverage-preserving multipath routing protocol in wireless sensor networks," in *Technical Report, Department of Computer Science, University of California*, 2004.

[6] B. Karp and H. Kung, "Greedy perimeter stateless routing for wireless networks," in *Proc. the ACM/IEEE International Conf. Mobile Computing and Networking*, Boston, Massachusetts, August 2000, pp. 243–254.

[7] Q. Fang, J. Gao, L. J. Guibas, V. de Silva, and L. Zhang, "GLIDER: Gradient landmark-based distributed routing for sensor networks," in *Proc. INFOCOM*

*2005. 24th Annual Conf. IEEE Computer and Communications Societies*, Miami, USA, March 2005, vol. 1, pp. 339–350.

[8] Jehoshua Bruck, Jie Gao, and Anxiao Jiang, "MAP: Medial axis based geometric routing in sensor networks," in *Proc. MOBICOM*, Cologne, Germany, August 2005, pp. 88–102.

[9] Chi-Fu Huang and Yu-Chee Tseng, "The coverage problem in a wireless sensor network," in *Proc. 2nd ACM International Conf. Wireless Sensor Networks and Applications*, San Diego, CA, USA, September 2003, pp. 115–121.

[10] F. Ye, G. Zhong, S. Lu, and L. Zhang, "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *Proc. 23rd International Conf. Distributed Computing Systems (ICDCS)*, Providence, Rhode Island, USA, May 2003, pp. 28–37.

[11] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," in *Proc. 1st International Conf. Embedded Networked Sensor Systems*, Los Angeles, California, USA, October 2003, pp. 28–39.

[12] Santosh Kumar, Ten H. Lai, and Anish Arora, "Barrier coverage with wireless sensors," in *Proc. International Conf. Mobile Computing and Networking*, Cologne, Germany, September 2005, pp. 284–298.

[13] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *Proc. International Conf. Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 139–150.

[14] S. Kumar, T.H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *Proc. 10th Annual International conf. Mobile Computing and Networking*, Philadelphia, PA, USA, September 2004, pp. 144–158.

[15] Zoe Abrams, Ashish Goel, and Serge Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," in *Proc. 3rd International Workshop on Information Processing in Sensor Networks*, Berkeley, California, USA, April 2004, pp. 424–432.

[16] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, and L. Gu, "Energy-efficient surveillance system using wireless sensor networks," in *Proc. 2nd International Conf. Mobile Systems, Applications, and Services*, Boston, MA, USA, June 2004, pp. 270–283.

[17] David B. Johnson, David A. Maltz, and Josh Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," Available: http://www.monarch.cs.cmu.edu/monarch-papers/dsr-chapter00.pdf, 5/3/2005.

[18] Hogil Kim and Eun Jung Kim and Ki Hwan Yum, "ROAL: A randomly ordered activation and layering protocol for ensuring k-coverage in wireless sensor networks," in *Proc. 3rd International Conf. Wireless and Mobile Communications*, Guadeloupe, French Caribbean, March 2007, p. 4.

[19] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. IEEE International Conf. on Communications*, Helsinki, Finland, June 2001, pp. 472–476.

[20] Douglas M. Blough and Paolo Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad

hoc networks," in *Proc. MOBICOM*, Atlanta, Georgia, USA, September 2002, pp. 183–192.

[21] Ning Li, J.C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," *IEEE Trans. Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, 2005.

[22] Benyuan Liu, Peter Brass, Olivier Dousse, Philippe Nain, and Don Towsley, "Mobility improves coverage of sensor networks," in *Proc. 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Urbana-Champaign, IL, USA, May 2005, pp. 300–308.

[23] Michael McGuire, "Stationary distributions of random walk mobility models for wireless ad hoc networks," in *Proc. 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Urbana-Champaign, IL, USA, May 2005, pp. 90–98.

