Resilient Network Control based on Stochastic Mathematical Model of Biological Behavior for Wireless Sensor and Actuator Networks

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2015

Takuya IWAI

List of Publications

Journal Papers

1. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Error-tolerant and energy-efficient coverage control based on biological attractor selection model in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, February 2012.

2. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Response threshold model-based device assignment for cooperative resource sharing in a WSAN,” *International Journal of Swarm Intelligence and Evolutionary Computation*, April 2012.

Refereed Conference Papers

1. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Bio-inspired Autonomous and Adaptive Coverage Control for Wireless Sensor Networks,” in *Proceedings of the 2nd International Workshop on Sensor Networks and Ambient Intelligence*, September 2009.

2. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Error-tolerant coverage control based on bio-inspired attractor selection model for wireless sensor networks,” in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology*, June 2010.

3. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Proposal for dynamic organization of service networks over a wireless sensor and actuator network,” in *Proceedings of the 2nd International Conference on Ambient Systems, Networks and Technologies*, September 2011.

4. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Analysis of response threshold model and its application for self-organizing network control,” in *Proceedings of the 7th International Workshop on Self-Organizing Systems*, May 2013.

5. Takuya Iwai, Daichi Kominami, Masayuki Murata, and Tetsuya Yomo, “Thermodynamics-based entropy ajustment for robust self-organized network controls,” in *Proceedings of the 38th Annual IEEE International Computers, Software, and Applications Conference*, July 2014.

6. Takuya Iwai, Daichi Kominami, Masayuki Murata, and Tetsuya Yomo, “Thermodynamic principle-based strategy to achieve balance between robustness and performance for self-organized network controls,” in *Proceedings of the 8th IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, September 2014.

Non-Refereed Technical Papers

1. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Proposal and evaluation of attractor selection-based coverage control in wireless sensor networks,” *Technical Report of IEICE (NS2009-120)*, December 2009 (in Japanese).

2. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Error-tolerant and energy-efficient coverage control based on attractor selection model for wireless sensor networks,” in *Proceedings of IEICE Society Conference*, September 2010.

3. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Proposal of autonomous task allocation for dynamic formation of service networks over wireless sensor and actuator networks,” *Technical Report of IEICE (NS2010-219)*, March 2011 (in Japanese).

4. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “A self-organization based device assignment mechanism for cooperative resource sharing in a wireless sensor and actuator network,” *Technical Report of IEICE (IN2012-8)*, April 2012 (in Japanese).

5. Takuya Iwai, Naoki Wakamiya, and Masayuki Murata, “Characteristic evaluation of response

threshold model for self-organizing network control," in *Proceedings of IEICE General Conference*, September 2012 (in Japanese).

6. Takuya Iwai, Daichi Kominami, and Masayuki Murata, "Thermodynamics-based coordinated control for self-organizing information networks," in *Proceedings of IEICE General Conference*, March 2014 (in Japanese).

7. Takuya Iwai, Daichi Kominami, Masayuki Murata, and Tetsuya Yomo, "Interpretation of self-organized network controls in terms of thermodynamics," *Workshop on IEICE Technical Committee on Information Network Science (NetSci)*, August 2014 (in Japanese).

Preface

Wireless sensor and actuator networks (WSANs) are increasingly recognized as an important part of social infrastructure. To make these networks work accurately in real environments, we enhance their robustness, which we define here as the ability to reach the normal operating state in any environment where location errors occur, where optimization of parameter settings is difficult, where node failures occur, and so on. Although robustness is an important characteristic, we cannot fully protect the network by only enhancing its robustness. The reason is that the environment changes dynamically. It is not sufficient for the network to stably select a solution that is suitable for a particular environment. We need to improve not only robustness but also adaptability. We thus define adaptability as the ability for the network to search for an appropriate state in a dynamically changing environment. We then refer to a system that has high robustness and adaptability as a resilient system. Biological systems are typical resilient systems, and there is no doubt that we draw some inspiration from biological systems.

In this thesis, we create a resilient WSAN. We achieve this goal in two steps. The first step is to realize mechanisms for the robustness of the network. We expect that the robustness of a biological system is derived from mechanisms by which an individual decides on its state based on the goodness of the system under the realistic assumption that there are many alternative solutions in its solution space. We can realize robust network controls by applying these mechanisms. To verify this expectation, we propose, evaluate, and analyze bio-inspired WSAN controls. The second step is to realize an adaptive mechanism for the network. Biological systems do not stabilize to a unique state. This contributes to achieving high adaptability, but existing network controls do not handle this issue well. We consider the design of a steady state with some perturbations in order to yield various solutions in addition to the optimal one, if it is known at all. This feature induces rapid transitions to a new steady state after environmental changes.

To begin, we realize a robust mechanism for bio-inspired WSAN controls against node location errors and sensing area shape errors caused by environmental noise. For this purpose, we propose and evaluate a bio-inspired coverage control as an example of bio-inspired WSAN controls. Most existing protocols use geometric algorithms for each node to estimate the degree of coverage inside the sensing area of a node and determine whether to monitor the surroundings or sleep. These algorithms require accurate information about the location, sensing area, and sensing state of neighbor nodes. Therefore, they suffer from environmental noise, which leads to node location errors and sensing area shape errors, leading to degradation of coverage and redundancy of active nodes. Our proposal succeeds in achieving high robustness against errors by successfully having each node autonomously decide on its state based on the degree of coverage of the target region. Through simulation, we show that the proposal can maintain coverage and prevent a redundant number of nodes from becoming active nodes.

Next, we realize a mechanism for robustness of bio-inspired WSAN controls against parameter settings. For this purpose, we propose and evaluate a bio-inspired device assignment control, so that we can focus on the robustness of bio-inspired WSAN controls against parameter settings. Since it is apparently wasteful and redundant to deploy an independent WSAN for each of the envisioned services, we need a device assignment mechanism for effectively sharing available devices among concurrent services while taking into account service requirements and devices. Although there are several proposals for centralized or deterministic device assignment mechanisms, they suffer from difficulty in designing an appropriate set of rules with fine-tuned parameters. In our proposal, an appropriate device assignment emerges as a consequence of the autonomous decisions of individual nodes by a fully distributed and self-organizing mechanism without deterministic and complicated rules. We conduct simulation experiments to show that our proposal can accomplish device assignment that is as effective as an existing deterministic mechanism. In addition, we show that our proposal is less sensitive to parameter setting, which is quite an important feature in realistic deployments.

Robustness against node failure and message loss is also an important property for WSANs.

We analyze the above-mentioned bio-inspired device assignment control from the perspective of robustness against node failures in a lossy environment. Throughout the analysis, we confirm that the number of assigned devices converges after failure as long as a sufficient number of devices for the service remain.

Finally, we propose a thermodynamics-based design policy for a bio-inspired WSAN control to retain appropriate randomness in its steady state. The bio-inspired network control relies on a probabilistic mechanism that is composed of positive and negative feedbacks and the control eventually stabilizes on the best solution. As a result, we cannot avoid a temporary loss of function once the solution fails due to environmental fluctuations. To prevent this and improve adaptability, we need to prevent stabilization on a specific solution. We thus make the network control continually search for alternative solutions. However, most bio-inspired network controls are not designed taking this issue into account. For this problem, we focus on the fact that the steady state in a general system can be analyzed in terms of thermodynamic free energy. On this basis, we formulate the free energy of the steady state in a bio-inspired WSAN control and discuss its design policy depending on the degree of environmental fluctuations for which it is prepared. Through simulation, we confirm that the design policy is valid for existing bio-inspired WSAN controls.

Through the work presented in this thesis, we realize resilient WSAN controls by proposing bio-inspired robust and adaptive mechanisms.

Acknowledgments

This thesis could not have been accomplished without the assistance of many people, and I would like to acknowledge all of them.

First of all, I would like to express my great gratitude to my superviser, Professor Masayuki Murata, for his generous guidance and insightful comments throughout my master's program and doctor's program.

I am heartily grateful to the members of my thesis committee, Professor Takashi Watanabe, Professor Toru Hasegawa and Professor Teruo Higashino of Graduate School of Information Science and Technology, Osaka University, and Professor Morito Matsuoka of Cyber Media Center, Osaka University, for their multilateral reviews and perceptive comments.

Also, I especially would like to express my sincere appreciation for Professor Naoki Wakamiya of Graduate School of Information Science and Technology, Osaka University. Without his continuous advices and supports, I would not have entered the Ph.D. program. I am also very grateful to Assistant Professor Daichi Kominami of Graduate School of Economics, Osaka University. He gave me a chance for a new research area and direction.

Furthermore, I must acknowledge Associate Professor Shin'ichi Arakawa, Associate Professor Go Hasegawa, Associate Professor Jun-nosuke Teramae, Assistant Professor Yuichi Ohsita, Assistant Professor Yuya Tarutani, Assistant Professor Masafumi Hashimoto and Assistant Professor Yuki Koizumi of Graduate School of Information Science and Technology, Osaka University and Dr. Kenji Leibnitz of National Institute of Information and Communications Technology and Assistant Professor Shinsuke Kajioka of Information Technology Center, Nagoya Institute of Technology for their valuable comments and suggestions on my study.

I express my appreciation to all of past and present colleagues, friends, and secretaries of the Advanced Network Architecture Research Laboratory and the Bio-system Analysis Laboratory, Graduate School of Information Science and Technology, Osaka University.

Finally, I deeply thank my parents and family for giving me invaluable supports throughout my life. This work would not have been possible with them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In addition to computers and smart-phones, sensor and actuator devices are becoming equipped with the functionality to communicate via the Internet. These devices continually generate a huge volume, variety, and velocity of data, which is generally called “Big Data”, and we can extract very useful knowledge by analyzing this data. Using the extracted knowledge, we can realize advanced information and environment control services, for example, logistics and power networks. The network environment for this is called the “Internet of Things” (IoT) which has recently received a lot of attention [1]. As a fundamental technology for realizing the IoT, WSANs, which are described later, have been attracting attention from many researchers [2, 3]. Many technical challenges remain, but the most important one is achieving reliability as information and communication network systems become increasingly important parts of social infrastructure. To improve reliability, we enhance robustness which we define here as the ability to reach a normal operating state in any environment. Robustness is an important characteristic. However, we cannot fully protect a network by only enhancing robustness. This is because there is no pre-defined appropriate state in a dynamically changing environment. It is not reasonable to assume that the network control can stably select an appropriate solution at a certain time. We therefore need to improve not only robustness but also adaptability. We define adaptability as the ability for a network control to search for a state that is suitable for the current environment. We refer to systems that have high robustness and adaptability as resilient systems. In conventional network controls, we make an appropriate control rule taking into account all situations which may occur. Actually, it is quite difficult, and it is inevitable that unexpected situations occur in an actual environment. In some cases, it causes fetal damage on wireless sensor and actuator networks. Additionally, the control rule becomes more complicated as we cope with more various situations. It leads to difficulty in optimizing it and removing bugs from it. In this way, there is a limit on improving its robustness and its adaptability by the above mentioned conventional approach, and we need another approach to realize resilient network controls. In this thesis, we focus on biological systems for this purpose. The biological systems are typical examples of resilient systems, and there is no doubt that we draw some inspiration from biological systems.

Single-purpose Wireless Sensor and Actuator Networks

WSANs [2, 3] are composed of nodes that have various types of sensor devices, such as thermometers, hygrometers, and illuminators; and actuator devices, such as buttons, switches, and monitors. Based on various information about the environment, machinery, and people collected from the sensor devices, WSANs can provide users with desirable information services and environmental and machinery controls, as appropriate for the time, place, occasion, and person. There are many technical requirements on the mechanisms in WSANs. For example, the control mechanism must be robust to packet loss and delay because wireless communication is very sensitive to environmental noise [4]. As another example, control mechanisms are required to be energy efficient. Nodes have very limited resources, such as a poor processor performance, small memory size, and small battery capacity. Therefore, an important challenge is to prolong network lifetime by simple mechanisms.

Multi-purpose Wireless Sensor and Actuator Networks

In single-purpose WSANs, nodes are deployed for a specific service, and are not shared among services. For example, both WSANs for intelligent indoor temperature control and intrusion detection employ nodes with a motion sensor to detect the location of people in a room, but these WSANs

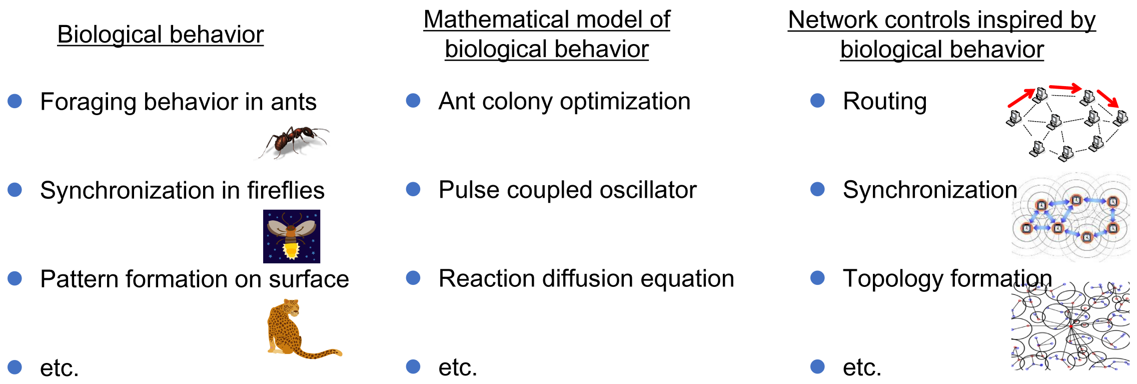

Biological behavior
Foraging behavior in ants
Synchronization in fireflies
Pattern formation on surface
etc.
Mathematical model of biological behavior
Ant colony optimization
Pulse coupled oscillator
Reaction diffusion equation
etc.
Network controls inspired by biological behavior
Routing
Synchronization
Topology formation
etc.

Figure 1.1: Example of bio-inspired network controls

are composed of dedicated nodes and are independent from each other. If they share those nodes and their sensing data, network resources and energy consumption can be saved. Furthermore, service-oriented deployment requires previous knowledge and careful planning about the types and locations of nodes to be deployed. However, it is hard to predict all the events that may occur in an area and prepare WSANs for all envisioned service requirements. Considering above-mentioned issues, the interests of researchers are shifting from single-purpose WSANs to multipurpose WSANs where multiple concurrent services are running on a single WSAN [5]. One solution for handling the dynamic and diverse requirements of services is to distribute heterogeneous nodes and organize them as a single monolithic WSAN [6, 7]. However, it is expensive in terms of bandwidth and energy to control and manage a large WSAN.

Network Control based on Mathematical Model of Biological Behavior

Biologically inspired (bio-inspired) network controls mimic the behavior of biological organisms, and they are based on nonlinear mathematical models that explain or reproduce biological behavior. As illustrated in Figure 1.1, examples of mathematical models include an ant colony optimization model for the foraging behavior of ants, a pulse-coupled oscillator model for the synchronized flashing of fireflies, and a reaction diffusion model for pattern formation on the body surface [8–10]. These mathematical models are applied to network controls such as routing, time synchronization, and topology organization [11–13]. Successful attempts published in the literature support the usefulness and versatility of bio-inspired network controls.

Since mathematical models of biological behavior have been shown to have excellent characteristics, network controls based on biological mathematical models are expected to be robust, adaptive, and scalable. It is quite important that we deeply understand the roles of parameters in biological mathematical models. This contributes to realizing excellent network control by slight extension of the models. However, bio-inspired network controls are not necessarily versatile. One control may achieve the best performance in one environment while being useless in another. Furthermore, bio-inspired network controls often experience a variety of errors including packet loss, packet delay, and node failure, which the original mathematical models do not consider. As a result, the models do not give the intended results in actual environments. It is therefore important that we deeply understand the biological mathematical models, especially in regard to their fundamental limits and applicability to network controls affected by errors [14, 15].

Resilience and its Origin in Bio-inspired Wireless Sensor and Actuator Networks

We here describe resilience in a bio-inspired WSAN in terms of robustness and adaptability [16].

Robustness

Robustness is defined as the ability for a network to reach a normal operating state in any environment where location errors occur, where optimization of parameter setting is difficult, where node failures occur, and so on. We expect that the robustness of a biological system is derived from the mechanisms by which an individual decides on its state based on abstract information about the goodness of the system. We expect that robustness is also obtained from a feature where there are many alternative solutions in its solution space. We can realize a robust WSAN by applying these biological mechanisms and features in WSAN control.

Adaptability

We define adaptability as the ability of a network to search for an appropriate state in a dynamically changing environment. Biological systems are composed of positive and negative feedbacks. On finding a better solution, selection becomes aggressive through positive feedback. The best solution is then eventually selected more frequently through negative feedback. However, biological systems do not stabilize to a specific state. This contributes to achieving high adaptability, but existing network controls do not give good consideration to this issue. We therefore consider designing a steady state with some perturbation in order to yield various solutions in addition to the optimum solution. This feature induces rapid transitions to new steady states when the environmental changes.

1.2 Outline of Thesis

In this thesis, we consider two issues in order to realize bio-inspired resilient WSANs. The first issue is that we need to realize robust bio-inspired mechanisms for WSANs. In bio-inspired WSAN controls, a node can select its state to improve the goodness of the control state. This feature contributes to high robustness against small errors, but the details have not been well studied. Therefore, we propose bio-inspired WSAN controls and evaluate them from the perspective of robustness against errors such as location errors, sensing area shape errors, packet loss, and node failures. The second issue is that we need to design a steady state of bio-inspired WSAN controls. Although bio-inspired WSAN controls rely on probabilistic mechanisms, most of them eventually stabilize on a specific solution. As a result, disruption of communication once the solution fails is unavoidable. For high adaptability, we make the system continually and randomly search for alternative solutions. However, their mechanisms are not designed taking this issue into account. Therefore, we propose a thermodynamics-based design policy for bio-inspired WSAN control to retain appropriate randomness in the steady state.

Robust Coverage Control against Localization Error [17–21]

We begin this thesis by proposing a robust bio-inspired mechanism for WSAN against node location errors and sensing area shape errors caused by environmental noise in Chapter 2. The coverage problem of WSANs is an important issue for prolonging network lifetime while guaranteeing that the target region is monitored by a sufficient number of active nodes [22]. Most existing protocols use geometric algorithms for each node to estimate the degree of coverage and determine whether to monitor the surroundings or sleep [23–25]. These algorithms require accurate information about the location, sensing area, and sensing state of neighbor nodes. Therefore, they suffer from localization errors leading to degradation of coverage and redundancy of active nodes. In addition, they introduce communication overhead leading to energy depletion. In this chapter, we propose a novel coverage control mechanism in which each node relies on neither accurate location information nor communication with neighbor nodes. To enable autonomous decisions by the nodes, we adopt a nonlinear mathematical model of adaptive behavior from biological systems for dynamically changing environments. Through simulation experiments, we show that the proposal outperforms the existing protocol in terms of the degree of coverage per node and the overhead under the influence of localization error.

Robust Device Assignment Control against Parameter Setting [26–29]

In Chapter 3, we focus on the robustness of bio-inspired WSAN controls against parameter settings. Since it is apparently wasteful and redundant to deploy an independent WSAN for each envisioned application, building a multipurpose WSAN consisting of heterogeneous sensors and actuators and sharing them among applications are considered promising [5]. However, we need a mechanism for effectively sharing available resources among concurrent applications while taking into account application requirements and resources [7, 30, 31]. Although several proposals have been made on centralized or deterministic device assignment mechanisms, they suffer from difficulty in designing an appropriate set of rules with fine-tuned parameters [32]. In this chapter, we propose a fully distributed and self-organizing device assignment mechanism by adopting a response threshold model which imitates the division of labor in a colony of social insects. Our proposal does not require deterministic and complicated rules, and appropriate device assignment emerges as a consequence of the autonomous decisions of individual nodes. Through simulation experiments, we confirm that our proposal accomplishes device assignment that is as effective as an existing deterministic mechanism, and that our proposal is less sensitive to parameter settings.

Robustness of Division of Labors-based Network Control against Failure [33–35]

We next investigate the robustness of bio-inspired WSAN controls against node failure in lossy environments in Chapter 4. Bio-inspired WSAN controls are not necessarily versatile. A control may achieve the best performance in one environment while it being useless in another. Furthermore, bio-inspired network control mechanisms often experience a variety of perturbations such as loss of information and failure of nodes, which original bio-models do not consider. As a result, they fail to provide the intended results in the actual environment. We therefore need a deep understanding of biological mathematical models especially in regard to their fundamental limits and applicability to network controls suffering from perturbation [12, 14, 15]. In this chapter, we analyze the robustness of response threshold model-based network control against information loss and individual failures for the example of a response threshold model. Through analysis, we confirm that the number of workers recovers from individual failures even if dead individuals and isolated individuals exist. Moreover, the recovery time does not diverge as more individuals die.

Design Policy for Bio-inspired Network Control to Achieve High Adaptability [36–40]

Finally, in Chapter 5, we consider a method for equipping a network control with sufficient adaptability depending on its design requirements. The behavior of a bio-inspired network control is explained as a competition between ordering energy, which keeps the behavior ordered, and disordering energy, which induces disordered behavior. On finding a good solution, the ordering energy overcomes the disordering energy, and the system stabilizes on the found state. If it uses only a single state due to weakness of its disordering energy, it is vulnerable to environmental fluctuations such as node failure and link failure. The balance between both of these energies in the steady state implicitly determines the maximum degree of environmental fluctuations that the network can endure. Therefore, we must design the balance depending on the frequency of environmental fluctuations in the operating environment. However, there is no universal interpretation of both energy and policy for designing the quantities of both energies. In this chapter, we focus on the fact that the temperature of a natural substance determines the balance of ordering and disordering energies as an idea for these tackling these problems [41]. By treating the degree of environmental fluctuation as temperature, we can determine an appropriate balance between both the energies. Through mathematical analysis and simulation experiments, we verify the design policy using an existing multi-path routing control.

Chapter 2

Robust Coverage Control against Localization Error

A coverage problem is one of important issues to prolong the lifetime of a wireless sensor and actuator network while guaranteeing that the target region is monitored by a sufficient number of active nodes. Most of existing protocols use geometric algorithm for each node to estimate the degree of coverage and determine whether to monitor around or sleep. These algorithms require accurate information about the location, sensing area, and sensing state of neighbor nodes. Therefore, they suffer from localization error leading to degradation of coverage and redundancy of active nodes. In addition, they introduce communication overhead leading to energy depletion. In this chapter, we propose a novel coverage control mechanism, where each node relies on neither accurate location information nor communication with neighbor nodes. To enable autonomous decision on nodes, we adopt the nonlinear mathematical model of adaptive behavior of biological systems to dynamically changing environment. Through simulation, we show that the proposal outperforms the existing protocol in terms of the degree of coverage per node and the overhead under the influence of localization error.

2.1 Introduction

A wireless sensor and actuator network [3] has been attracting many researchers over the past ten years for a variety of its applications. Among them, surveillance, monitoring, and observation of items, objects, and regions are most promising and useful. These applications require that a sufficient number of sensor nodes monitor the target region. Due to the uncertainty and instability of location and sensing area, it is difficult to deploy and manage sensor nodes in an optimal manner, i.e. placing a minimum number of sensor nodes at the optimal positions. Therefore, a redundant number of sensor nodes are generally deployed in the target region. Then, for energy conservation, a sophisticated sleep scheduling mechanism is employed to keep the number of active sensor nodes as small as possible and let sensor nodes sleep as much as possible while satisfying the application's requirement on the degree of coverage. Such an issue to minimize the number of active sensor nodes while guaranteeing the required degree of coverage is called coverage problem [22, 42, 43]. There are many proposals on coverage problem. However most of them rely on unrealistic assumptions, e.g. accurate location and perfect circular sensing area, and do not work well in the error-prone environment.

In this chapter, to solve the problem, we propose a novel coverage control protocol, which is free from the above-mentioned unrealistic assumptions. Each sensor node does not need to know the shape and size of sensing area and the location and state of neighbor sensor nodes. A sensor node only relies on the information about the degree of coverage of the target region. To enable autonomous decision on sensor nodes, we adopt the nonlinear mathematical model called the attractor selection model. The model imitates flexible and adaptive behavior of biological systems to dynamically changing environment [44]. A biological system can autonomously and adaptively select an appropriate state for the environment only based on the condition of itself. Through simulation, we show that the proposal outperforms an existing protocol in the terms of the degree of coverage per sensor node under the influence of localization error. In addition, our proposal requires less energy in monitoring the target region.

The rest of this chapter is organized as follows. First in Section 2.2, we briefly discuss related work. Next, in Section 2.3, we introduce the biological attractor selection model. Then, in Section 2.4, we propose a novel coverage maintenance protocol adopting the attractor selection model. In Section 2.5, we evaluate the proposal through comparison with an existing protocol. In Section 2.6, we discuss the robustness of the proposal against localization error. Finally, in Section 2.7, we conclude this chapter.

2.2 Related Work

There are many proposals on coverage problem, but most of them use geometric algorithm in order to estimate the degree of coverage. Based on the estimated degree of coverage, each sensor node determines whether to monitor around or sleep. For example, CCP [23] adopts the so-called K_s-Eligibility algorithm. First a sensor node identifies intersection points of borders of sensing areas of neighbor sensor nodes using a geometric arithmetic. Then, the sensor node evaluates whether all of intersection points inside its sensing area are inside sensing areas of more than K_s active sensor nodes or not. Since CCP assumes the accurate location information and perfect circular sensing area with radius R_s on all sensor nodes, it suffers from errors in the location information and the irregularity of the size and shape of sensing area. In addition, for a sensor node to evaluate the K_s-Eligibility algorithm, it has to obtain information about the location, sensing area, and state of neighbor sensor nodes at the sacrifice of bandwidth and energy in message exchanges. To increase the robustness against localization error, a location free coverage maintenance protocol is proposed in [25]. The protocol adopts dominating set of graph theory, but it requires a sensing area to be circular and a transmission range to be adjustable. CARES [24] is another location free protocol, where each sensor node stochastically and independently chooses its state based on a general Markov model. However, sensor nodes must be uniformly distributed in the target region and the shape of sensing area must be circular. In the actual environment, localization error amounts to as much as several meters [45] and the shape of sensing area is not always circular at all. Therefore, these existing schemes do not work well outside the ideal environment and an error tolerant coverage control method is desired.

2.3 Attractor Selection Model

The attractor selection model imitates the adaptive metabolic synthesis of *Escherichia coli* cells to dynamically changing nutrient condition [44]. A mutant bacterial cell has a metabolic network consisting of two mutually inhibitory operons, each of which synthesizes the different nutrient. When a cell is in the neutral medium, where both nutrients sufficiently exist, mRNA concentrations dominating protein production are at the similar level. This means that a cell can live and grow independently of the nutrient, which the cell synthesizes. Once one of nutrients becomes insufficient in the environment, the level of gene expression of an operon corresponding to the missing nutrient eventually increases so that a cell can survive by compensating the nutrient. Although there is no embedded adaptation rule as a signal transduction pathway, a cell can successfully adapt gene expression in accordance with the surrounding condition.

In the attractor selection model, the dynamics of mRNA concentration m_1 and m_2 are formulated by following equations.

$$\frac{dm_1}{dt} = \frac{s(A)}{1+m_2^2} - d(A)m_1 + \eta_1 \tag{2.1}$$

$$\frac{dm_2}{dt} = \frac{s(A)}{1+m_1^2} - d(A)m_2 + \eta_2 \tag{2.2}$$

A $(1 \geq A \geq 0)$ is the cellular activity such as growth rate and expresses the goodness of the current behavior, i.e. the state of gene expression. Functions $s(A)$ and $d(A)$ are rational coefficients of mRNA synthesis and decomposition, respectively. In [44], $s(A) = \frac{6A}{2+A}$ and $d(A) = A$ are used. η_i $(i = 1, 2)$ corresponds to internal and external noise or fluctuation in gene expression.

Now let us explain the dynamics of mRNA concentrations following the attractor selection model. An attractor is a stable state, where a nonlinear dynamic system reaches after an arbitrary initial state. When the activity A is high, the nonlinear dynamic system formulated by the above equations has one attractor where $m_1 = m_2 = m^*$. Here, m^* is a constant and larger than one. When the sufficient nutrients are available, a cell grows well. Thus, a cell stays at the attractor and generates either one of two nutrients. Next, we assume that the environment lacks the nutrient, which a cell does not synthesize. Since it does not have the sufficient nutrient to grow, the activity decreases. When the activity becomes low, there appears two attractors, i.e. $m_1 = m^*$ and $m_2 = 1/m^*$, or $m_1 = 1/m^*$ and $m_2 = m^*$, where either one of mRNA concentrations is higher than the other. Since the first two terms of the right side of Equations (2.1) and (2.2) are multiplied by the activity, potential of attractors are shallow and dynamics is dominated by the noise terms. Consequently, m_1 and m_2 begin to change at random. When the mRNA concentration of the missing nutrient occasionally becomes large in a cell, the activity slightly increases as the cell can live better. The increase in the activity makes the potential of the attractor deeper and the state of a cell moves toward the attractor by entrainment. The activity further increases accordingly and the influence of noise becomes smaller. Eventually, the state of a cell reaches an appropriate attractor and stays there stably as far as the nutrient condition does not change.

The attractor selection model is a kind of meta-heuristics of optimization problem with dynamically changing given conditions. In the model, possible solutions are defined as attractors of the dynamic system by stochastic differential equations. An objective function to maximize is defined as the activity. In the biological case, a bacterial cell adaptively selects one of solutions, i.e. synthesis of either one of two nutrients, so that the cell can maximize its growth rate according to the environmental nutrient condition. In our application of the attractor selection model to coverage control, a sensor node selects one of two states, i.e. monitor around or sleep, to maximize the activity defined as the degree of coverage in the target region.

2.4 Our Proposal

In this section, we first outline the basic behavior of our proposal. Then, we describe the attractor selection model adopted in our proposal and the definition of the activity in coverage control. Finally, we describe the detailed behavior of sensor nodes in our proposal.

2.4.1 Overview

In this section, we consider a periodic monitoring application, where a sink collects sensing data from sensor nodes at regular intervals as illustrated in Figure 2.1. We refer to the interval as *data gathering interval* and the beginning of data gathering as *timing of data gathering*. We define

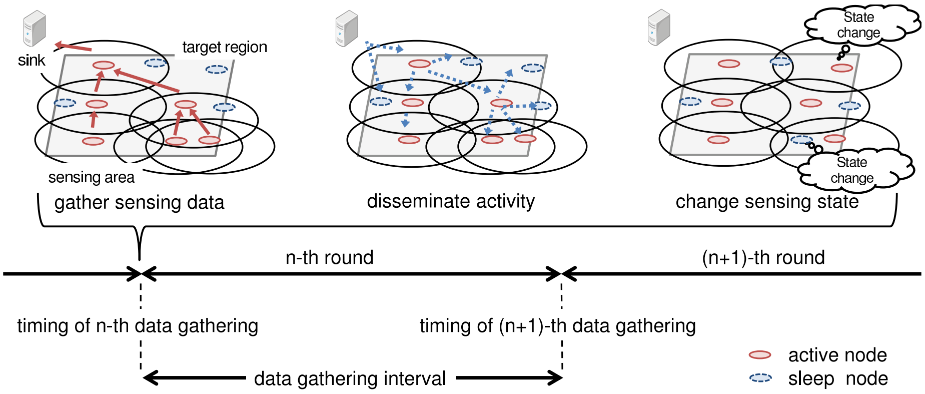


Figure 2.1: Overview of our coverage control

the duration from the n-th timing of data gathering until just before the $(n + 1)$-th timing of data gathering as the n-th *round*.

At each timing of data gathering, each sensor node, which was active in the preceding round, transmits a message to a sink by single or multi-hop communication. A message consists of sensing data and the information for the sink to estimate the degree of coverage of the target region. Since we focus on coverage control, we do not assume any specific data gathering mechanism to collect sensing data from sensor nodes. We also assume that the connectivity is maintained when the sufficient coverage is achieved [23]. Using received messages, a sink evaluates the degree of coverage of the target region. The way to evaluate the degree of coverage depends on the requirement of application and the information that sensor nodes can provide. When any localization mechanism is available at sensor nodes, the coverage is estimated based on the relative or absolute location of sensor nodes. An identifier of objects that a sensor node monitors is also useful information when a sink knows locations of the objects in the target region. In this case, each sensor node does not need to know its own location. From the degree of coverage, a sink derives the activity.

Then, a sink disseminates the activity information over a wireless sensor and actuator network by using any efficient dissemination mechanism, e.g. flooding, gossiping, or tree-based. Not only sensor nodes that are active in the preceding round but one whose sleep timer expires at the timing of data gathering receive the activity. Sensor nodes that receive the activity decide whether to be active or sleep using the attractor selection model-based state selection mechanisms described below. If a sensor node decides to be active, it starts monitoring its surroundings. Otherwise, the sensor node sets its sleep timer at multiples of data gathering interval and sleep immediately.

2.4.2 Extended Attractor Selection Model

In our proposal, we use the following attractor selection model, which is introduced in [46] for adaptive ad-hoc network routing.

$$\frac{dm_1}{dt} = \frac{syn(\alpha)}{1+m_2^2} - deg(\alpha)m_1 + \eta_1 \tag{2.3}$$

$$\frac{dm_2}{dt} = \frac{syn(\alpha)}{1+m_1^2} - deg(\alpha)m_2 + \eta_2 \tag{2.4}$$

and

$$syn(\alpha) = \alpha \times (\beta \times \alpha^{\gamma} + \varphi^*) \tag{2.5}$$

$$deg(\alpha) = \alpha \tag{2.6}$$

This model has two attractors, i.e. $m_1 > m_2$ or $m_2 > m_1$. $\beta\ (> 0)$ is a parameter related to the stability of attractor and $\gamma\ (> 0)$ is a parameter related to the speed of convergence. φ^* is a constant for the dynamic system to have stable attractors and we use $1/\sqrt{2}$. $\alpha\ (1 \geq \alpha \geq 0)$ is the activity derived from the degree of coverage. The derivation of the activity will be explained in the next Section.

2.4.3 Derivation of Activity

In our proposal, as stated in Section 2.4.1, any estimation algorithm of the degree of coverage can be adopted. Therefore, our proposal does not necessarily need coordinates of sensor nodes, and identifiers of their installation locations, e.g. room number and landmark number, can be substituted for this algorithm. In this chapter, we consider the following derivation for the sake of easy implementation and comparison.

First, the target region is divided into small regions of 1 [m] $\times\, 1$ [m], which is called *patch*. In the target region of x_t [m] $\times\, y_t$ [m], a patch at the column x $(x_t \geq x \geq 1)$ and the row y $(y_t \geq y \geq 1)$ is indicated by (x, y). The degree of coverage $C(x, y)$ of patch (x, y) is approximated by the number of active sensor nodes whose sensing area covers a center of patch (x, y).

Guaranteeing any point of the target region to be monitored by k active sensor nodes is called k-coverage. When an application requires k-coverage, the sensing ratio S $(1 \geq S \geq 0)$ of the whole target region is derived by the following equation.

$$S = \frac{\left|\{(x, y) \mid C(x, y) \geq k\}\right|}{x_t y_t} \tag{2.7}$$

The sensing ratio S does not take into account the excess and deficiency in monitoring, that is, whether a patch is in the sensing area of more or less than k active sensor nodes. Therefore, coverage control using the sensing ratio S as the activity α leads to the waste of energy or deficient coverage. To solve this problem, we formulate the excess and deficiency ratio E (≥ 1) for the whole region.

$$E = \frac{\sum_{i=1}^{x_t} \sum_{j=1}^{y_t} |C(i, j) - k|}{x_t y_t} + 1 \tag{2.8}$$

Then, the activity α for the whole target region is derived as follows.

$$\alpha = \frac{S}{\max\{1, wE\}}, \tag{2.9}$$

where larger w $(1 \geq w > 0)$ leads to more efficient control with less active sensor nodes, but it becomes difficult for sensor nodes to reach solutions, which are deficient or redundant coverage. Operator "$\max$" is introduced to prevent the activity from exceeding one. We call the activity derived in Equation (2.9) the *global activity*.

For fine-grained control, we can also define the area activity using the sensing ratio per small areas of the target region. In this case, the target region is divided into some sub-areas of x_s [m] $\times$ y_s [m], where x_s and y_s are divisors of x_t and y_t. A sub-area at the column x and the row y is indicated by (x, y), where $x_t/x_s \geq x \geq 1$ and $y_t/y_s \geq y \geq 1$. The sensing ratio $S'(x, y)$ of sub-area (x, y) is derived by the following equation.

$$\begin{aligned} S'(x', y') = \big|\{(x, y) \mid C(x, y) &\geq k, \\ (x' - 1)x_s + 1 \leq x &\leq x'x_s, \\ (y' - 1)y_s + 1 \leq y &\leq y'y_s\}\big|/x_s y_s \end{aligned} \tag{2.10}$$

We formulate the excess and deficiency ratio $E'(x, y)$ (≥ 1) for the sub-area (x, y) as follows.

$$E'(x, y) = \frac{\sum_{i=(x-1)x_s+1}^{xx_s} \sum_{j=(y-1)y_s+1}^{yy_s} |C(i, j) - k|}{x_s y_s} + 1 \tag{2.11}$$

Then, the activity $\alpha'(x, y)$ of the sub-area (x, y) is given as follows.

$$\alpha'(x, y) = \frac{S'(x, y)}{\max\{1, wE'(x, y)\}} \tag{2.12}$$

The activity derived by Equation (2.12) is called the area activity. In the case of the area activity-based control, a sink evaluates all area activities $\alpha'(x, y)$ in Equation (2.12) and a message from a sink contains all area activities. A sensor node uses the area activity of a sub-area in which the sensor node considers to be located. It implies that a sensor node with inaccurate location information uses the area activity of an inaccurate sub-area.

2.4.4 Node Behavior

A sensor node has three states, i.e. active, sleep, and intermediate as illustrated in Figure 2.2. In each state, a sensor node behaves as follows.

Active state: A sensor node monitors its sensing area by turning and keeping sensor modules on and transceiver modules off for the fixed period I_s (> 0) [s], which is called *sensing interval*. When the timing of data gathering arrives, a sensor node turns on transceiver modules and sends sensing data toward the sink. Then, it moves to the intermediate state.

Sleep state: A sensor node turns and keeps all modules off to save its battery. When a sleep timer expires, a sensor node turns on transceiver modules and moves to the intermediate state.

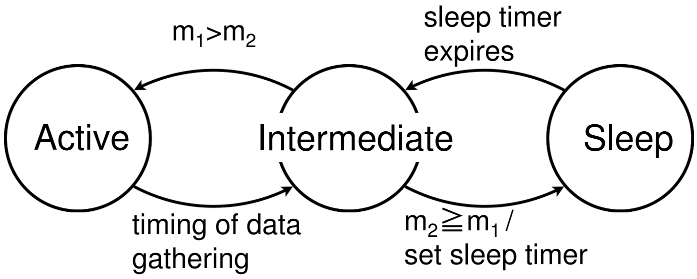


Figure 2.2: State diagram of our coverage control

Intermediate state: A sensor node waits for receiving a feedback message from the sink during the fixed period I_w (> 0) [s], called *intermediate interval*. The feedback message contains the activity α, which reflects the degree of coverage. The node evaluates two equations in Section 2.4.2 to update m_1 and m_2 using the received activity. In this chapter, we assume that above-mentioned transactions are finished within the constant time I_w. Using updated m_1 and m_2, sensor nodes select the next state as following. In case of $m_1 > m_2$, the sensor node moves to the active state. On the other hand, in case of $m_2 \geq m_1$, the sensor node sets its sleep timer as $I_s + l \times (I_s + I_w)$ and moves to the sleep state. l (> 0) is a control parameter which is randomly chosen with uniform distribution between 0 and 4 to avoid synchronous behavior of sensor nodes. $I_s + I_w$ corresponds to the data gathering interval introduced in Section 2.4.1.

Next we briefly explain how a sensor node behaves in message transmission in simulation. In the case of a sensor node which was in the active state in the preceding round, it participates in both data gathering and feedback dissemination as illustrated in Figures 2.3 and 2.4. At the end of active period I_s, the timing of data gathering comes. Although mechanisms of data gathering and feedback dissemination are out of scope of this chapter, we here consider a tree-based routing. A sensor node located at i hop from a sink receives messages from its child sensor nodes and aggregates their sensing data with its own. Then, it sends a message containing aggregated sensing data to its parent sensor node located at $(i - 1)$ hop from a sink and moves to the intermediate state.

During feedback dissemination, a sensor node located at i hop from a sink first receives a feedback message from its parent sensor node during the intermediate interval I_w. Then, it broadcasts the message to its child sensor nodes located at $(i + 1)$ hop from a sink and determines the next

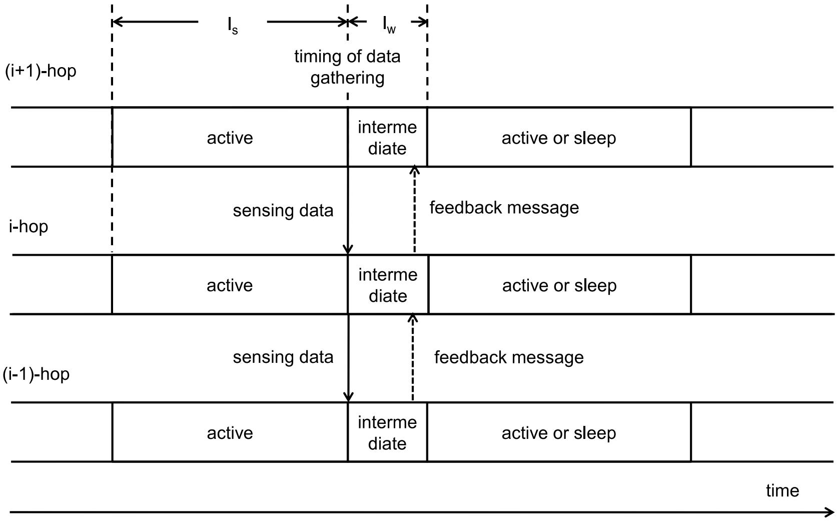


Figure 2.3: Behavior of a sensor node on i-hop when its preceding state is active

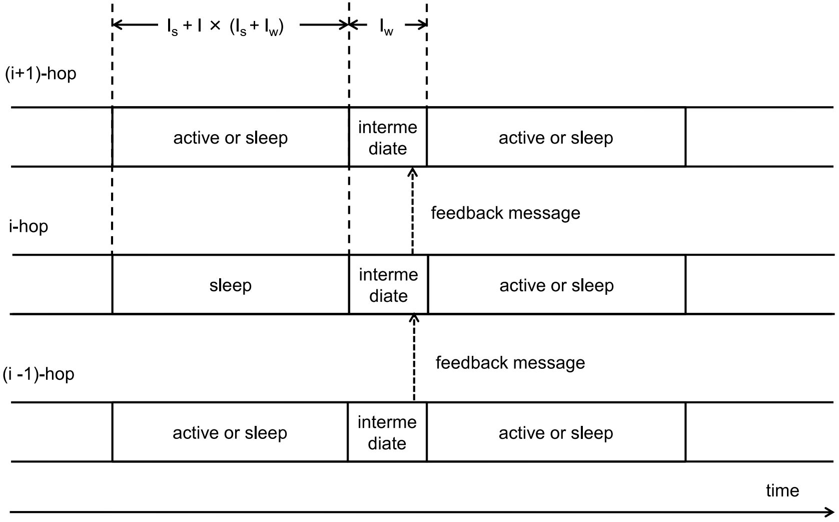


Figure 2.4: Behavior of a sensor node on i-hop when its preceding state is sleep

state. On the contrary, when a sensor node located at i hop from a sink was in the sleep state in the preceding round, it does not send sensing data. It wakes up at the timing of data gathering and immediately moves to the intermediate state. Next, it receives a feedback message from its parent sensor node, which was in either of the active or sleep state in the preceding round. Then, it forwards the message to its child sensor nodes and makes a decision on the next state.

2.4.5 Advantages

Our proposal has advantages over existing protocols, which require a sensor node to obtain the information of neighbor sensor nodes, i.e. location and state. First, our proposal is more robust against the inaccuracy of location information and the irregularity or uncertainty of sensing area than others. In our proposal, a sensor node only requires the degree of coverage of the whole target region or the located area. Even if the derivation of the degree of coverage at a sink uses location information of sensor nodes, the influence of localization error can be mitigated by considering the degree of coverage over the whole target region or the area of a certain size.