[24] Christian Bettstetter, "Mobility modeling in wireless networks: Categorization, smooth movement, and border effects," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 3, pp. 55–66, 2001.

[25] Christian Bettstetter, Giovanni Resta, and Paolo Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 2, no. 3, pp. 257–269, 2003.

[26] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 481–494, 2002.

[27] Daehyoung Hong and S. S. Rappaport, "Traffic model and performance analysis

for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures," *IEEE Trans. Vehicular Technology*, vol. 35, no. 3, pp. 77–92, 1986.

[28] Christian Bettstetter, Hannes Hartenstein, and Xavier Pérez-Costa, "Stochastic properties of the random waypoint mobility model," *Wirel. Netw.*, vol. 10, no. 5, pp. 555–567, 2004.

[29] Jungkeun Yoon, Mingyan Liu, and Brian Noble, "Random waypoint considered harmful," in *Proc. INFOCOM 2003*, San Francisco, California, USA, March 2003, vol. 2, pp. 1312–1321.

[30] Guiling Wang, Guohong Cao, and T. La Porta, "Movement-assisted sensor deployment," in *Proc. INFOCOM 2004*, March 2004, vol. 4, pp. 2469–2479.

[31] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[32] G. Lin, G. Noubir, and R. Rajamaran, "Mobility models for ad-hoc network simulation," in *Proc. INFOCOM 2004*, Hong Kong, China, March 2004, vol. 1, pp. 7–11.

[33] Guoliang Xing, Xiaorui Wang, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill, "Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks," *ACM Trans. Sensor Networks*, vol. 1, no. 1, pp. 36–72, 2005.

[34] *NIST/SEMATECH e-Handbook of Statistical Methods*, Available: http://www.itl.nist.gov/div898/handbook/apr/apr.htm, 10/12/2005.

[35] Igor Bazovsky, *Reliability Theory and Practice*, Dover Publications, Mineola, NY, USA, 2004.

[36] E. M. Royer and C. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," vol. 6, no. 2, pp. 46–55, April 1999.

[37] Mahesh K. Marina and Samir R. Das, "Ad hoc on-demand multipath distance vector routing," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, pp. 92–93, 2002.

[38] C. E. Perkins and E. M. Royer and S. R. Das, "Ad hoc on demand distance vector routing," in *Proc. 2nd Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, USA, February 1999, pp. 90–100.

[39] Zygmunt J. Haas, Joseph Y. Halpern, and Erran L. Li, "Gossip-based ad hoc routing," in *Proc. INFOCOM*, New York, USA, June 2002, pp. 1707–1716.

[40] Jie Wu and Hailan Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *DIALM '99: Proc. 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Seattle, Washington, USA, August 1999, pp. 7–14.

[41] T.W. Haynes F. Harary, "Double domination in graphs," *ARS Combinatoria*, vol. 55, pp. 201–213, 2000.

[42] Fei Dai and Jie Wu, "On constructing k-connected k-dominating set in wireless networks," *Parallel and Distributed Computing*, vol. 66, no. 7, pp. 947–958, 2006.

[43] C. J. Colbourn B. N. Clark and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.

[44] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proc. INFOCOM*, New York, USA, June 2002, pp. 141–149.

[45] J. Wu and H. Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *Telecommunication Systems*, vol. 18, no. 1-3, pp. 13–36, 2001.

[46] VINT Project, "The UCB/LBNL/VINT network simulator-ns (Version 2)," Available: http://www.isi.edu/nsnam/ns, 10/12/2004.

VITA

**Ho Gil Kim**

## Education

- Master of Engineering, Electrical Engineering, Younsei University, South Korea, 1996
- Bachelor of Engineering, Electrical Engineering, Korea Military Academy, South Korea, 1992

## Contact Address


- *Permanent mailing address* : Department of Computer Science, Texas A&M University, TAMU 3112, College Station, TX, 77843
- *E-mail* : hogil@cs.tamu.edu, khglex@gmail.com