Second, our proposal requires less energy in coverage control than others. In other existing proposals, so that a sensor node can appropriately determine the next state using a geometric algorithm, it has to collect sufficient amount of information by receiving many messages from neighbor sensor nodes. Although a sensor node only needs to broadcast a message once to inform neighbor sensor nodes of its information, such message exchanges must be done in addition to regular message transmission for data gathering. On the other hand, our proposal only requires a sensor node to obtain the activity for selecting its sensing state. A sensor node only needs to transmit one message for data gathering and one more for feedback dissemination. Therefore, a sensor node can effectively turn off its transceiver for longer duration than others. These advantages of our proposal will be proved by simulation in the next Section.

2.5 Simulation Experiments

In this section, we first explain error models, i.e. localization error and shape error. Simulation results follow to compare our proposal with CCP in terms of the sensing ratio, the number of active sensor nodes, the redundancy ratio, the contribution ratio, and the energy consumption.

2.5.1 Localization Error

Based on [47], we consider a simple model of localization error. The amount of error is uniformly distributed between $-u$ and u, where u is the maximum error in meter. Then, erroneous coordinates of a sensor node at geographical coordinates (x, y) is given at random in the area of $(x - u, y - u)$ as the left bottom corner and $(x + u, y + u)$ as the right top corner.

In our proposal, a sink evaluates the global or area activity with wrong location information received from neighbor sensor nodes. Therefore, the activity notified to sensor nodes is different from the actual degree of coverage. On the other hand, a sensor node with CCP calculates intersections of sensing areas based on wrong location information. Therefore, the K_s-Eligibility algorithm would give a wrong answer.

2.5.2 Shape Error

Since there is no model of the irregularity of sensing area, we adopt the model of the irregularity of radio propagation introduced in [48]. RIM (Radio Irregularity Model) models the variation in the received signal strength under the influence of heterogeneous energy loss. In wireless communication, the signal strength decreases in accordance with the distance from the transmitter. The following is the commonly used model to estimate path loss L [dBm] [4].

$$L = C + 10n \log_{10} d, \tag{2.13}$$

where C is a constant and n expresses the quality of transmission path. Parameter d is the distance between the transmitter and the receiver. Then, RIM introduces the irregularity in path loss as:

$$R = T - DOIAdjustedPathLoss + F, \tag{2.14}$$

$$DOIAdjustedPathLoss = L \times K_i. \tag{2.15}$$

R represents the received signal strength and T corresponds to the transmission power. F corresponds to the fading effect. K_i implements the difference in path loss at the i-th degree. K_i is given

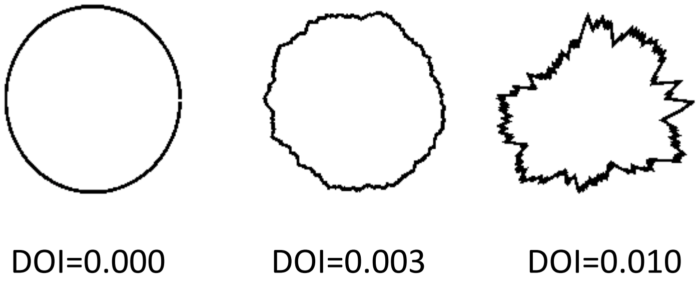


Figure 2.5: Shape of irregular sensing area

by the following equation:

$$K_i = \begin{cases} 1, & \text{if } i = 0 \\ K_{i-1} \pm rDOI, & \text{if } 360 > i > 0 \wedge i \in N \end{cases} \tag{2.16}$$
$$where\; DOI \geq |K_0 - K_{359}|.$$

Here, DOI (Degree Of Irregularity) is the coefficient of the irregularity. r is a random number following the Weibull distribution.

For example, we depict the impact of different DOI in Figure 2.5. Each shape shows the border of region where the received signal strength exceeds a certain threshold. As can be seen, $DOI = 0$ gives a circular shape. As DOI increases, the shape becomes more irregular. We first set parameters of RIM appropriately to obtain the regular circle shape of the desired sensing radius and then change DOI to see the influence of irregularity in simulation experiments.

2.5.3 Energy Model

We define the energy model based on MICAz [49, 50]. CPU consumes 8 [mA] when it is on and 15 [uA] when it is off. A transceiver module consumes 19.7 [mA] in listening a channel and receiving a message and 17.4 [mA] in transmitting a message. A sensor module consumes 10 [uA] when it is on and 0 [uA] when it is off. When a sensor module monitors objects, CPU is activated as well. We assume that a sensor node runs on two AA batteries of 3 [V].

As explained in Section 2.4.4, we consider a tree-based routing for data gathering and feedback dissemination. In data gathering, a sensor node receives sensing data from its child sensor nodes, generates the aggregated data of the same size of a single sensing data, and sends it to a parent sensor node. In disseminating feedback messages, a sensor node receives a message containing the activity from its parent sensor node and broadcasts it to all child sensor nodes.

2.5.4 Simulation Setting

We distribute about 10,000 sensor nodes in the square target region. A sink is located in the center of the target region. In the case of the global activity-based control, 10,000 sensor nodes are randomly deployed in the target region of 500 [m] $\times$ 500 [m]. In the case of the area activity-based control, we first set the size of a sub-area and then determine the size of the target region as the multiple of a sub-area around 500 [m], while keeping the density 0.04 [node/m^2]. For example, when the size of sub-area is 15 [m] $\times$ 15 [m], 10,404 sensor nodes are distributed in the target region of 510 [m] $\times$ 510 [m]. An application requires $1, 2,$ or 3-coverage ($k = 1, 2, \text{ or } 3$). Data gathering interval $(I_s + I_w)$ is set at 10 [s]. Sensing interval I_s is 9 [s] and wakeup interval I_w is 1 [s]. At the beginning of a simulation run, all sensor nodes are in active state.

In our proposal, both m_1 and m_2 are initialized to 1 and the initial activity is initialized to 0. Parameter β and γ are set at 2.5 and 1.2, respectively. Weight w is set at 0.5. The parameter l of rounds of sleep state in our proposal is randomly chosen between 0 and 4 with uniform distribution. These parameters are selected through preliminary experiments. In CCP, HELLO interval, SLEEP, WITHDRAW, JOIN, and LISTEN timers are set at 1 [s], 10 [s], 1 [s], 1 [s], and 1 [s], respectively. Regarding details of these parameters, refer to [23]. For the purpose of comparison, we define ACTIVE and JOIN state of CCP as active state.

The communication range R_c is set at 20 [m]. We use our own simulator and we assume the ideal communication environment. That is, there is no loss or delay of message. The shape of sensing area is a circle of radius $R_s = 10$ [m] and identical among sensor nodes under the condition without shape error. In our proposal, a sink assumes the circular sensing area with radius 10 [m] and believes the location information reported by sensor nodes in derivation of the activity. In CCP, intersection points between borders of sensing areas of neighbor sensor nodes are calculated under the assumption that there is neither localization error nor shape error. For evaluation of the tolerance to localization error, we change the maximum location error u from 0 [m] to 10 [m], e.g. GPS-based localization. For evaluation of the tolerance to shape error, we change DOI from 0 to 0.03.

2.5.5 Performance Measures

As performance measures, we use the number of active nodes N, the contribution ratio B, the redundancy ratio U, and the energy consumption O. The contribution ratio B indicates the degree of contribution of an active sensor node to coverage. B is derived as,

$$B = \frac{M \times S}{N} \text{ [m}^2\text{]}, \tag{2.17}$$

where M [m^2] is the size of target region and S is the sensing ratio derived by Equation (2.7) with the accurate coordinates and sensing area. Therefore, the contribution ratio represents the average area that an active sensor node is responsible for monitoring. The larger contribution ratio means that sensor nodes are more efficiently monitoring.

Next, we define the redundancy ratio U as the averaged extra-degree of coverage per patch for achieving k-coverage. The redundancy ratio is derived as:

$$U = \frac{\sum_{i=1}^{x_t} \sum_{j=1}^{y_t} Z(C(i,j))}{\left|\{(x,y) \mid C(x,y) \geq k\}\right|} \tag{2.18}$$

and

$$Z(x) = \begin{cases} x - k + 1, & \text{if } x \geq k \\ 0, & \text{if } x < k \end{cases} \tag{2.19}$$

where the target region is x_t [m] $\times$ y_t [m] and the coverage $C(x,y)$ of patch (x,y) is approximated by the number of active nodes that has a center of patch (x,y) in its own sensing area. Therefore, the larger redundancy ratio means that too many nodes are in the active state.

Finally, the energy consumption O is derived using our energy model described in Section 2.5.3.

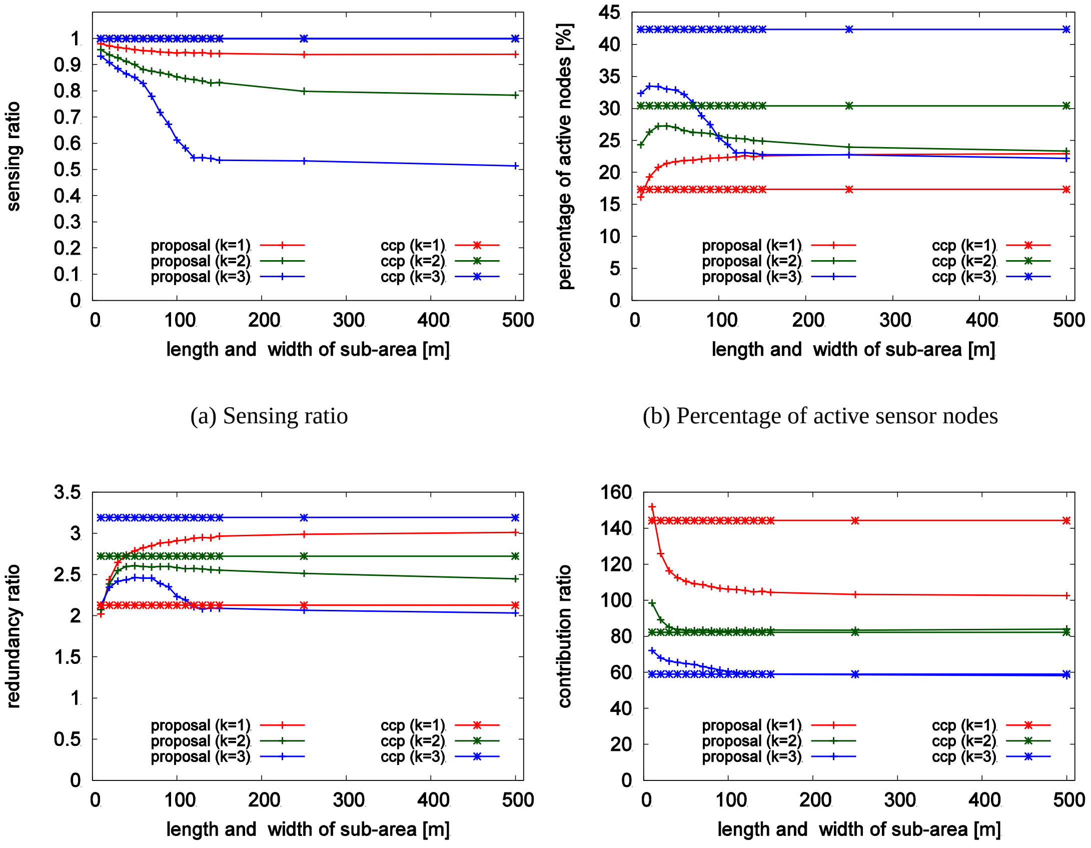
We take into account state-dependent energy consumption and energy consumed in message transmission and reception. We should note here that the overhead related to management of location information is not considered in the evaluations. First, we assume that a sink obtains identifiers and location-related information from all sensor nodes in advance. We further assume that both of CCP and our proposal adopt the same localization technique. Messages sent from a sensor node contain its identifier, whose size is small enough. As a result, the amount of overhead regarding management of location information is almost the same among CCP and our proposal and the difference is negligible. Influences of inaccuracy in location information are taken into account in the energy consumption O, since inaccurate location information affects states of sensor nodes and the amount of message transmission.

2.5.6 Basic Evaluation

First we compare our proposal with CCP under the ideal environment, where there is neither localization error nor shape error. In Figure 2.6, the x-axis indicates the width and height of a sub-area, i.e. x_s and y_s, for the area activity-based control. $x_s = y_s = 500$ [m] corresponds to the case of the global activity-based control where the target region is not divided into any sub-area. The y-axis shows the sensing ratio derived by Equation (2.7). When there is no error, CCP accomplishes the sensing ratio S of 1.0 for $k = 1$, 2, and 3 as shown in Figure 2.6(a).

Under the ideal environment, sensor nodes can accurately estimate the degree of coverage inside sensing areas of themselves. Figure 2.6(b) shows that the percentage of active sensor nodes with CCP increases almost in proportional to the required coverage. In spite of a deterministic and geometric algorithm of CCP, the redundancy ratio is higher than 2 and up to 3.2 as shown in Figure 2.6(c). Even if an uncovered area inside a sensing area of a sensor node is small, a sensor node becomes active state to cover the area. This results in the redundant coverage of the other area which is already covered. However, such redundancy is unavoidable for the irregularity of deployment of sensor nodes and the shape of sensing area.

Compared to CCP, the sensing ratio with our proposal is lower especially when the size of sub-area is large as shown in Figure 2.6(a). Our proposal adopts the meta-heuristic algorithm, i.e.



(a) Sensing ratio

(b) Percentage of active sensor nodes

(c) Redundancy ratio

(d) Contribution ratio

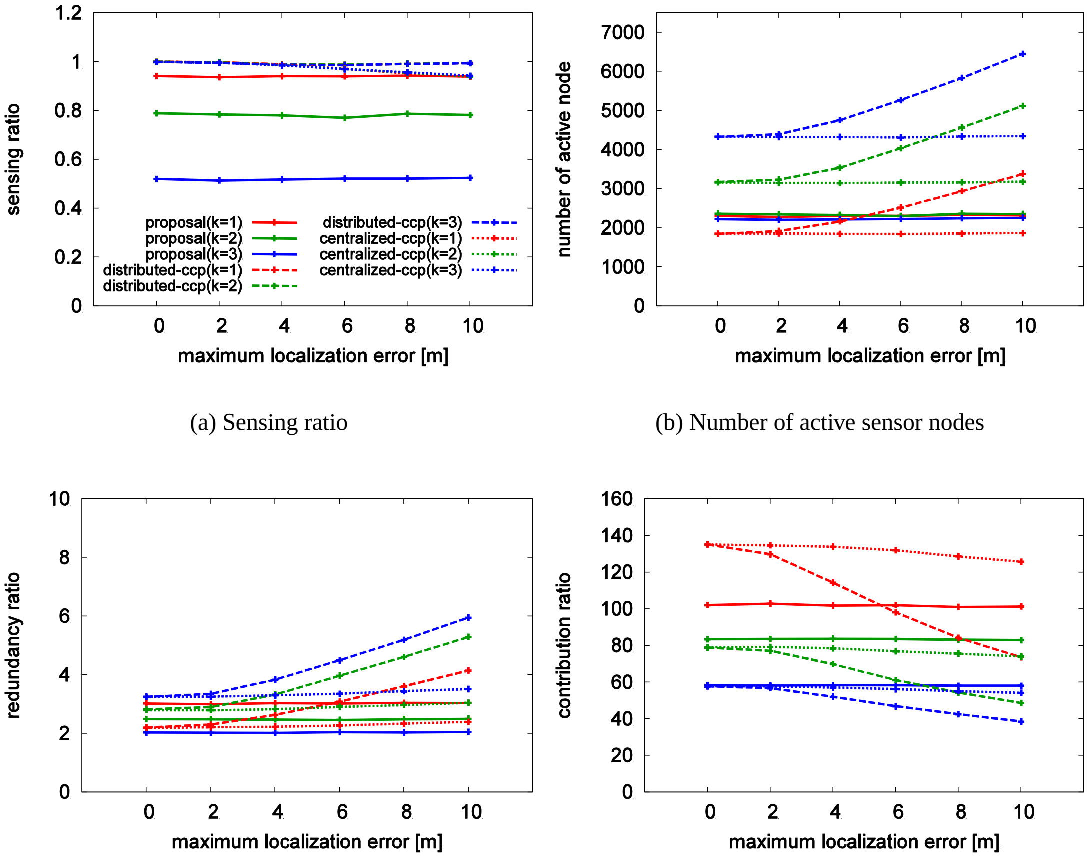
Figure 2.6: Comparison without localization and shape errors

attractor selection model, to find a solution. As such, the size of search space affects the optimality of the found solution. In case of the global activity-based control, the number of combinations of state of sensor node is as large as 2^{10000}. In addition, a state of a sensor node does not influence others very much. Therefore, our proposal often falls into local optimal. However, as the size of sub-area decreases, the sensing ratio of our proposal approaches 1. When the size of sub-area is smaller, the number of sensor nodes per sub-area decreases. As a result, the size of solution space becomes smaller and there appears stronger interdependency among state of sensor nodes. In other word, with the smaller size of sub-area, sensor nodes can find better solution, which has higher sensing ratio and less redundancy ratio. In general, when a sensor node selects the active state, it increases both of the sensing ratio and the redundancy ratio. When the sensing ratio is low, an increase in the sensing ratio increases the activity more than the decrease caused by increased redundancy ratio. It is a reason that there are more active sensor nodes with smaller sub-areas in Figure 2.6(b) for $k = 2$ and 3. On the other hand, when k is 1, even with a small sub-area, it is hard for an additional active sensor node to increase the sensing ratio, which is already high enough. Therefore, the coverage control moves toward reducing the redundancy ratio to increase the activity as shown in Figure 2.6(c).

Regarding the contribution ratio, a smaller sub-area leads to the higher contribution ratio. As shown in Figure 2.6(d), when k is 2 or 3, our proposal can achieve higher contribution ratio than CCP in any size of sub-area. On the contrary, when k is 1, CCP achieves higher contribution ratio than our proposal in almost all size of sub-area. When x_s and y_s are 500 $[\mathrm{m}]$ and k is 1, 2, or 3, about 22 percent of sensor nodes becomes active state. In comparison with CCP, in case of $k = 1$, the number of active sensor nodes is redundant to achieve the perfect 1-coverage ($k = 1$). In addition, due to the low optimality of the found solution, our proposal achieved less sensing ratio than CCP. Because of low sensing ratio and redundant active sensor nodes, our proposal achieves less contribution ratio than CCP. Using smaller sub-areas, our proposal can find better solutions, i.e. achieving higher sensing ratio by less active sensor nodes. In particular, when k is 1 and x_s and y_s are 5 $[\mathrm{m}]$, the number of active sensor nodes in our proposal drops to below CCP, and the magnitude relation of contribution ratio is reversed. In addition, when k is 2 and 3, the number of active sensor nodes increases unlike when k is 1, but the sensing ratio also more increases. Therefore, higher contribution ratio can be achieved as sub-areas become smaller.

2.5.7 Influence of Localization Error

In this section, we compare CCP and two variants of our proposal, i.e. the global activity-based control and the area activity-based control whose sub-area size is set at 10 $[\mathrm{m}] \times 10$ $[\mathrm{m}]$, under the influence of localization error. For the sake of argument about the origin of the error tolerance of our proposal, we show the results of CCP with the center-point control in addition to the results of original CCP. We call the original CCP “distributed-CCP” and the CCP with the center-point



(a) Sensing ratio

(b) Number of active sensor nodes

(c) Redundancy ratio

(d) Contribution ratio

Figure 2.7: Influence of localization error (global activity)

control “centralized-CCP”. In the centralized-CCP, a sink collects the sensing state and location-related information from all sensor nodes and conducts the K_s-Eligibility algorithm for each of the sensor nodes. Then, the determined state is sent back to each sensor nodes. To ignore the influence of shape error, DOI is set at zero. Figures 2.7 and 2.8 summarize results averaged over 10 simulation runs.

Figure 2.7(a) shows the average sensing ratio S of the global activity-based control against the different degree u of localization error. In the figure, it is obvious that neither our proposal nor distributed-CCP is affected by localization error. In our proposal, a sink calculates the activity from collected sensing data. Since the effect of localization error is averaged over the whole region, the derived activity is not seriously affected by localization error. On the contrary, distributed-CCP uses geometric and deterministic algorithm, and as such state selection heavily depends on the accuracy of location information. Nevertheless, distributed-CCP keeps the high sensing ratio. The reason is that localization error and wrong state selection are compensated by the increased number of active sensor nodes and the higher redundancy as shown in Figures 2.7(b) and 2.7(c).

In CCP, localization errors contribute to both of increase and decrease in the number of active nodes. When a sensor node wrongly considers that a neighbor sensor node is far and there is no overlap between their sensing areas by localization error, it is likely to become active to monitor intersections which seem to be uncovered. At the same time, localization error makes a sensor node consider a further neighbor to be located close. Consequently, the affected node is likely to move to the sleep state. In the case of the distributed-CCP, a decision of a sensor node is affected only by neighbor sensor nodes within its communication range. From results of the distributed-CCP in Figure 2.7(b), localization error results in the increase more than the decrease. On the contrary, in the case of the centralized-CCP, a sensor node is further affected by localization error of a sensor node whose actual location is out of its communication range. The actual sensing area of such a distant sensor node does not overlap with the sensing area of the sensor node. Therefore, even if the distant sensor node is considered to be located further by localization error, it does not influence a decision of the sensor node at all. However, when the sensor node considers the distant sensor node is located closer to itself by localization error, it would move to the sleep state. As a result, the number of active sensor nodes becomes smaller than that of the distributed-CCP. Since the increase and decrease are occasionally balanced for uniformly random distribution of sensor nodes, the number of active sensor nodes becomes constant against localization errors.

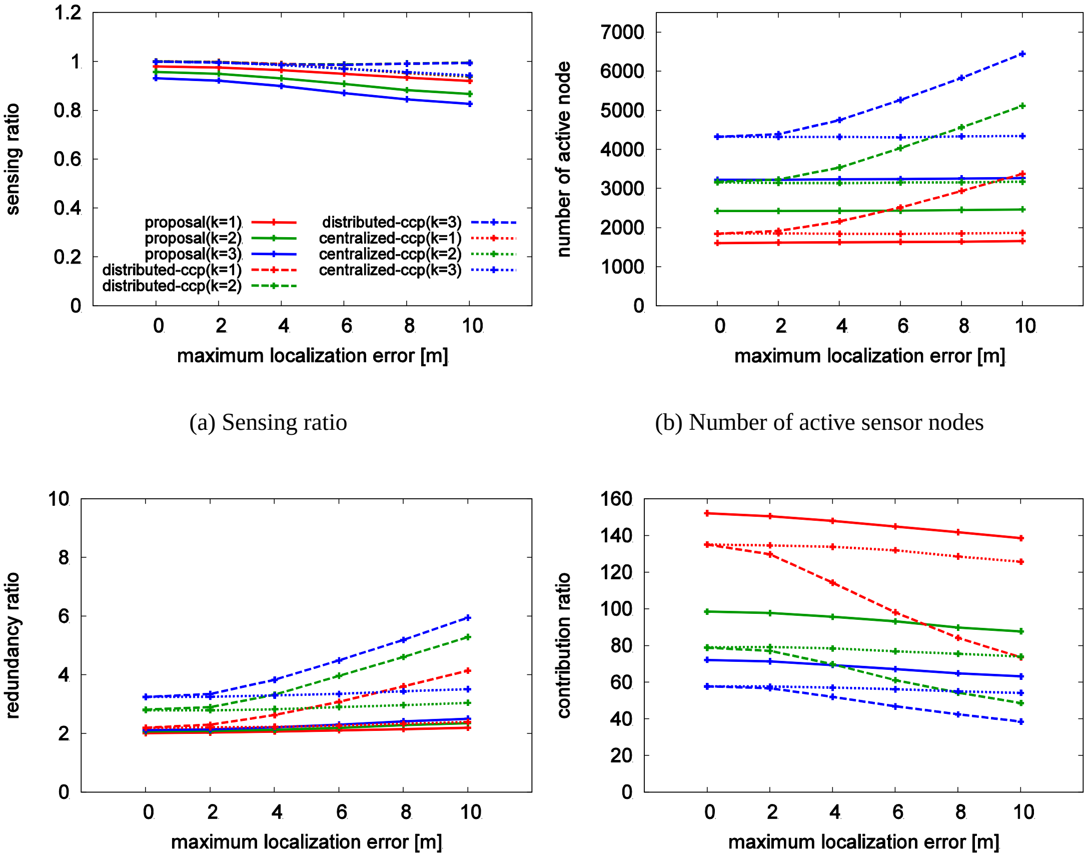
As a result, the redundancy ratio with the centralized-CCP becomes smaller than the distributed-CCP (Figure2.7(c)) and the centralized-CCP is more prone to the localization error than the distributed-CCP in terms of the sensing ratio (Figure2.7(a)). Similarly, in our proposal, the derived activity is not also seriously affected by localization error by averaging error over the whole region, we can achieve the similar performance without increasing the number of active sensor nodes.

To evaluate the efficiency of coverage control, Figure 2.7(d) shows the contribution ratio B

against the different degree of localization error. As can be expected from Figure 2.7(b), the contribution ratio of distributed-CCP decreases as the maximum localization error increases. For example, when an application requires 1-coverage ($k = 1$), the global activity-based control accomplishes more efficient coverage control than CCP with maximum localization error u of 6 meters or more. When an application requires 2 or 3-coverage ($k = 2$ or 3), our proposal always outperforms both the distributed-CCP and the centralized-CCP in terms of the contribution ratio. When we divide the target region into sub-areas whose size is 10 $[\mathrm{m}]$ and apply the area activity-based control, we can achieve higher sensing ratio than the global activity-based control. Especially, in the case of $k = 1$, the similar degree of sensing ratio can be achieved with the smaller number of active sensor nodes. Moreover, the area activity-based control outperforms both distributed-CCP and centralized-CCP in terms of the contribution ratio while the sensing ratio is sufficiently high such as more than 0.8.

However, the sensing ratio gradually decreases as the localization error increases. In comparison with the global activity-based control, the redundancy ratio is lower and the contribution ratio is higher with the area activity-based control (compare Figure 2.7(c) with Figure 2.8(c), Figure 2.7(d) with Figure 2.8(d)). It implies that an uncovered patch has less chance to be covered by a nearby active sensor node than with the global activity-based control. However, even if there is the high localization error, the area activity-based control can achieve the sensing ratio similar to or better than the global activity-based control.

From the above results, we can conclude that our proposals are more robust than distributed-CCP. Although centralized-CCP exhibits the similar robustness in the number of active nodes to our proposal due to the center-point control, our proposal is superior to centralized-CCP. Further discussions will be given in Section 2.6. Although distributed-CCP can maintain sensing ratio close to one against localization error, the number of active sensor nodes considerably increases. It depletes batteries and shortens the lifetime of a sensor network. Although sensing ratio is slightly lower with the area activity-based control than distributed-CCP even without localization error. The number of active sensor nodes does not change much and we can expect the similar lifetime under the influence of localization error, which is quite common in the actual environment. When we consider such applications that do not always require sensing ratio of 100%, e.g. precision agriculture and



(a) Sensing ratio

(b) Number of active sensor nodes

(c) Redundancy ratio

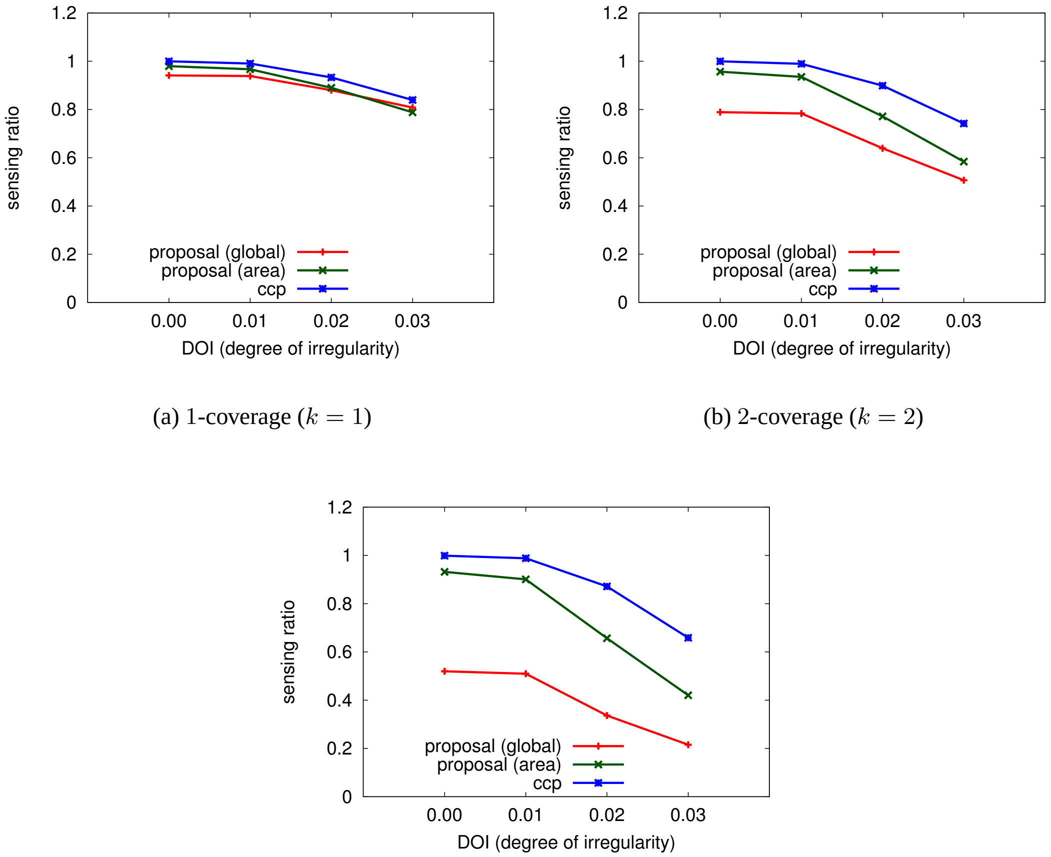
(d) Contribution ratio

Figure 2.8: Influence of localization error (area activity)

environmental monitoring, our proposal is more practical and useful than distributed-CCP.

2.5.8 Influence of Shape Error

Figure 2.9 evaluates the influence of shape error on the sensing ratio under the condition without localization error. As shown in the figure, the sensing ratio decreases independently of protocols and their order does not change against the degree of irregularity. When there are shape errors, a patch considered to be inside the ideal and circular sensing area of an active sensor node is not always inside the actual and irregular sensing area. It leads to decreasing the sensing ratio. On the other hand, even if a patch is covered by a distant active sensor node whose actual sensing area



(a) 1-coverage ($k = 1$)

(b) 2-coverage ($k = 2$)

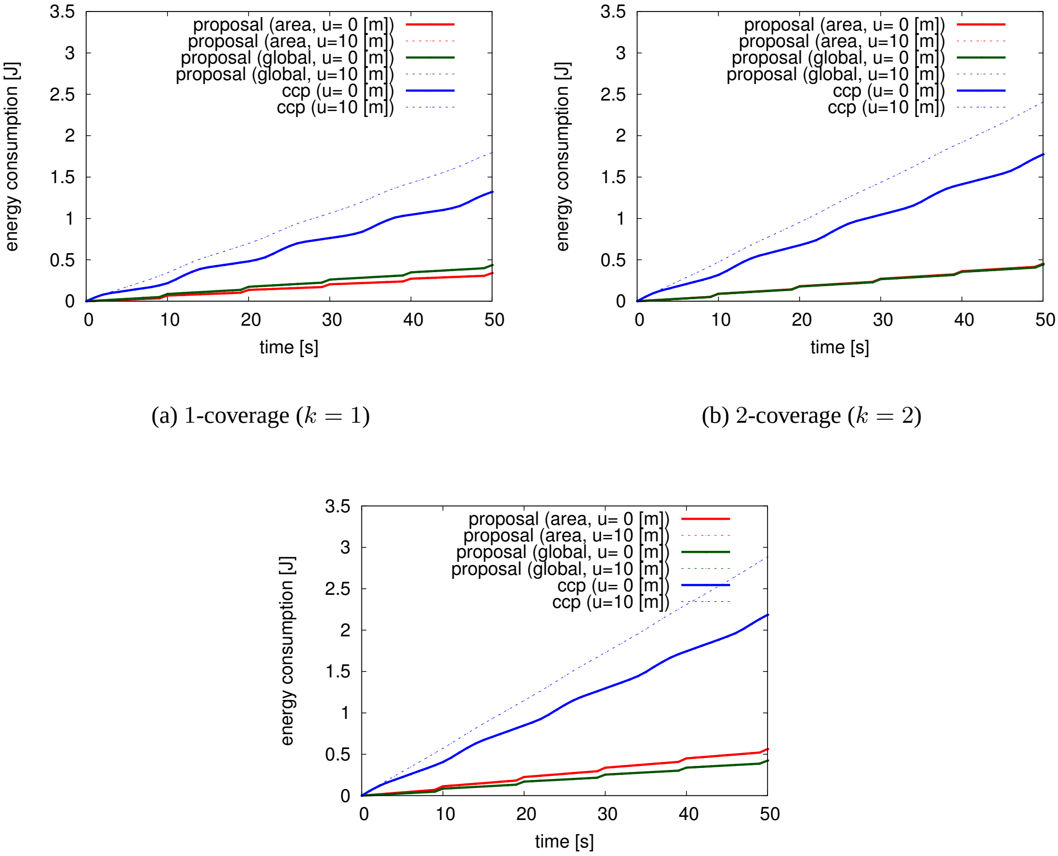
(c) 3-coverage ($k = 3$)

Figure 2.9: Influence of shape error

contains the patch, it does not contribute to the sensing ratio calculated at a sensor node or a sink. It is because another node whose circular sensing area contains the patch decides to become active state for insufficient coverage from a viewpoint of the sensor node and the patch becomes covered anyway. As a result, the shape error causes deterioration of sensing ratio.

2.5.9 Evaluation of Energy Consumption

Finally, we evaluate energy consumption of our proposal and CCP. Figure 2.10 shows the averaged energy consumption per sensor node over 10 simulation runs against time for cases with and



(a) 1-coverage ($k = 1$)

(b) 2-coverage ($k = 2$)

(c) 3-coverage ($k = 3$)

Figure 2.10: Energy consumption

without localization error. Results of our proposal with and without localization error overlap with each other. This is because the number of active sensor nodes does not increase even with high localization error. A reason why the global activity-based control requires more energy than the area activity-based control for $k = 1$, similar energy for $k = 2$, and less energy for $k = 3$ is that it requires more, similar number of, and less active sensor nodes for $k = 1$, 2, and 3, respectively as shown in Figures 2.7 and 2.8. On the contrary, in the case of CCP, localization error depletes more energy for the increased number of active sensor nodes (see Figures 2.7 and 2.8). For 1-coverage ($k = 1$), the amount of energy consumption with localization error becomes 1.35 times as much as that without localization error, whereas the number of active sensor nodes increases by about 1.8-fold.

Independently of the required coverage, it is apparent that our proposal consumes only between one sixth and one third energy of CCP. The primary reason lies in less communication overhead of our proposal. Our proposal does not involve any additional communication among sensor nodes except for dissemination of activity. Therefore, sensor nodes can turn off transceiver modules except for data gathering and feedback dissemination and hold down energy consumption. On the other hand, CCP consumes energy in the listen mode of transceivers for information exchanges and state transitions. To evaluate the K_s-Eligibility and confirm state transition, a sensor node has to keep a transceiver module listening a channel for longer duration than our proposal. Furthermore, CCP requires a larger number of sensor nodes to be active than our proposal when there is a large localization error. Because of the smaller energy consumption, our proposals can accomplish the longer lifetime of sensor network than CCP. For example, although the sensing ratio with the area activity-based control is about 0.8 for $k = 3$ and $u = 10$ [m] as shown in Figure 2.8(a), the lifetime of a sensor network is about six times as long as that with CCP.

2.6 Discussion

As seen in the results of centralized-CCP, center-point control leads to the robustness against localization error in the number of active sensor nodes. This results in the higher contribution ratio of the centralized-CCP than that of the distributed-CCP. Since our proposal adopts a kind of center-point control, where the activity, expressing the degree of coverage of the whole region or each sub-area, is derived at a sink, they have the similar robustness. However, the center-point control alone is not sufficient to explain the reason of higher performance of our proposal than the CCP-based control schemes. A reason that our proposal can outperform the CCP-based schemes by the smaller number of active sensor nodes is in the bio-inspired algorithm. CCP relies on the deterministic and rigorous algorithm, aiming at the perfect coverage. As a result, many sensor nodes are forced to be active to fully fill out the region with active nodes. For example, a sensor node decides to become the active state to cover a small void, whose size is less than $\frac{1}{10}$ of the sensing area. On the contrary, the bio-inspired algorithm is more flexible and relaxed. A single scalar, called the activity, is used to express the degree of coverage of the whole region or each sub-area in a rough and vague manner. In addition, each sensor node decides its state stochastically and autonomously. As such, the number of active sensor nodes is efficiently reduced while leaving some voids are uncovered with our proposal and the sensing ratio is sacrificed to some extent.

2.7 Summary

In this chapter, by adopting the attractor selection model of adaptive behavior of biological systems, we proposed an error-tolerant and energy-efficient coverage control and showed that our proposal can achieve the sensing ratio S of up to 0.98 and prolong the life time of the network up to 6-fold by comparison with CCP.

Chapter 3

Robust Device Assignment Control against Parameter Setting

Many researchers have been attracted by wireless sensor and actuator networks (WSANs) for its wide range of applications. In WSANs, embedded sensors detect and conjecture environmental and personal conditions and actuators provide users with information services and environmental control which are suited for time, place, occasion, and people. Since it apparently is wasteful and redundant to deploy an independent WSAN for each of envisioned applications, building a multi-purpose WSAN consisting of heterogeneous sensors and actuators and sharing them among applications are considered promising. However, we need a mechanism to effectively share available resources among concurrent applications while taking into account application requirements and resources. Although there are several proposals on centralized or deterministic device assignment mechanisms, they suffer from difficulty in designing an appropriate set of rules with fine-tuned parameters. In this chapter, we propose a fully distributed and self-organizing device assignment mechanism by adopting a response threshold model, which imitates division of labors in a colony of social insects. Our proposal does not require deterministic and complicated rules and appropriate device assignment emerges as a consequence of autonomous decision of individual nodes. Through simulation, we confirm the our proposal accomplishes as effective device assignment as an existing deterministic mechanism and our proposal is less sensitive to parameter setting errors.

3.1 Introduction

In recent years, many researchers have been actively working in the field of wireless sensor and actuator networks (WSANs) [2]. A WSAN consists of embedded sensors, e.g. thermometer, hygrometer, and motion sensor, that detect and obtain environmental and personal conditions and actuators, e.g. heater, cooler, buzzer, light, and switch, that control environment and machinery. By distributing nodes with appropriate sensors and/or actuators at appropriate locations in an area, e.g. field, building, and room, and organizing a network by wireless multi-hop communication, a variety of applications can be provided in the area. We hereafter call sensors and actuators '*devices*' and a '*node*' corresponds to an equipment with CPU, memory, wireless transceiver, and one or more devices.

In general, WSANs are constructed and managed in an application-oriented manner to answer specific requirements of an individual application. Therefore, nodes are deployed for a specific application and they are not shared with others. For example, both of WSANs for illumination control and intrusion detection employ nodes with a motion sensor to detect location of people and nodes with a switch to turn on or off a light. Although applications use the same kind of devices in the same way, their WSANs are made of dedicated nodes and independent from each other with current form of deployment. It is apparently redundant and wasteful. Furthermore, an application-oriented deployment requires previous knowledge about the operational environment and careful planning of types and locations of nodes to place. However, it is impossible to predict all events that may occur in the area and make WSANs well prepared for unpredictable events.

Considering above-mentioned issues, interests of researchers are shifting from a *special purpose WSAN* to a *multi-purpose WSAN* where multiple concurrent applications are running over a single WSAN [5]. In a multi-purpose WSAN, heterogeneous nodes are deployed in the area and applications employ those nodes with desired devices. The first challenge exists in the heterogeneity in node architecture [31, 51], which makes application implementation and interoperation of nodes difficult. As an example of solution of the challenge, SOA (Service Oriented Architecture) provides an application with a common interface with nodes having different architecture [31, 52]. Once heterogeneous nodes can be handled through the common interface, another challenge arises in selection of nodes and devices [6, 7]. For example, in starting an intrusion detection application in the area where an illumination control application already exists, is it better to use the node with a motion sensor that the illumination control application is using? If they share the node, other nodes with a motion sensor can sleep and save energy and network bandwidth. A decision on device assignment must be made taking into account a variety of conditions, e.g. the degree of device sharing and the amount of residual energy, and it is not trivial. For this purpose, there are several proposals on dynamic device assignment [53, 54], but they usually employ rule-based mechanisms. As such, as a WSAN becomes large and the number and heterogeneity of applications increase, they will suffer from difficulty in making an appropriate set of rules without contradictions.

In [27], to effectively share available resources among concurrent applications while taking into account application requirements and resources, we presented a basic idea of fully-distributed and self-organizing device assignment mechanism where each node determines whether to offer its own devices to an application or not. Results of preliminary simulation experiments showed that nodes and devices were appropriately selected taking into account the amount of residual energy and the degree of contribution to applications. In this chapter, we improved our mechanism by incorporating with SPAN [55] to efficiently share nodes engaged in message relaying among applications and simplifying a decision making algorithm to make parameter setting easier. In our proposal, the minimum connectivity is maintained by SPAN, where a set of coordinator nodes constructs a forwarding backbone. Once a need for device assignment occurs, a request message is disseminated from a request node of an application to all nodes through a forwarding backbone. On receiving the request, each node determines whether to offer its devices to the application or not. The decision is sent back to the request node through the forwarding backbone.

For autonomous decision making without deterministic if-then type of rules, we adopt a response threshold model [56], which imitates a mechanism of division of labors in a colony of social insects. In a colony, each individual decides to be engaged in a task without any centralized control and the number of workers is dynamically adapted in accordance with the demand of the task. In our proposal, a request message advertised by a request node expresses the demand intensity to stimulate nodes to offer their devices. A request node does not appoint nodes to offer their devices as existing mechanisms do. Instead, each node has has the right to make a decision of device assignment in our proposal. Furthermore, device assignment is performed stochastically at a node. Therefore, our proposal is not deterministic. As such, as will be verified in the chapter, our proposal is less sensitive to parameter setting than the existing mechanism. It implies that our proposal can be used in the area where a variety of applications emerge and their requirements dynamically change.

The remainder of this chapter is organized as follows. First, in Section 3.2, we describe related work. Next, in Section 3.3, we describe application scenario that our proposal assumes. Then, in Section 3.4, we propose a mechanism for a node to autonomously decide whether it assigns its devices to an application or not. In Section 3.5, we show results of simulation to evaluate our proposal and compare it with an existing mechanism. Finally, in Section 3.6, we provide concluding remarks.

3.2 Related Work

The heterogeneity of node architecture makes application implementation and node management difficult. There are several proposals to deal with heterogeneous nodes through the common interface [31,51]. For example, TinySOA, which is based on the concept of SOA [52], allows application developers to write application programs without concerning differences in node architecture, OS, and programming languages.

To share sensors among multiple applications, TinyONet is proposed in [30]. The main focus of TinyONet is reuse of sensing data gathered at a sink. When a sink receive a request for assignment of sensors from an application, it organizes a slice, i.e. a group of virtual sensors. A virtual sensor is a representative of a physical node and it provides an application with cached sensing data. From a viewpoint of an application, a dedicated sensor network is tailored over heterogeneous sensors, from which it can collect required sensing data at the desired frequency. Since a virtual sensor is a cache, it is possible to accommodate multiple applications without putting extra load on a physical sensor network. However, TinyONet assumes that sensing data are collected from all sensor nodes at regular intervals, which should be the minimum among all applications. That is, it is energy and bandwidth consuming. Furthermore, it does not consider actuators.

VSN (Virtual Sensor Network) is another example of a mechanism to overlay application-oriented virtual networks over physical WSANs [57]. In their proposal, a VSN can consist of nodes belonging to different WSANs. A VSN is a single network of tree topology and all messages are exchanged over the tree. Although VSN realizes service-oriented and inter-WSAN overlay networking, it is inefficient to concentrate all traffic at a single tree. Furthermore, when there are multiple concurrent applications, there exist multiple and independent VSN trees in the area.

Regarding on-demand selection of nodes which offer devices or functions to an application, mechanisms for generic role assignment are proposed in [53,54]. In their framework, an application developer injects a role specification to a WSAN through a gateway. A role specification defines roles and rules to assign roles to nodes. Rules are in the form of Boolean expression, i.e. if-the-else statement. A specification is disseminated in a WSAN and a node receiving it decides whether to play a role or not in accordance with the specified rules and its properties. As far as rules are well defined, roles are assigned to appropriate nodes. However, it is not trivial to define an appropriate set of consistent rules to appoint the necessary and sufficient number of nodes with desired properties taking into account a variety of conditions.

Our proposal can assign an appropriate set of nodes and their devices to an application taking into account application's requirements and multiple conditions, i.e. residual energy and resource sharing, without centralized control or deterministic rules.

3.3 Application Scenario

There are applications operating in the area or being introduced on demand. Each application has one or more application servers or control units, such as a home server of a home automation system, which manages the application. However, a server does not have the complete knowledge of the whole WSAN, e.g. type and location of nodes and their devices. We assume that an application consists of a series of *processes,* such as turn on or off the light, and a process is realized by devices embedded in the area. For example, a lighting control application (1) senses user presence and illuminance and (2) turns a room light on or off depending on the situation.

Sharing a sensor among multiple applications is not harmful as far as the sensor can provide them with requested sensor data at the desired precision and frequency. On the other hand, sharing an actuator among multiple applications sometimes causes a problem, which we call ‘actuator contention’. An actuator usually has multiple operations that cannot be performed simultaneously, e.g. turn on and off a light. To solve competition for a device among applications, we assume that each process of an application has a priority value. A priority value is predetermined at implementation and deployment, but it can dynamically change in response to environmental conditions. A process with a higher priority takes precedence over a process with a lower priority. When there is a tie, a decision making algorithm of a WSAN selects a process to assign a device. An assigned device operates as requested by the designated application.

3.4 Our Proposal

The proposal adopts a response threshold model of division of labors in a colony of social insects [56] to accomplish autonomous and fully distributed decision making of nodes on whether to assign embedded devices to an application.

3.4.1 Service Network

An application is realized by devices which are selected by our decision making algorithm. We call a network consisting of nodes contributing to an application a ‘service network’, which is logically laid on a physical WSAN. Nodes constituting a service network are a ‘*request node*’ that initiates organization of the service network, ‘*member nodes*’ that are equipped with devices which can satisfy application requirements, and ‘*relay nodes*’ that deliver messages among a request node and member nodes. In addition, there are two types of member nodes, i.e. ‘*active member nodes*’ and ‘*idle member nodes*’. A role of a node is determined per application and changes in the course of operation. For example, a node is an active member node of application A, a relay node of application B, and a non-member node of application C.

An active member node assigns devices to one or more applications. We call a device which provides a sensing or actuation function to an application an ‘*active device*’. On the contrary,

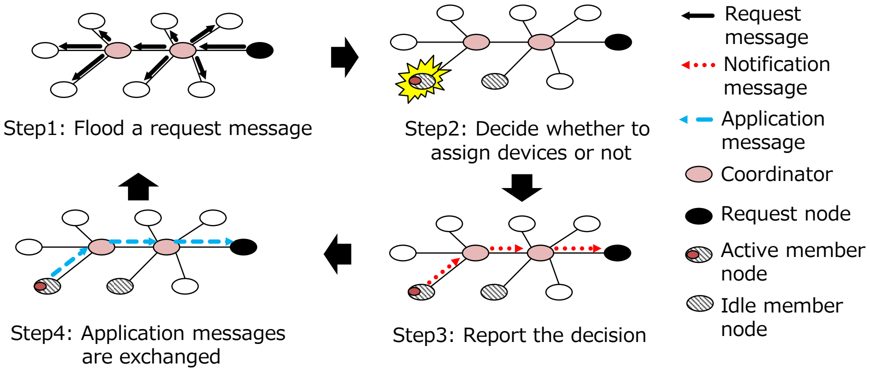

Step1: Flood a request message
Step2: Decide whether to assign devices or not
Step3: Report the decision
Step4: Application messages are exchanged
Request message
Notification message
Application message
Coordinator
Request node
Active member node
Idle member node

Figure 3.1: Overview of our device assignment

an idle member node is equipped with devices which can answer application requirements, but it does not assign them to any application. We call an unassigned device an ‘*idle device*’. An active member node has one or more active devices and some idle devices, but an idle member node has only idle devices. A decision on whether to become an active or not is made by a node taking into account several conditions such as application requirements, the degree that devices are shared among applications, and the residual energy, to efficiently share active member nodes among applications and balance energy consumption of member nodes.

When there is no operating application, no node is active in a WSAN. Device assignment is initiated by a request node, e.g. an application server or a getaway node between a WSAN and an outside application server. We should note here that our proposal can be applied to both of a static and dynamic application. In the case of a static application, a request node is a sink of data of periodic monitoring, for example. In the case of a dynamic application, a node detecting an event becomes a request node, for example.

3.4.2 Basic Behavior

A request node first disseminates a request message which specifies necessary devices and their desired operational mode to all nodes (step 1 in Figure 3.1). The minimum connectivity of a WSAN is maintained by SPAN [55]. SPAN forms the forwarding backbone, which consists of coordinator nodes. A coordinator node is a node which stays awake to maintain connectivity of neighbor nodes. Nodes which are not a coordinator can sleep and communicate with each other through the forwarding backbone when needed. Decision to become a coordinator is made locally by a node. A request message is sent to all nodes in the whole area or nodes in the specified area of interest when location information is available, through the forwarding backbone.

When a node receives a request message, it first examines whether its devices can answer the request. If the node is equipped with such devices, it becomes a member node. Next, a member node decides whether to assign devices to a requesting application or not by using the response threshold model-based decision making algorithm (step 2). Then active member nodes report the decision to the request node by sending a notification message. Nodes where notification messages traverse become a relay node and they adjust the sleep scheduling if necessary (step 3). A member node of a certain application can be a relay node of the same application. A coordinator node is likely to become a relay node. Finally, application messages including sensing and control information are exchanged among active member nodes and a request node through relay nodes until the next timing of periodic dissemination of a request message (step 4).

Above-mentioned steps are repeated while an application is running. A request node can change contents of a request message to perform a different process of the application. It is also possible for a member node to issue a new request message for the application. At the end of an application, a request node stops sending request messages. Those internal values that a node holds for the application is removed when a timer expires without receiving a request message for a predetermined duration.

3.4.3 Internal Values of Nodes

In our proposal, a node maintains a set of information summarized in Table 3.1. Details of each information is given in the followings.

A node is equipped with a set $\mathbf{D}$ of devices. A node also has a set $\mathbf{O_j}$ of operational modes of device $j \in \mathbf{D}$. A set $\mathbf{O_j}$ is represented by the following expression, where $n = |\mathbf{O_j}|$, i.e. the number of operational modes.

$$\mathbf{O_j} = \{\mathrm{mode}_1,\ \cdots,\ \mathrm{mode}_{n-1},\ \mathrm{mode}_n\}$$

Table 3.1: Internal values of a node

Notation	Default	Description
$\mathbf{D}$	ϕ	set of devices
$\mathbf{O_j}$		set of possible operational modes of device j
$\mathbf{S}$	ϕ	set of requirements of applications
$\mathbf{X}$	ϕ	set of $X_{i,j}$
$\mathbf{Y}$	ϕ	set of Y_j
$\mathbf{\Theta}$	ϕ	set of $\theta_{i,j}$
$X_{i,j}$	$false$	boolean flag of assignment of device j to application i
Y_j	default mode	operational mode of device j
$\theta_{i,j}$	5	threshold of assignment of device j to application i

A device cannot operate in different operational modes simultaneously. 'mode_n' is a '***default mode***' of a device and an idle device is in mode_n. When device j is a sensor, a typical set is $\mathbf{O_j} = \{\mathrm{sensing,\ sleep}\}$. In the case of an ON/OFF switch, $\mathbf{O_j} = \{\mathrm{ON,\ OFF}\}$. In general, a default mode is an operational mode where a device and a facility can save energy.

A node also maintains a set $\mathbf{X}$ of $X_{i,j}$, a set $\mathbf{Y}$ of Y_j, and a set $\mathbf{\Theta}$ of $\theta_{i,j}$ for application i and device j ($i \in$ **I** and $j \in \mathbf{D}$), which are used by the response threshold model-based decision making algorithm. **I** is a set of identifiers of application for which a node received a request message. $X_{i,j} \in \{true, false\}$ represents whether device j is assigned to application i ($X_{i,j} = true$) or not ($X_{i,j} = false$). $Y_j \in \mathbf{O_j}$ represents the current operational mode of device j. $\theta_{i,j}$ $(0 < \theta_{i,j} \leq \theta_{max})$ is a threshold representing hesitation of node in assigning device j to application i.

A node maintains a set $\mathbf{S}$ of 7-tuples $(i, j, m, k, h, s_i(t), r_i)$. These values are updated on receiving the t-th request message of application $i \in$ **I**. The identifier i which is unique in the whole network can be generated as concatenation of a node identifier and a sequence number of application it initiates. j is an identifier of a device which application i requires. When application i requires multiple devices, the tuple is generated for each of devices. m is an operational mode which application i request to device j. k is a sequence number of the last request message. h is an identifier of a neighbor node where it received the request message. $s_i(t)$ is the demand intensity representing the degree that the request node wants its request to be satisfied. $s_i(t)$ is calculated by the request node in accordance with the number of devices assigned to application i. Finally, r_i is the priority of application i or its process.

3.4.4 Node Behavior

A request node sends a request message at regular intervals of I_{demand} s. We call an interval between successive emissions of a request message a ‘*round*’. As a simple example, assume a process for periodic data gathering which requires a motion sensor to report the condition at coordinates (x, y) every I_{data} s. In this case, a request message emitted at the t-th round is a pair of attributes in the form $(i, k, s_i(t), r_i)$ and a request body in the form $(\text{motion sensor}, \text{sensing}, (x, y), I_{data})$. The content of a request message can be extended by using an XML-based method [58].

When a node other than a request node of an application receives a request message, it behaves following a flow chart shown in Figure 3.2. First, if a node is a coordinator of SPAN, it forwards the request message to neighbor nodes and it becomes a candidate of a relay node. Next, if it does not have an element of application i in set $\mathbf{S}$, it generates a new 7-tuple element. If the corresponding tuple exists, it is updated. Then, a node examines whether it has a device which satisfy the request. If it has, the node becomes a member node. A member node initializes elements $X_{i,j}$, Y_j, and $\theta_{i,j}$ of application i in sets $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{\Theta}$, respectively, if not exist. If the priority of the application is the highest in all applications in set $\mathbf{S}$ or the requested devices are not assigned, values $X_{i,j}$, Y_j, and $\theta_{i,j}$ are updated by a decision making algorithm explained in Section 3.4.5. Then, if the node assigns device j to application i, the decision is reported to the request node by sending a notification message which contains an identifier of the active member node, an identifier of the application and $X_{i,j}$. A notification message is sent to neighbor h, from which a node received the corresponding request message. Following a reverse path, a notification message reaches the request node.

An assigned device operates in the decided operational mode. In the above example, an active member node sends sensing data of a point (x, y) obtained by a motion sensor to the request node at regular intervals of I_{data} s. Data messages are sent to the request node through the forwarding backbone of SPAN. Every time a member node receives a request message, the above steps

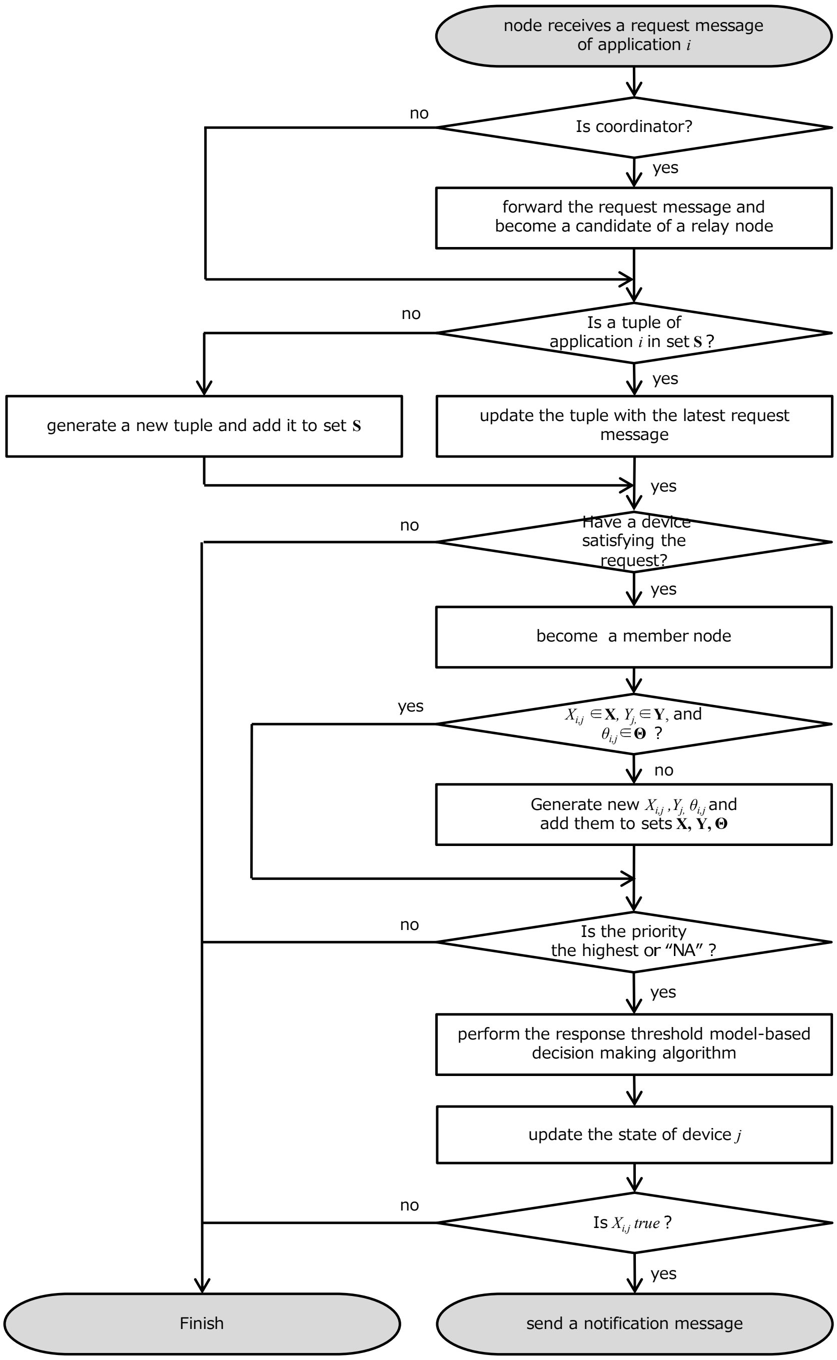

node receives a request message of application i
Is coordinator?
no
yes
forward the request message and become a candidate of a relay node
Is a tuple of application i in set $\mathbf{S}$?
no
yes
generate a new tuple and add it to set $\mathbf{S}$
update the tuple with the latest request message
yes
Have a device satisfying the request?
no
yes
become a member node
$X_{i,j} \in \mathbf{X}$, $Y_{j,} \in \mathbf{Y}$, and $\theta_{i,j} \in \mathbf{\Theta}$?
yes
no
Generate new $X_{i,j}$,$Y_{j,}$ $\theta_{i,j}$ and add them to sets $\mathbf{X}, \mathbf{Y}, \mathbf{\Theta}$
Is the priority the highest or “NA” ?
no
yes
perform the response threshold model-based decision making algorithm
update the state of device j
Is $X_{i,j}$ *true* ?
no
yes
Finish
send a notification message

Figure 3.2: Behavior of a node on receiving a request message

are conducted. If a member node does not receive a request message for E_i s, it considers the corresponding application terminates and it removes corresponding information from the memory.

A request node receives notification messages from active member nodes. In the proposal, a request node uses a scalar value, called the demand intensity, to control the number of active member nodes while leaving decision making to nodes. The demand intensity at the beginning of the $(t + 1)$-th round is calculated from the number of notification messages by the following equation, where the initial demand intensity $s_i(0)$ is set at 0.

$$s_i(t+1) = s_i(t) + \delta_i - N_i(t) \tag{3.1}$$

Here, δ_i $(\delta_i \geq 0)$ is an increasing rate of demand intensity of application i. $N_i(t)$ $(N_i(t) \geq 0)$ is the number of active member nodes which is equal to the number of notification messages stating $X_{i,j} = true$ received in response to the t-th request message. The equation means that, when the number of active member nodes is less than δ_i, the demand intensity gradually increases and the request node requires more member nodes to become an active member node. When active member nodes fail due to node failures, the demand intensity gradually increases, and other idle member nodes additionally become active member nodes. As far as an enough number of alive member nodes remain in the network, the number of active member nodes can recover. Details about robustness against failures are described in the next chapter. On the other hand, when the number is greater than δ_i, the demand intensity gradually decreases and active member nodes become inactive. As such, the parameter δ_i determines the number of active member nodes on convergence and it can be used to adjust the degree that nodes are involved in the process. The updated demand intensity is notified to member nodes by a request message disseminated at the beginning of the next round. Until the next round, a request node exchanges messages with active member nodes.

3.4.5 Response Threshold Model-based Decision Making

It is known that a colony of social insects is divided into two groups of workers and non-workers based on autonomous decision of individuals using a simple rule. A response threshold model is a mathematical model of the division of labors of social insects [56]. We adopt the model as an algorithm for a member node to decide whether it assigns a device to an application or not. For details of the response threshold model, refer to [56]. The size of group is well adjusted based on the task-associated intensity of stimuli [59].

The probability $P(X_{i,j} = false \rightarrow X_{i,j} = true)$ that an idle node ($X_{i,j} = false$) assigns device j to application i is derived by the following equation.

$$P(X_{i,j} = false \rightarrow X_{i,j} = true) = \frac{s_i^2(t)}{s_i^2(t) + A_j \theta_{i,j}^2(t)} \tag{3.2}$$

Here, $s_i(t)$ $(s_i(t) \geq 0)$ is the demand intensity of application i at the t-th round. $\theta_{i,j}$ $(\theta_{max} \geq \theta_{i,j} > 0)$ is a threshold which corresponds to hesitation of the node in assigning device j to application i. The equation is extended from the basic model by introducing variable A_j. A_j $(A_j \geq 1)$ is a variable related to the degree that device j is shared among applications, and the residual energy of the node. Derivation of A_j will be explained in Section 3.4.6.

The probability $P(X_{i,j} = true \rightarrow X_{i,j} = false)$ that an active member node ($X_{i,j} = true$) quits assigning device j to application i is derived by the following equation.

$$P(X_{i,j} = true \rightarrow X_{i,j} = false) = p_j \tag{3.3}$$

Here, p_j $(1 \geq p_j > 0)$ is a constant defined per device. This prevents an active member node from devoting its devices too long and enables rotation of task among member nodes. It further leads to avoidance of redundant device assignment. The average duration that an active member node assigns device j to an application is $1/p_j$ rounds.

Similarly to the basic response threshold model, our proposal also has a mechanism of reinforcement which makes specialists. Threshold $\theta_{i,j}$ is adjusted as follows.

$$\theta_{i,j} = \begin{cases} \theta_{i,j} - \xi_j, & \text{if } X_{i,j} \text{ is } true \\ \theta_{i,j} + \varphi_j, & \text{if } X_{i,j} \text{ is } false \end{cases} \tag{3.4}$$

where ξ_j $(\xi_j > 0)$ and φ_j $(\varphi_j > 0)$ are parameters of the speed of differentiation. With the threshold adjustment, an active member node is more likely to become active again than an inactive member node.

3.4.6 Variable A_j for Device Sharing and Energy Efficiency

In the proposal, for efficient device sharing and balancing energy consumption for a longer lifetime variable A_j $(1 \leq A_j)$ is derived by the following equation from the degree that device j is shared among applications and the residual energy.

$$A_j = (S_j - F_j)^m + \left(\frac{P_{full}}{P_{res}}\right)^n - 1 \tag{3.5}$$

Parameters are summarized in Table 3.2. Here, the first term of the right side is used for device sharing among applications. Variable S_j $(S_j \geq 1)$ represents the number of applications where a node is a member regarding device j. S_j is derived as $S_j = |\{X_{i,j} \in \mathbf{X} \mid i \in \mathbf{L}\}|$ where $\mathbf{L}$ is a set of identifiers of application where a node is a member node. Variable F_j $(F_j \geq 0)$ represents the number of applications where a node is an active member node. F_j is derived as $F_j = |\{X_{i,j} \in \mathbf{X} \mid i \in \mathbf{I}, X_{i,j} = true\}|$ and $F_j \leq S_j - 1$. Exponent m $(m \geq 1)$ influences the sensitivity of the algorithm to the degree that the device is sharing. The second term is used for balancing energy consumption. P_{full}/P_{res} is the ratio of the battery capacity P_{full} $(P_{full} > 0)$ to the residual energy P_{res} $(P_{full} \geq P_{res} > 0)$. Exponent n $(n \geq 1)$ influences the influence of the amount of residual energy on decision making. The third term is used for shifting minimum value of valuable A_j from 2 to 1. Variable A_j becomes smaller and probability $P(X_{i,j} = false \rightarrow X_{i,j} = true)$ becomes higher on a node which is engaged in more applications as an active member node and has more residual energy.

3.5 Performance Evaluation

In this section, we evaluate our proposal through comparison with directed diffusion [32] and our former proposal [27]. We first briefly explain directed diffusion and its extension made for comparison purposes. Then, we will show results of evaluation from viewpoints of efficiency of device assignment and robustness against parameter setting errors.

Table 3.2: Parameters of variable A_j

Notations	Description
m	exponent regulating the sensitivity to the degree of sharing
n	exponent regulating the sensitivity to the residual energy
S_j	number of applications where a node is a member for device j
F_j	number of applications where a node is an active member node for device j
P_{res}	amount of residual energy of a node
P_{full}	total capacity of battery of a node

3.5.1 Directed Diffusion

Directed diffusion is a data-centric information gathering mechanism [32]. A *sink* which corresponds to a request node in our proposal first disseminates an *interest* message. An interest message specifies a required sensing task and a reporting interval. Initially, a reporting interval is set longer than one that an application requires.

When a node receives an interest message, it sets an entry called *gradient*, which consists of the information about a task, an identifier of a link with a neighbor node from which it received the interest message as a precursor, and a report interval specified in the message. If a node can perform the requested sensing task, it becomes a source node, which we call a member node in our proposal, and begins to send data messages at the specified report interval. Data messages reach the sink by following gradients. The first data message is called an *exploratory data* message.

A sink would receive multiple exploratory messages from different source nodes. Among them, it selects one based on a reinforcement rule, for example, to select an exploratory data message received first. Then, the sink sends an interest message, called a *reinforcement* message to a neighbor node from which the selected exploratory data message comes. A reinforcement message is in the same format as an interest message, but it specifies an application-specific reporting interval which is shorter than a reporting interval written in an interest message. A reinforcement message is sent to a source node following gradients in the reverse direction while updating gradients on the route with the new reporting interval. The gradient does not hold information about a source node. Therefore, when there are two or more source nodes in the downstream of the selected neighbor

Table 3.3: Prioritization rule for reinforcement in directed diffusion

		R_{energy}	
		$\geq T_{energy}$	$< T_{energy}$
R_{share}	$\leq T_{share}$	1	3
	$> T_{share}$	2	4

node, a reinforcement message does not necessarily reach a source node which sent the corresponding exploratory data message. A sink keeps sending both of interest messages and reinforcement messages at regular intervals to maintain and update gradients.

3.5.2 Extension of Directed Diffusion

In our proposal, a request node can control the number of devices which contribute to an application by the demand intensity as will be verified in Section 3.5.4, while device assignment relies on an autonomous decision of each node. On the other hand, the number of source nodes is uncontrollable and it would dynamically change in directed diffusion. Since gradient on a node does not have information about either of a sink or a source node, interest messages and reinforcement messages do not always reach the same set of source nodes. Therefore, for comparison, we extended directed diffusion for controlling the number of nodes or devices which contributes to an application as follows.

First, for a sink to obtain information of a source node, we extend a data message to have additional fields for an identifier id_{sink} of a sink, id_{src} of a source node, the amount P_{res} $(P_{res} > 0)$ of residual energy, the battery capacity P_{full} $(P_{full} > P_{res})$, the number M_{dd} $(M_{dd} \geq 1)$ of sinks from which it receives interest or reinforcement messages, and the number N_{dd} $(N_{dd} \geq 0)$ of sinks from which it receives reinforcement messages. Consequently, the extended data message takes the form of [type, data, id_{sink}, id_{src}, P_{res}, P_{full}, M_{dd}, N_{dd}, time-stamp]. Please refer to [32] for details of other fields than those newly introduced. Next, to identify a path between a specific sink and a specific source node, we add a field of an identifier id_{sink} of a sink and id_{src} of a source node to the gradient. First time when a node receives an interest message, it makes a gradient while leaving id_{src} empty. It fills in the field when a data message is received. Consequently, the extended gradient has the form of [type, region, data rate, time stamp, expired-AT, id_{sink}, id_{src}]. While leaving the form of an interest message as it is, i.e. [type, region, interval, time-stamp, expired-AT], we extended the form of a reinforcement message to have a new field for an identifier id_{sink} of a sink node and id_{src} of a source node. As a result, the reinforcement message has the form of [type, region, interval, id_{sink}, id_{src}, time-stamp, expired-AT].

Now based on the extension, we consider a reinforcement rule which takes into account the amount of residual energy and the degree of device sharing. A sink in the extended directed diffusion first disseminates an interest message to all nodes. Next source nodes begin to send data messages. A data message sent by a source node contains information about its energy, P_{res} and P_{full} and its task M_{dd} and N_{dd}. After receiving the sufficient number of data messages, a sink evaluates R_{energy} which is derived as P_{res}/P_{full} and R_{share} which is derived as $(M_{dd} - N_{dd})/M_{dd}$ for each source node. Then, it determines priority of the node in reinforcement. For the sake of simplicity, we use threshold-based prioritization summarized in Table 3.3, where T_{energy} and T_{share} are thresholds. A smaller number has higher priority. For example, if a source node has plenty of energy, i.e. large R_{energy}, and is contributing to many sinks, i.e. small R_{share}, it is the best source node to reinforce. Finally, following an ascending order of priority value, a sink selects the required number N of source nodes and send reinforcement messages to them. In the following, we call a source node which receives a reinforcement message an active source node.

3.5.3 Simulation Setting

We used OMNet++ [60] for simulation. 25 nodes are placed in the area of $25\ \mathrm{m} \times 25\ \mathrm{m}$. 5 nodes, A, B, C, D, and E, among them are located at the edge of the area, while remaining 20 nodes are randomly distributed. Figure 3.3 illustrates an example of node layout where the x and y axes are coordinates and filled circles, open circles, crosses, and triangles indicate nodes. Each line corresponds to a path between an active member node and a request node to exchange application messages.

Nodes are identical in battery capacity, embedded device, and communication capability. They operate on two AA batteries of 3.3 V. Based on the data sheet of MICAz [49], a transceiver module

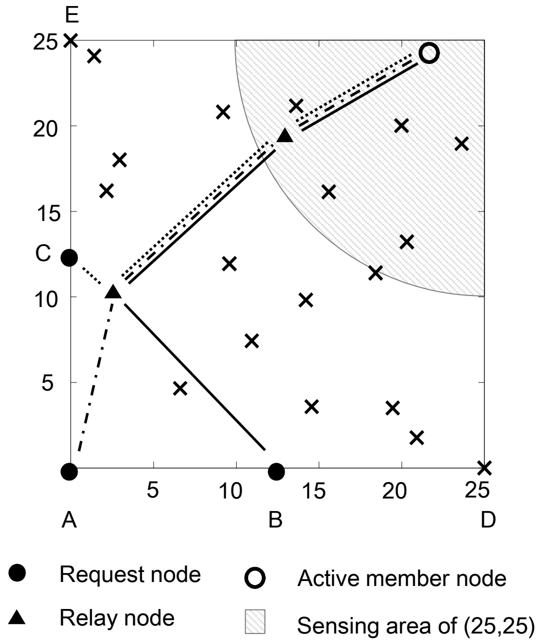


Figure 3.3: Snapshot of a simulation

consumes 18.8 mA in listening a channel and receiving a message, 17.4 mA in transmitting a message, and 0.021 μA in a sleep mode. A node is equipped with a sensing device with identifier j. A sensing device can obtain information about a certain point in the diameter of 15 m. We assume that energy consumption of the device in sensing is negligible in evaluation.

The communication range is 15 m on the IEEE 802.15.4 non-beacon mode MAC/PHY protocol. The length of a request message, an interest message, and a reinforce message is set at 36 byte without 6 byte header. Regarding a notification message, an exploratory message, and a data message, the length is set at 64 byte without a 6 byte header. Parameters used in the simulation experiments are summarized as set A in Table 3.4, which are chosen based on preliminary experiments.

Table 3.4: Parameter setting of performance evaluation

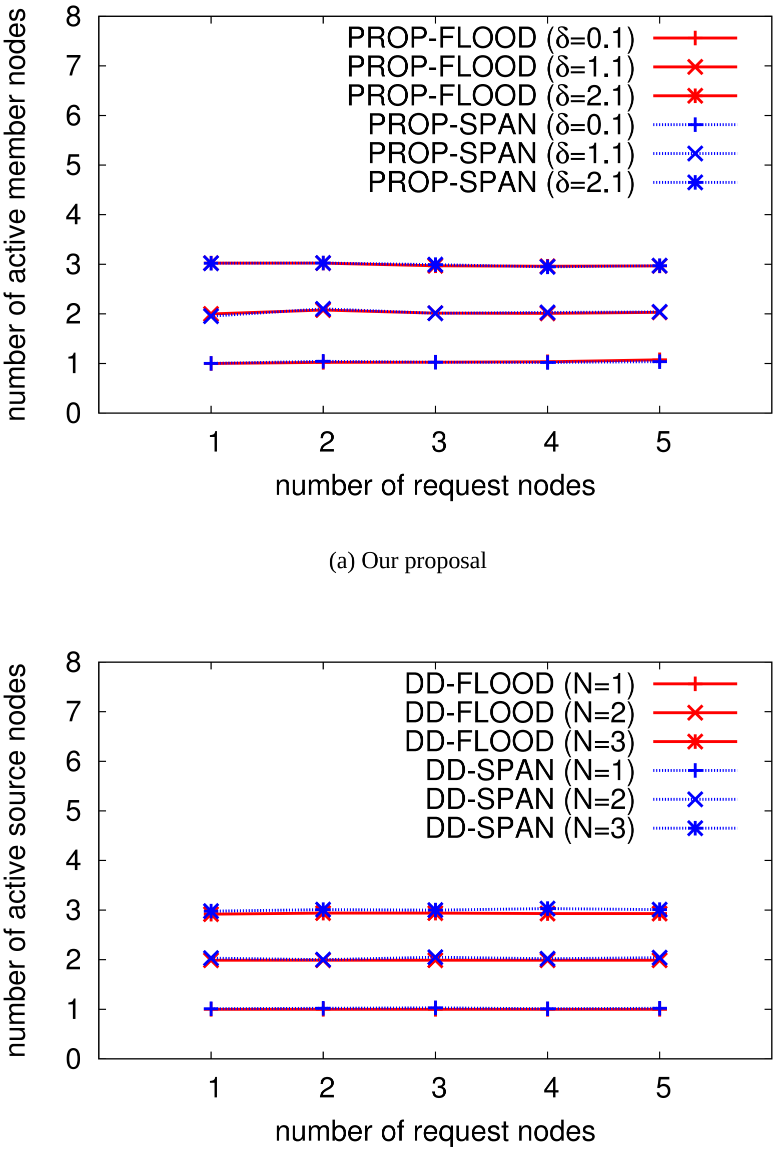
Notation	Description	set A	set B
p_j	probability of quitting task in Eq. (3.3)	0.01	0.01
ξ_j	threshold adaptation parameter in Eq. (3.4)	0.1	0.1
φ_j	threshold adaptation parameter in Eq. (3.4)	1	1
m	influence of degree of sharing in Eq. (3.5)	3	6
n	influence of residual energy in Eq. (3.5)	3	6
L	interval for exploratory data messages	0.5	0.5
T_{share}	threshold of reinforcement rule	0.5	0.1
T_{energy}	threshold of reinforcement rule	0.5	0.9
I_{demand}	interval of request message	10	10

3.5.4 Evaluation of Task Assignment

Since self-organization does not always achieve the optimal result due to its autonomous behavior, in this section, we first verify that our proposal can accomplish as good device assignment as directed diffusion which employs deterministic rules. Evaluation is conducted from a viewpoint of the number of active member nodes and relay nodes.

We configure 5 edge nodes as request nodes of 5 independent applications of the same priority, respectively. All request nodes require the information about the corner point at $(25, 25)$. In the other words, they require assignment of a sensor device within a circular area centered at $(25, 25)$ with radius 15 m, which is illustrated as a shaded quadrant in Figure 3.3. Each request node sends a request message at a regular interval of I_{demand}, which is 10 s in the experiments. Timings of emission of the first request message from request nodes are randomly distributed in 1 sec to avoid collision. We conducted 100 simulation runs for each of 30 combinations of simulation parameters by changing the number of request nodes which send request messages from 1 to 5, the increase rate δ_i from 0.1 to 2.1, and the required number N of active source nodes from 1 to 3.

Figures 3.4 and 3.5 summarize results of the number of active member nodes or active source nodes and the number of relay nodes in the network at the end of a simulation run of 20000 s, respectively. The number of active member or source nodes can be controlled by adjusting δ in our proposal and N in directed diffusion. Each point is an average of 100 simulation runs. In the figures, PROP-SPAN means that our proposal is adopted, while PROP-FLOOD, that is our former



(a) Our proposal

(b) Directed diffusion

Figure 3.4: Number of active member or source nodes

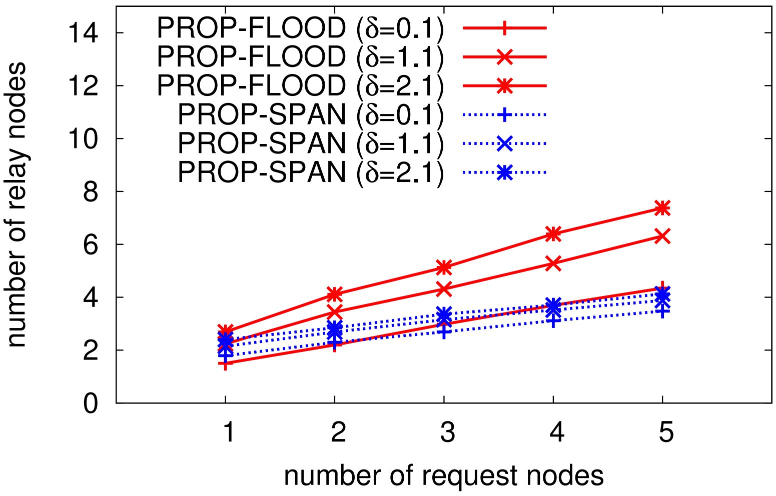
proposal, employs our proposed scheme but without SPAN. Instead, a request node use simple flooding to disseminate a request message in PROP-FLOOD. We also consider combination of directed diffusion with flooding and SPAN as DD-FLOOD and DD-SPAN, respectively.

Figure 3.4(a) shows that both of variations of the proposal, i.e. PROP-SPAN and PROP-FLOOD, keep the number of active member nodes constant even if the number of request nodes increases. Although not shown in the figure, the average number of active member nodes per application is one, two, and three with $\delta = 0.1$, 1.1, and 2.1.

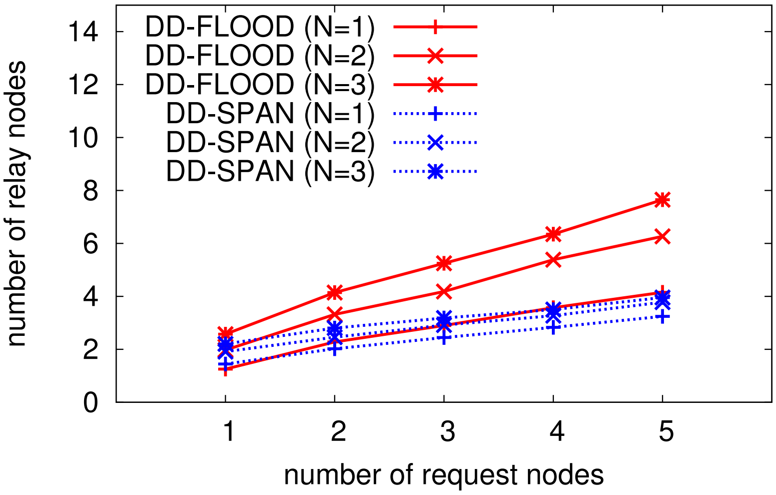
This implies that our proposal can share active member nodes among applications without involving redundant devices. In addition, we also observe that the same δ_i results in the same number of active member nodes independently of the number of applications, while different δ_i results in the different number of active member nodes. When parameter δ_i is 0.1, the number of active member nodes stays 1. If there is no active member node, the demand intensity s_i gradually increases. Consequently, the probability $P(X_{i,j} = false \rightarrow X_{i,j} = true)$ in Eq. (3.2) becomes large at an idle member node. Then some idle member nodes become active and the demand intensity gradually decreases. At the beginning, the number of active member nodes is more than one. However, an active member node eventually becomes idle with probability p_j. If all active member nodes change to idle occasionally, the demand intensity increases again. Through the course, threshold $\theta_{i,j}$ is adjusted on each node. Consequently there appears a node which has the smallest threshold among all. As a result, the number of active member nodes converges to 1. Similarly, when parameter δ_i are 1.1 and 2.1, the number of active member nodes per application converges to 2 and 3, respectively.

In both of variations of directed diffusion, i.e. DD-SPAN and DD-FLOOD, the number of active source nodes is kept constant as shown in Figure 3.4(b). A sink selects the pre-determined number N of source nodes based on the algorithm explained in Section 3.5.2. Then, it sends reinforcement messages to those nodes. By receiving a reinforcement message, R_{share} increases and the priority of the source node becomes higher. Consequently, a source node selected by a sink node is likely to be selected by other sink. As a result, the desired number of source nodes, which are engaged in data reporting at the application-specific rate, are well shared among applications.

Regarding relay nodes, Figure 3.5(a) shows that incorporation with SPAN results in the smaller number of relay nodes than with flooding except for the case of $\delta = 0.1$ and the number of applications is 1. Being incorporated with SPAN, messages traverses the forwarding backbone between a



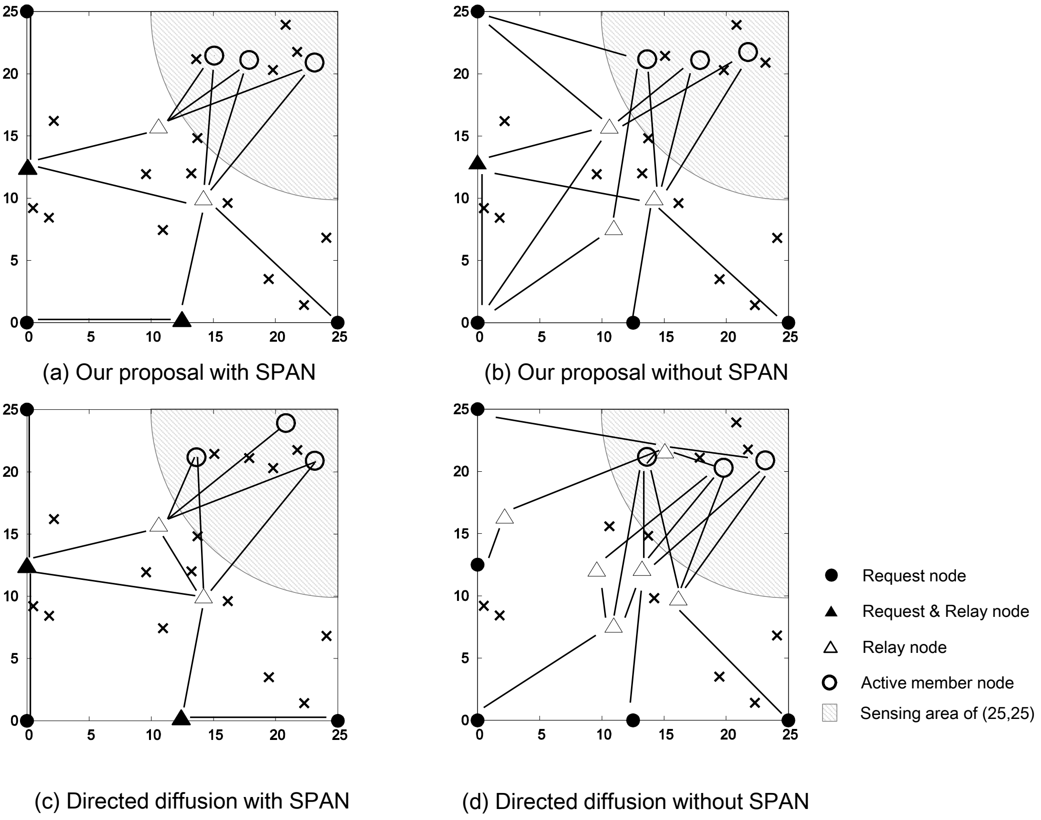
(a) Our proposal



(b) Directed diffusion

Figure 3.5: Number of relay nodes

request node and an active member node. Since there is only one forwarding backbone in the network and it is shared by nodes, a path between them is not necessary the shortest. On the contrary, a message disseminated by flooding follows the shortest path from a request node to a member



(a) Our proposal with SPAN

(b) Our proposal without SPAN

(c) Directed diffusion with SPAN

(d) Directed diffusion without SPAN

Figure 3.6: Snapshots of networks at the end of a simulation run

node. As a result, the average number of relay nodes becomes larger with SPAN than with flooding, whereas the difference is only 0.2. When there are two or more applications or δ is set at a larger value to have two or more active member nodes, the proposal results in the smaller number of relay nodes. With flooding, in the worst case scenario, there exist the same number of independent and disjoint paths between all pairs of a request node and an active member node. On the other hand, the forwarding backbone is always shared among paths with SPAN. This apparently contributes to reduction in the number of relay nodes and the lifetime of a network can be prolonged. As shown in Figure 3.5(b), directed diffusion also benefits from SPAN. Comparing the proposal and directed diffusion, the number of relay nodes is similar, since the number of active member nodes and the number of active source nodes are the same. For example, Figure 3.6 shows snapshots of networks at the end of a simulation run. In the figure, the number of request nodes is 5, $\delta = 2.1$ for our proposal and $N = 3$ for directed diffusion. As shown in the figure, both of our proposal with SPAN and directed diffusion with SPAN involve the minimum number of active source nodes and relay nodes to satisfy five applications. On the other hand, they involve more relay nodes and constructed networks become more complex if they do not adopt SPAN.

From the above results, we can conclude that the proposal can effectively share active member nodes and relay nodes among applications and keep the number of active member nodes constant independently of the number of applications in the current simulation setting. Since directed diffusion is a centralized protocol, where a sink decides source nodes to reinforce with rule-based decision making, it is not surprising that the number of nodes is kept as intended. On the other hand, each member node has the right to make a decision of device assignment in our proposal. Nevertheless, a response threshold model-based decision making algorithm brings results similar to directed diffusion's. That is, our proposal accomplishes self-organizing device assignment which is as optimal as a centralized and deterministic scheme.

3.5.5 Evaluation of Robustness against Parameter Setting

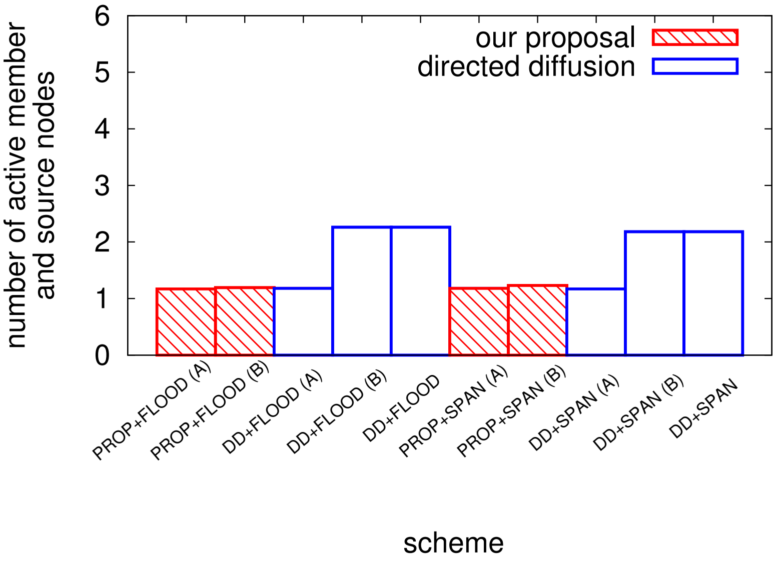
We discuss advantages of the self-organization based proposal over the deterministic and complicated rule-based directed diffusion in regard to the robustness against errors in parameter setting. In this section, we assume that three applications are operating in the area. Three nodes at coordinates $(0, 0)$, $(12.5, 0)$, and $(0, 12.5)$ are their request nodes as illustrated in Figure 3.3. The requested number N of active source nodes per application is set at 1 and parameter δ_i of the proposal is set at 0.1. To make nodes heterogeneous in energy condition, the initial residual energy of each node is set at random value ranging from 25% to 80%.

In general, a response threshold model is less sensitive to parameter setting similarly to other bio-inspired mechanisms [61]. To confirm this, we use the different parameter setting, i.e. set B in Table 3.4, we changed m and n in Eq. (3.5) from 3 to 6. With larger m and n, it prevents a member node from becoming an active member node and getting the sufficient number of active member nodes become difficult. To examine the robustness of a decision making algorithm against parameter setting, we also change thresholds T_{share} and T_{energy} from 0.5 to 0.1 and 0.9, respectively. Since R_{energy} ranges from 0.75 to 0.80 from the beginning of a simulation run and R_{share} is always equal to or larger than $(3-2)/3 \simeq 0.33$. All source nodes have the same priority of 4 in Table 3.3. That is, directed diffusion cannot take into account the heterogeneity of nodes in device assignment.

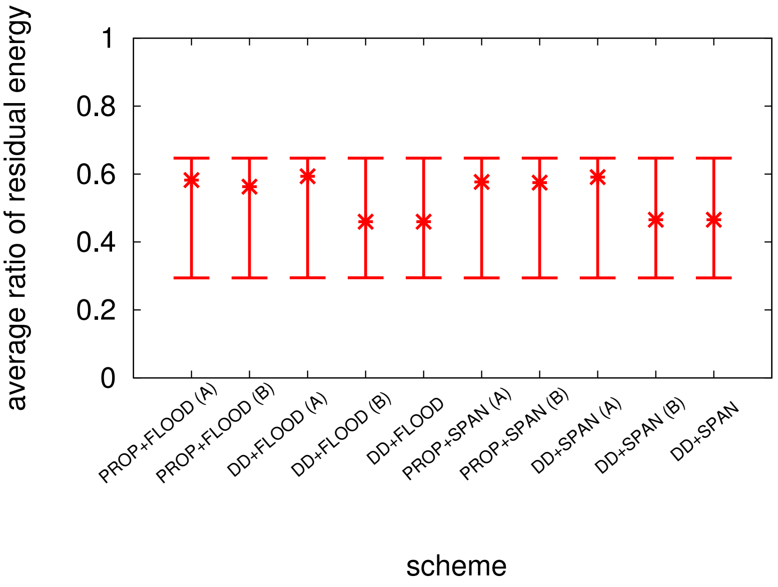
Simulation results averaged over 100 runs are depicted in Figure 3.7. We considered 10 different schemes in the experiments. 'DD+FLOOD' and 'DD+SPAN' without an identifier of parameter setting corresponds to the original directed diffusion without and with SPAN. In Figure 3.7(a), the x-axis indicates schemes and the y-axis is the average number of active member nodes or active source nodes at 20000 s. In Figure 3.7(b), the y-axis indicates the averaged ratio of residual energy of member nodes. Each cross show an average ratio of residual energy of active member nodes. The top and bottom of each bar indicates the maximum and minimum ratios, respectively.

As shown in Figure 3.7(a), from a view point of the number of active member nodes, the change of parameter setting does not have strong impact on our proposal. On the contrary, directed diffusion suffers from an error and the number of active source nodes increases. This is because, all source nodes are categorized into one priority class. As a result, DD+FLOOD(B) and DD+SPAN(B) become identical to DD+FLOOD and DD+SPAN, respectively, where devices are not effectively shared among applications. In appropriate threshold setting further results in the unbalanced energy consumption as shown in Figure 3.7(b). We also observe that our proposal, independently of parameter setting errors, can achieve the same level of energy saving as the extended directed diffusion with appropriate thresholds.

From the above results, we can conclude that our self-organizing device assignment is less sensitive to errors in parameter setting and, as such, to operational conditions, than directed diffusion, while achieving as efficient device assignment as directed diffusion with appropriate parameters does.



(a) The number of active member/source nodes



(b) The residual energy

Figure 3.7: Rubustness of our proposal against parameter setting

3.6 Summary

In this chapter, we proposed a self-organizing device assignment mechanism for a multi-purpose WSAN. Results of simulation support the proposal, but there still remains room for further evaluation and improvement. When there is actuator contention among two applications with the same priority, our proposal first assigns an actuator to one application and then to another application by being stimulated by the increased demand intensity of the latter. The frequency that a device is assigned depends on δ_i, i.e. an increasing rate of demand intensity. In other words, δ_i is another parameter with which an application can control device assignment. We need to confirm this by conducting additional experiments. We also need to evaluate the scalability and adaptability of the proposal, which are inherent characteristics of self-organizing systems.

Chapter 4

Robustness of Division of Labors-based Network Control against Node Failures

It is an emerging research area to adopt bio-inspired algorithms to develop a new information network. Despite their benefits, i.e. high robustness, adaptability, and scalability, none can clearly identify the range of application of a bio-inspired algorithm to challenging issues of information networks. To tackle the problem and accelerate researches in this area, we need to understand characteristics of bio-inspired algorithms from the perspective of network controls. For example, characteristics of bio-inspired mechanisms under non-negligible failure-prone and lossy environment should be investigated. In this chapter, we study a response threshold model as an example of bio-inspired algorithms first under the assumption that individuals are fully connected each other. We build its analytical model and conduct mathematical analysis in order to show the robustness against individual failures in lossy environment and discuss the condition and the time required for the recovery from individual failures in lossy environment. Next, we assume multi-hop communication as in information networks. Individual failures in this case cause isolated individuals that are alive but do not contribute to performing a task. Through analysis, we confirm that the number of workers recovers from individual failures even if dead individuals and isolated individuals exist. Moreover, the recovery time does not diverge as more individuals die.

4.1 Introduction

Information and communication networks must be more robust, adaptive, and scalable against ever-increasing size, complexity, and dynamics. To this end, many researchers focus on self-organizing behavior of biological systems, where a global pattern emerges from mutual and local interactions among simple individuals, and develop novel control mechanisms in a biologically-inspired manner [62].

Bio-inspired control mechanisms not only mimic behavior of biological organisms but are based on nonlinear mathematical models, which explain or reproduce biological self-organization. Examples include an ant colony optimization model for foraging behavior of ants, a pulse-coupled oscillator model for synchronized flashing of fireflies, and a reaction diffusion model for pattern formation on body surface [8–10]. Since bio-inspired mathematical models, which we call *bio-models* in this chapter, are shown to have excellent self-organizing characteristics, network control mechanisms based on bio-models are expected to be robust, adaptive, and scalable [15, 63–65]. Successful attempts published in literatures support this expectation, and there is no doubt about the usefulness of bio-models [66, 67].

However, bio-models are not necessarily versatile. One can achieve the best performance in one environment while it is useless in other. Furthermore, a bio-inspired network control mechanism often experiences a variety of perturbation such as loss of information and failure of nodes, which original bio-models do not consider. As a result, it would fail in providing intended results in the actual environment. Therefore, we need deep understanding of bio-models especially in regard to their fundamental limits and applicability to network controls suffering from perturbation. For example in [14], they evaluated the influence of delay on a bio-inspired synchronization mechanism adopting the pulse-coupled oscillator model and showed that synchronization error comparable to propagation delay would occur at the worst cases. In [12], it is shown that reaction-diffusion based autonomous pattern formation can tolerate information loss of as high as 35%.

In this chapter, by taking a response threshold model [56] as an example, we analyze the robustness of a response threshold model-based network control against information loss and individual failures. The response threshold model is a mathematical model of division of labors in a colony of social insects. It has been applied to a variety of self-organizing network controls, such as task allocation [68], topology control [69], routing [11], and cache replacement [61].

First, we investigate the robustness of the response threshold model against individual failures in lossy environment under an assumption that individuals are fully connected each other, that is, organize a mesh network. We build its analytical model, which takes individual failures and information loss into account, because the original model assumes that there is no dead individual and no information loss. Then, we analyze characteristics of the steady state and the transient state toward the steady state. As to the steady state, we investigate an influence from the existence of dead individuals in a colony on the distribution of the number of workers. As to the transient state toward the steady state, we investigate conditions and time required for a colony to recover from individual failures in lossy environment. We additionally clarify the influence of tunable control parameters on the recovery time as a guideline of parameter tuning.

Next, we conduct simulation experiments to study the response threshold model under an assumption that individuals are connected in a multi-hop manner as in information networks. In this case, individual failures cause isolated individuals that are alive but do not contribute to performing a task. We evaluate the influence of dead individuals and isolated individuals through simulation experiments.

The rest of this chapter is organized as follows. First, in Section 4.2, we briefly describe the response threshold model. Next in Section 4.3, we build an analytical model which takes information loss and individual failures into account and in Section 4.4 we conduct mathematical analysis to discuss the robustness of the model. Then, in Section 4.5, we evaluate the influence of individual failures and information loss in a multi-hop network through simulation experiments. Finally, we conclude this chapter in Section 4.6.

4.2 Mathematical Model of Division of Labors

A response threshold model is a biological mathematical model which imitates adaptive division of labors in a colony of social insects [56]. A colony is autonomously divided into two groups of workers and non-workers based on autonomous decision of individuals. The size of each group

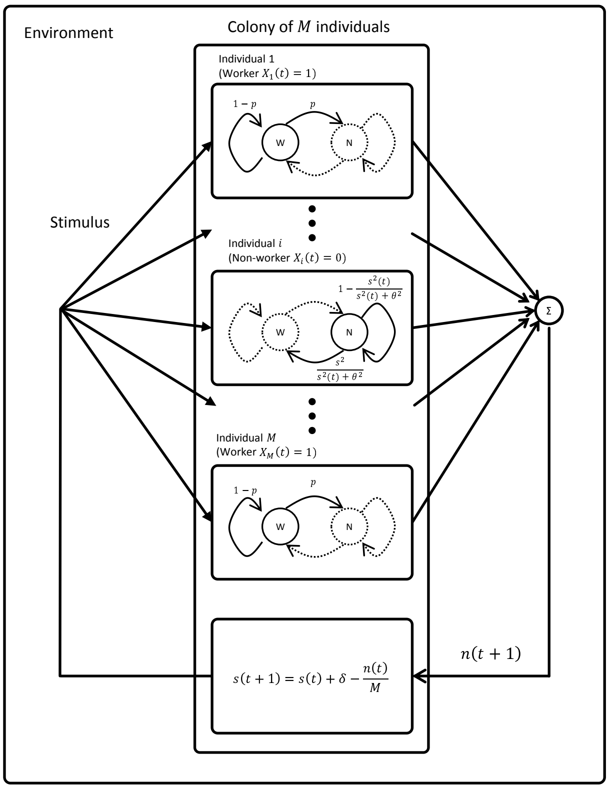

Environment
Colony of M individuals
Individual 1
(Worker $X_1(t) = 1$)
$1 - p$
p
W
N
Stimulus
Individual i
(Non-worker $X_i(t) = 0$)
$1 - \frac{s^2(t)}{s^2(t) + \theta^2}$
W
N
$\frac{s^2}{s^2(t) + \theta^2}$
Individual M
(Worker $X_M(t) = 1$)
$1 - p$
p
W
N
Σ
$n(t+1)$
$s(t+1) = s(t) + \delta - \frac{n(t)}{M}$

Figure 4.1: Overview of the original response threshold model

is well adjusted to meet the task-associated demand or stimulus intensity. In the following, we assume that there is one task to be performed in the colony for the sake of simplicity of explanation. Under this assumption, an overview of the original response threshold model can be illustrated as Figure 4.1.

Let $s(t)$ (≥ 0) be the task-associated stimulus intensity at time t. The stimulus intensity gradually increases over time and decreases as individuals work by the following discrete equation:

$$s(t+1) = s(t) + \delta - \frac{n(t)}{M}. \tag{4.1}$$

Here δ $(0 \leq \delta \leq 1)$ is the increasing rate of the stimulus intensity. $n(t)$ is the number of workers at time t. M (> 0) is the total number of individuals which are capable of performing the task. Based on the model, the stimulus intensity becomes stable when the ratio of workers in the colony is equal to δ. The model can easily be extended to consider the absolute number of workers not the ratio by appropriately defining δ.

By being stimulated by the stimulus, each individual stochastically decides whether to perform the task or not at time $t+1$. That is, the stimulus is a global parameter shared among individuals in the original model. The state of individual i at time t is denoted as $X_i(t)$ $(\in \{0,1\})$, where 0 and 1 mean a non-worker and a worker, respectively. The probability $P\left(X_i\left(t+1\right)=1 \mid X_i\left(t\right)=0\right)$ that non-worker i at time t becomes a worker and begins performing the task at time $t+1$ is given by the following equation:

$$P\left(X_i\left(t+1\right)=1 \mid X_i\left(t\right)=0\right)=\frac{s^2(t)}{s^2(t)+\theta_i^2(t)}, \tag{4.2}$$

where $\theta_i(t)$ (>0) is a threshold value at time t, which corresponds to hesitation of individual i in performing the task. Therefore, an individual with a smaller threshold value is more likely to become a worker than one with a larger threshold value.

The probability $P\left(X_i\left(t+1\right)=0 \mid X_i\left(t\right)=1\right)$ that a worker at time t quits working at time $t+1$ is given by a constant p $(0 \leq p \leq 1)$.

$$P\left(X_i\left(t+1\right)=0 \mid X_i\left(t\right)=1\right)=p \tag{4.3}$$

Quitting a task at the constant rate enables rotation of the task among individuals, that is, work-sharing or load balancing. Given p, the average duration that an individual performs the task becomes $1/p$.

When the number of non-workers occasionally increases by addition of newcomers or the number of workers decreases for sudden death, the stimulus intensity eventually increases. The increased stimulus makes individuals with high threshold value turn into workers. Eventually, the ratio of workers is maintained at around the equilibrium point determined by the increasing rate δ. Figure 4.2 illustrates an example of temporal variations of the number of workers and the stimulus intensity, where $M=20$, $\delta=0.25$, $\theta=1$, and $p=0.01$. Initially, $n(0)=0$ and $s(0)=0$. As shown in the figure, the number of workers first increases to 9 and then converges to $5=\delta\times M$.

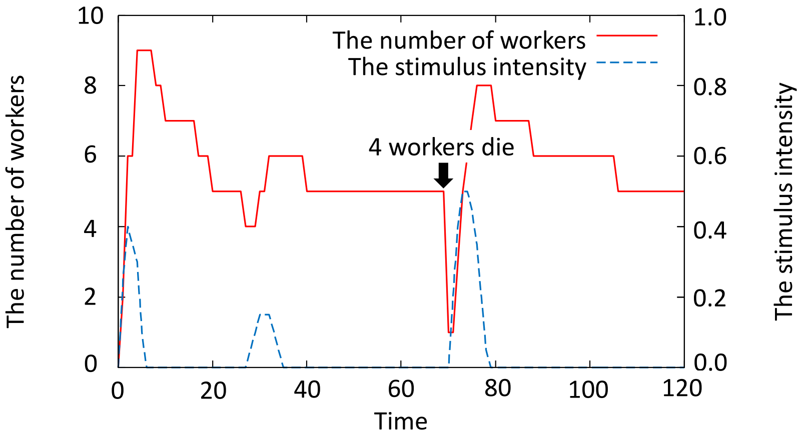


Figure 4.2: Temporal variations of the number of workers and the stimulus intensity

Then at time 70, 4 workers die. The stimulus intensity instantaneously increases, and non-workers become workers to compensate dead workers. The number of workers once increases to 8 and finally converges to 5 again.

4.3 Analytical Model Considering Failures and Information Loss

4.3.1 Overview

Our main objective in this chapter is to analyze the robustness of the response threshold model-based network control against node failures in lossy environment. The original response threshold model however assumes that there is no dead individual and no information loss, and it is not clear yet how individual failures and information loss affect the steady and transient states. In addition, from the standpoint of application to network control, we are rather interested in the collective dynamics of a colony such as the number of workers, but the dynamics is not formulated in [56].

We extended the original model to have dead individuals and information losses as shown in Figure 4.3. In the figure, there are D dead individuals. Information about a state X_i of individual i is lost with a probability q_w $(0 \leq q_w \leq 1)$ and thus it is not taken into account in derivation of $s(t+1)$. On the other hand, individual i is not stimulated with a probability q_s $(0 \leq q_s \leq 1)$ and does not change its state. For the sake of analysis of the dynamics of the colony, we consider a

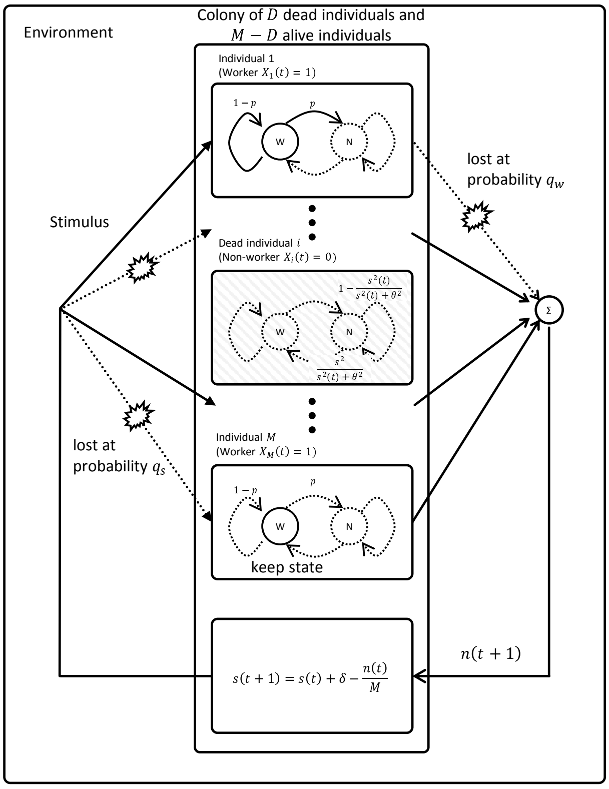

Environment
Colony of D dead individuals and $M-D$ alive individuals
Individual 1
(Worker $X_1(t)=1$)
$1-p$
p
W
N
lost at probability q_w
Stimulus
Dead individual i
(Non-worker $X_i(t)=0$)
$1-\frac{s^2(t)}{s^2(t)+\theta^2}$
W
N
$\frac{s^2}{s^2(t)+\theta^2}$
Σ
lost at probability q_s
Individual M
(Worker $X_M(t)=1$)
$1-p$
p
W
N
keep state
$n(t+1)$
$s(t+1)=s(t)+\delta-\frac{n(t)}{M}$

Figure 4.3: Overview of the extended response threshold model

continuous time model of the stimulus intensity S_t and the number N_t of workers at time t. In the following, for each of S_t and N_t, we first define the random valuable, next derive the probability distribution, and finally obtain the dynamics of an estimated value.

4.3.2 Random Variable S_t of Stimulus Intensity

First we define a random variable $S_{t+\Delta_t}$ ($\in \{0, 1/M, 2/M, \ldots, S_{max}\}$) of the stimulus intensity at time $t + \Delta_t$, where a constant S_{max} is a maximum value of the stimulus intensity. Based on Equation (4.1), the random variable $S_{t+\Delta_t}$ can be defined as follows:

$$S_{t+\Delta_t} = S_t + \Delta_t \left(\delta - \frac{\sum_{i=1} C_t^i X_t^i}{M} \right), \tag{4.4}$$

where a constant Δ_t is infinitesimal time. A constant δ $(0 \leq \delta \leq 1)$ is an increasing rate of the stimulus intensity. A constant M (> 0) is the number of individuals in a colony. A random variable X_t^i $(\in \{0, 1\})$ corresponds to the state of individual i at time t, where 0 is for a dead individual or a non-worker and 1 is for an alive worker. C_t^i is a variable to reflect information loss, which is 0 with a probability of q_w and 1 with a probability of $1 - q_w$.

Next, we derive a probability distribution $P(S_{t+\Delta_t} = s^+)$ that the stimulus intensity $S_{t+\Delta_t}$ becomes s^+ at time $t + \Delta_t$ from $S_t = s$ and $N_t = n$. The conditional probability $P(S_{t+\Delta_t} = s^+ | S_t = s, N_t = n)$ is derived by the next equation:

$$P(S_{t+\Delta_t} = s^+ | S_t = s, N_t = n) = \sum_{l=0}^{l=n} B(n, l, q_w) K(s^+, C(s, n, l)). \tag{4.5}$$

The function $B(n, l, q_w)$ is a probability mass function of the binomial distribution:

$$B(n, l, q_w) = \binom{n}{l} q_w^l \left(1 - q_w\right)^{n-l}, \tag{4.6}$$

The function $K(s^+, C(s, n, l))$ is 1 when $s^+ = C(s, n, l)$ and 0 otherwise. The function $C(s, n, l)$ is the stimulus intensity when state information of l workers is lost. It is defined as follows:

$$C(s, n, l) = \begin{cases} 0, & \text{if } l < n - M(s + \delta) \\ S_{max}, & \text{if } l > n - M(s - S_{max} + \delta) \\ s + \delta - \frac{n-l}{M}, & \text{otherwise} \end{cases} \tag{4.7}$$

Finally, we derive the dynamics of the expected value of the stimulus intensity $S_{t+\Delta_t}$. From Equation (4.4), the expected value is derived by the next equation:

$$\mathbb{E}[S_{t+\Delta_t}] \quad = \quad \mathbb{E}[S_t] + \Delta_t \left(\delta - \frac{\sum_{i=1} \mathbb{E}[C_t^i X_t^i]}{M} \right). \tag{4.8}$$

To derive the dynamics of the expected value, we first consider the conditional expected value

$\mathbb{E}[S_{t+\Delta_t}|N_t = n]$ which is derived by the next equation:

$$\mathbb{E}[S_{t+\Delta_t}|N_t = n] \;=\; \mathbb{E}[S_t|N_t = n] + \Delta_t\left(\delta - \frac{1-q_w}{M}n\right), \tag{4.9}$$

where we use $\mathbb{E}[C_t^i X_t^i | X_t^i = 1] = 1 - q_w$ and $\mathbb{E}[C_t^i X_t^i | X_t^i = 0] = 0$. Let $P(N_t = n)$ $(0 \leq P(N_t = n) \leq 1)$ be a probability that the number N_t of workers at time t is n. Then, with $P(N_t = n)$ and Equation (4.9), the expected value $\mathbb{E}[S_{t+\Delta_t}]$ is derived by the next equation:

$$\begin{aligned}\mathbb{E}[S_{t+\Delta_t}] &= \sum_{n=0} P(N_t = n)\, \mathbb{E}[S_{t+\Delta_t}|N_t = n] \\ &= \mathbb{E}[S_t] + \Delta_t\left(\delta - \frac{1-q_w}{M}\mathbb{E}[N_t]\right). \end{aligned}\tag{4.10}$$

Following the definition of differential, the dynamics of the expected value $\mathbb{E}[S_t]$ can be derived as follows:

$$\begin{aligned}\frac{d\mathbb{E}[S_t]}{dt} &= \lim_{\Delta_t \to 0} \frac{\mathbb{E}[S_{t+\Delta_t}] - \mathbb{E}[S_t]}{\Delta_t} \\ &= \delta - \frac{1-q_w}{M}\mathbb{E}[N_t]. \end{aligned}\tag{4.11}$$

4.3.3 Random Variable N_t of the Number of Workers

First we define a random variable $N_{t+\Delta_t}$ $(\in \{0, 1, \ldots, M\})$ of the number of workers at time $t + \Delta_t$. The random variable $N_{t+\Delta_t}$ can be defined as follows:

$$N_{t+\Delta_t} = N_t + \Delta_t\left(-Q_t + B_t\right), \tag{4.12}$$

where a constant Δ_t is infinitesimal time. A random variable Q_t $(\in \{0, 1, \ldots, M\})$ is the number of workers at time t, which change to non-workers at time $t + 1$. A random variable B_t $(\in \{0, 1, \ldots, M\})$ is the number of non-workers at time t, which change to workers at time $t + 1$.

Next, we derive a probability distribution $P(N_{t+\Delta_t} = n^+)$ that the number $N_{t+\Delta_t}$ of workers becomes n^+ at time $t + \Delta_t$ from $S_t = s$ and $N_t = n$. The conditional probability $P(N_{t+\Delta_t} =$

$n^+|S_t = s, N_t = n)$ is defined as follows:

$$P(N_{t+\Delta_t} = n^+|S_t = s, N_t = n) = \sum_{x=0}^{x=\min(n,n^+)} B(n, x, P_{1,s}^{work})B(M - D - n, n^+ - x, P_{0,s}^{work}), \tag{4.13}$$

where $P_{1,s}^{work}$ is formulated as $P_{1,s}^{work} = P(X_{t+\Delta_t}^i = 1|X_t^i = 1, S_t = s) = \Delta_t \times \{q_s + (1-q_s) \times (1-p)\}$, and $P_{0,s}^{work}$ is formulated as $P_{0,s}^{work} = P(X_{t+\Delta_t}^i = 1|X_t^i = 0, S_t = s) = \Delta_t \times (1-q_s) \times \frac{s^2}{s^2+\theta^2}$.

Finally, we derive the expected value of the number $N_{t+\Delta_t}$ of workers. From Equation (4.12), the expected value is formulated by the next equation:

$$\mathbb{E}[N_{t+\Delta_t}] = \mathbb{E}[N_t] + \Delta_t \left(-\mathbb{E}[Q_t] + \mathbb{E}[B_t]\right). \tag{4.14}$$

To derive the dynamics of the expected value, we first consider the conditional expected value $\mathbb{E}[Q_t|N_t = n, S_t = s]$ which is defined as follows:

$$\begin{aligned}\mathbb{E}[Q_t|N_t = n, S_t = s] &= \sum_{i=1}^{i=n} (1 - q_s)\, p, \\ &= (1 - q_s)\, pn, \end{aligned} \tag{4.15}$$

where $P(X_{t+\Delta_t}^i = 0|X_t^i = 1) = (1 - q_s) \times p$. Here, an individual is not stimulated with the probability q_s. Using the probability $P(N_t = n)$, we can rewrite the above equation as:

$$\begin{aligned}\mathbb{E}[Q_t|S_t = s] &= \sum_{n=0} (1 - q_s)\, pnP(N_t = n), \\ &= (1 - q_s)\, p\mathbb{E}[N_t]. \end{aligned} \tag{4.16}$$

We next consider the conditional expected value $\mathbb{E}[B_t|N_t = n, S_t = s]$. Here we use $P(X_{t+\Delta_t}^i = 1|X_t^i = 0, S_t = s) = (1 - q_s) \times \frac{s^2}{s^2+\theta^2}$. In this case, the conditional expected value $\mathbb{E}[B_t|N_t =$

$n, S_t = s]$ is formulated by the next equation:

$$\begin{aligned}\mathbb{E}[B_t|N_t = n, S_t = s] &= \sum_{i=1}^{i=M-D-n} (1-q_s)\,\frac{s^2}{s^2+\theta^2}, \\ &= (1-q_s)\,\frac{s^2}{s^2+\theta^2}\,(M-D-n)\,, \qquad (4.17)\end{aligned}$$

where a constant D is the number of dead individuals in a colony. Using the probability $P(N_t = n)$, we can rewrite the above equation as:

$$\begin{aligned}\mathbb{E}[B_t|S_t = s] &= \sum_{n=0} (1-q_s)\,\frac{s^2}{s^2+\theta^2}(M-D-n)P(N_t=n) \\ &= (1-q_s)\,\frac{s^2}{s^2+\theta^2}\,(M-D-\mathbb{E}[N_t]) \qquad (4.18)\end{aligned}$$

Finally, using the probability $P(S_t = s)$, we can get the expected value $\mathbb{E}[N_t]$, which is represented by the next equation:

$$\begin{aligned}\mathbb{E}[N_{t+\Delta_t}] &= \mathbb{E}[N_t] + \Delta_t \sum_{s=0} P(S_t = s)\,(-\mathbb{E}[Q_t|S_t = s] + \mathbb{E}[B_t|S_t = s]) \\ &= \mathbb{E}[N_t] - \Delta_t\,(1-q_s)\,p\mathbb{E}[N_t] \\ &\quad +\Delta_t\,(1-q_s)\,\mathbb{E}\left[1-\frac{\theta^2}{S_t^2+\theta^2}\right](M-D-\mathbb{E}[N_t])\,. \qquad (4.19)\end{aligned}$$

Following the definition of differential, the dynamics of the expected value $\mathbb{E}[N_t]$ can be formulated as follows:

$$\begin{aligned}\frac{d\mathbb{E}[N_t]}{dt} &= \lim_{\Delta_t \to 0} \frac{\mathbb{E}\left[N_{t+\Delta_t}\right] - \mathbb{E}\left[N_t\right]}{\Delta_t} \qquad (4.20) \\ &= (1-q_s)\left\{-p\mathbb{E}\left[N_t\right] + \mathbb{E}\left[1-\frac{\theta^2}{S_t^2+\theta^2}\right](M-D-\mathbb{E}\left[N_t\right])\right\}.\end{aligned}$$

4.4 Mathematical Analysis

In this section, we investigate the influence of individual failures in lossy environment using the analytical model described in section 4.3 and discuss the robustness of the response threshold model.

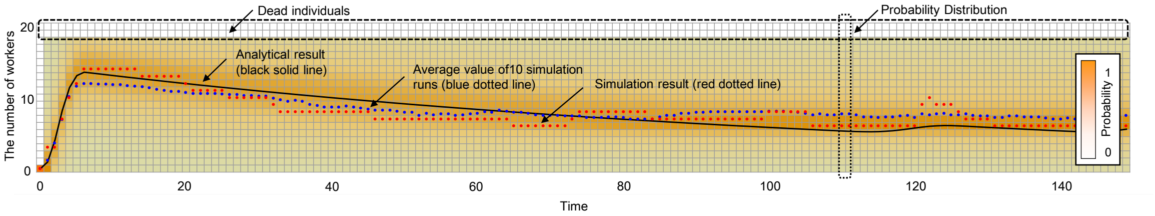


Figure 4.4: Comparison of the simulation results with the expected value and the PDF

We first verify the analytical model through comparison with simulation results. Next, we show that the existence of dead individuals does not seriously affect the distribution of the number of workers by comparing distributions in the steady state before and after failures. Then, we discuss conditions for the number of workers and the stimulus intensity to stay in the steady state after individual failures. For this purpose, we regard the colony as a dynamical system and conduct a linear stability analysis [70]. Finally, we discuss the influence of tunable control parameters on time required for convergence.

4.4.1 Validity of Analytical Model

Before we analyze the robustness against individual failures, we confirm the validity of the analytical model. For this purpose, we compare the expected value $\mathbb{E}[N_t]$ obtained by time evolution of Equations (4.11) and (4.20) with simulation results. In the experiments, we initially set the number N_0 of workers and the stimulus intensity S_0 at 0 and 0, respectively. Therefore, $\mathbb{E}[N_0] = \mathbb{E}[S_0] = 0$. It further means $X_i^0 = 0$ for all individuals. The other parameters θ, p, M, D, q_s, q_w, and δ are set at 1, 0.01, 20, 2, 0.1, 0.1 and 0.25, respectively. We consider the identical threshold $\theta_i = \theta$ for all i.

Figure 4.4 demonstrates the validity of the analytical model. In this figure, the x-axis corresponds to time t. The y-axis corresponds to the number N_t of workers. A color at (t, k) indicates the probability $P(N_t = k)$ obtained by calculating the joint probability $P(N_t = k | N_0, S_0, \ldots, N_{t-\Delta_t}, S_{t-\Delta_t})$ using Equations (4.5) and (4.13). As a color chart shows, a lighter color means a lower probability. A black solid line shows the expected value $\mathbb{E}[N_t]$, a blue dotted line corresponds to the average number of workers of 10 simulation experiments, and a red dotted line shows a sample of a simulation experiment.

We find that a blue dotted line lies within the dark-colored area, but not the darkest. It means that the average number of workers obtained from simulation experiments does not match with the mode, i.e. the most frequently appearing value. Now assume that the number of workers in a colony is identical to the mode. When some workers quit a task with the small quitting probability p, the number of workers decreases and consequently the stimulus intensity increases. In response to the increased stimulus, the number of workers increases but there appears redundant workers as shown in Figure 4.2. Then, the number of workers eventually decreases with the small quitting probability p. Since it takes time for the number of workers to reach the mode, the probability that the number of workers is more than the mode becomes higher than the one that the number of workers is less than the mode. As a result, the mode is different from the average. In Figure 4.4, we also notice that the average of simulation experiments, i.e. a blue dotted line, does not match with the expected value obtained from the analytical model, i.e. a black solid line. A reason for this is that there occurs an error in the expected value by deriving $\mathbb{E}[N_t]$ from $\mathbb{E}[N_{t-\Delta_t}]$. Furthermore, the error accumulates as time advances. As a result, the analytically derived expected value becomes different from the average value obtained from simulation experiments. However, analytical results draw the similar trajectory or shape with the blue dotted line. Therefore, we consider that the temporal variation or the transient behavior of the number of workers can be discussed by using our analytical model.

4.4.2 Robustness against Individual Failures

To investigate the robustness against failures of individuals in lossy environment, we evaluate the influence of dead individuals on the steady state, where the probability distribution of the number of workers is stable in time, under the condition that D out of M individuals in a colony are dead. In more details about experiments, we investigate the influence of dead individuals on the mode in order to show the number of workers is kept around the same value regardless of the number of dead individuals.

In the numerical experiments, both of N_0 and S_0 are set at 0. Using Equations (4.5) and (4.13), we iteratively calculate the joint probability $\forall_{i,j} P(N_{t+\Delta_t} = i, S_{t+\Delta_t} = j | N_0, S_0, \ldots, N_t, S_t)$ until the norm $\|P(N_{t+\Delta_t}, S_{t+\Delta_t}) - P(N_t, S_t)\|$ becomes infinitesimal, e.g. $< 10^{-7}$. Figures 4.5 and 4.6

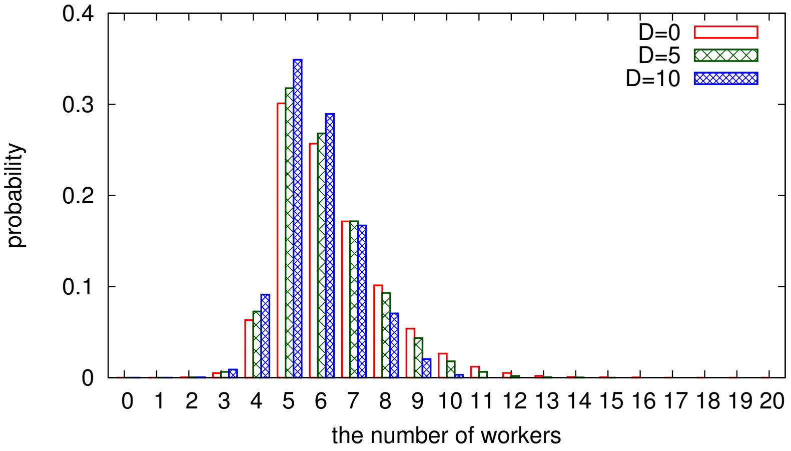


Figure 4.5: PDF of the number of workers in the steady state (the loss rate $q = 0.001$)

depict the obtained probability distribution of the number of workers when the probabilities q_w and q_s are equally set at 0.001 or 0.1. Hereinafter the probabilities q_w and q_s are collectively referred to as a "loss rate" q. The quitting probability p, the threshold value θ, the colony size M, and the increasing rate δ are set at 0.01, 1, 20 and $5/20$, respectively. In this parameter setting, the number of workers is kept around 5 if there is no influence of individual failures or information loss.

First, we investigate the mode of the number of workers. In each figure, bars correspond to cases of $D = 0$, 5, and 10, respectively. From Figure 4.5, we find that the mode is 5 regardless of the number of dead individuals. It means that the response threshold model can adjust the number of workers to an appropriate value even if a colony has some dead individuals. When the loss rate is 0.1, the mode increases to 6 but it is the same among different D. A reason of the increase is that $n(t)$ in Equation (4.1) becomes smaller than the actual number of workers due to loss of state information. As a result, the stimulus increases, and the number of workers increases.

Next, we focus on the probability that the number of workers is less than the mode. In both of Figures 4.5 and 4.6, the probability increases with a larger number of dead individuals. A reason for this is as follows. When the number of alive individuals is small, a larger stimulus is needed to have the sufficient number of workers. Since it takes time for a stimulus to increase, the period when the number of workers is insufficient is longer than a case of a smaller D.

The above mentioned results show the number of workers is kept around an appropriate value

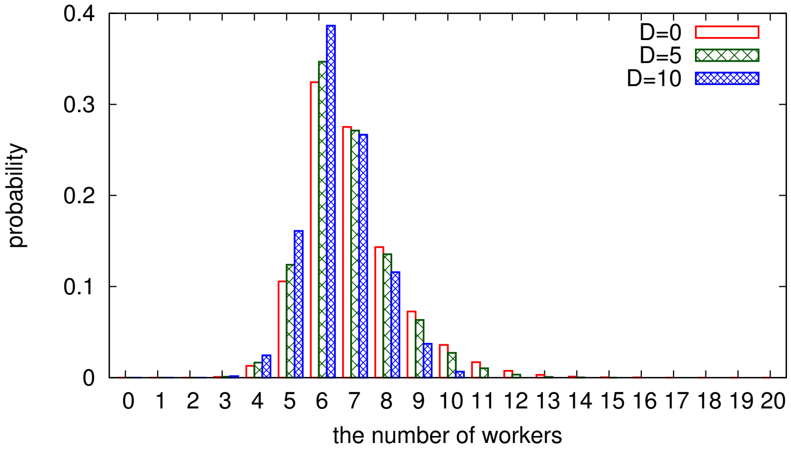


Figure 4.6: PDF of the number of workers in the steady state (the loss rate $q = 0.1$)

even if a colony has dead individuals. However, the existence of more dead individuals leads to a longer time for the number of workers to recover after it falls below the appropriate one.

4.4.3 Stability againt Individual Failures

We investigate conditions for the number of workers not to diffuse out of a fixed point. The fixed point is the condition that the number of workers and the increasing rate δ of the stimulus intensity are well balanced, and they do not change. We first consider an existence condition of the fixed point and then a stability condition for the number of workers to converge to the fixed point. For this purpose, we conduct a linear stability analysis [70]. In the following, for the simplicity of notation, we use $n(t)$ and $s(t)$ for $\mathbb{E}[N_t]$ and $\mathbb{E}[S_t]$, respectively.

The dynamics of the colony can be modeled as a general two-dimensional linear system in the form of $d\vec{x}(t)/dt = A \times \vec{x}(t)$, where $\vec{x}(t)$ denotes a two-dimensional state vector $(s(t),\ n(t))$, and A denotes a 2×2 state transition matrix. Using eigenvalues $\vec{\lambda}$ and eigenvectors $\vec{x}$, we can transform $d\vec{x}(t)/dt = A \times \vec{x}(t)$ to $d\vec{x}(t)/dt = \vec{\lambda} \times \vec{x}(t)$. Therefore, the dynamics of the system can be analyzed by evaluating eigenvalues of the matrix A. Eigenvalue λ_i $(i \in \{1, 2\})$ is generally formulated as $\alpha_i + j\beta_i$, where j is an imaginary number $\sqrt{-1}$, and α_i and β_i are real numbers. $s(t)$ is $s(0) \times \exp^{(\alpha_1+j\beta_1)t} = s(0) \times \exp^{\alpha_1 t} \times (\cos\beta_1 t + j\sin\beta_1 t)$. $n(t)$ is $n(0) \times \exp^{(\alpha_2+j\beta_2)t} = n(0) \times \exp^{\alpha_2 t} \times (\cos\beta_2 t + j\sin\beta_2 t)$. When any α_i is smaller than 0, $s(t)$ and $n(t)$ converge to 0 as

$t \to \infty$. That is, the system with $\forall_i \alpha_i < 0$ has asymptotic stability while smaller $|\alpha_i|$ leads to longer time to become stable. In this chapter, we call α_i a "damping factor". When the dynamical system is stable, $d\vec{x}/dt$ is 0. We denote the fixed point as $[\bar{s}\ \bar{n}]^T$. Linearizing the nonlinear analytical model defined by Equations (4.11) and (4.20) at the fixed point $[\bar{s}\ \bar{n}]^T$ by Taylor expansion, we analyze the influence of the number D of dead individuals and the loss rate $q\ (= q_w = q_s)$ on transient behavior during recovery from individual failures.

First, we derive a fixed point $[\bar{s}\ \bar{n}]^T$, where temporal variation of the expected values $s(t)$ and $n(t)$ are 0. By solving $ds/dt = 0$ and $dn/dt = 0$, we obtain

$$\bar{s} = \theta\sqrt{\frac{p\delta}{\left(1-\frac{D}{M}\right)(1-q_w) - \delta(1+p)}}, \tag{4.21}$$

$$\bar{n} = \frac{\delta M}{1-q_w}. \tag{4.22}$$

Since the stimulus intensity is 0 or a positive real number, $(1 - D/M)(1 - q_w) > \delta(1 + p)$ must hold. At the same time, $1 - D/M \geq \delta/(1 - q_w)$ must hold so that the number of workers is smaller than the number of living individuals. Using $\delta(1 + p) > \delta$, the condition that a feasible fixed point exists is consequently given by the following inequality:

$$\left(1 - \frac{D}{M}\right)(1 - q_w) - \delta(1 + p) > 0. \tag{4.23}$$

Next, the dynamics of error $\vec{e} = [e_s\ e_n]^T = [\bar{s}\ \bar{n}]^T - [s\ n]^T$ can be formulated as a basic linear equation $d\vec{e}/dt = A \times \vec{e}$.

$$\frac{de_s}{dt} = 0e_s - (1 - q_w)e_n \tag{4.24}$$

$$\frac{de_n}{dt} = (1 - q_s)\left\{(M - D - \bar{n})\frac{2\bar{s}\theta^2}{(\bar{s}^2 + \theta^2)^2}e_s - \left(p + \frac{\bar{s}^2}{\bar{s}^2 + \theta^2}\right)e_n\right\} \tag{4.25}$$

Then, we derive eigenvalues of the matrix A in $d\vec{e}/dt = A \times \vec{e}$ from a characteristic equation $\det|A - \lambda I| = 0$. Since the matrix A is in the form of $[0\ a;\ b\ c]$, an eigenvalue is formulated as $\frac{1}{2} \times (c \pm \sqrt{c^2 + 4ab})$. c and ab are specifically derived as follows, by substituting $\bar{s}$ and $\bar{n}$ of

Equations (4.21) and (4.22).

$$\begin{aligned} c &= -\frac{p(1-\frac{D}{M})\,(1-q_s)\,(1-q_w)}{\left(1-\frac{D}{M}\right)(1-q_w)-\delta} && (4.26) \\ ab &= -\frac{2}{\theta}\sqrt{p\delta}(1-q_s)\frac{\left\{(1-\frac{D}{M})(1-q_w)-\delta\,(1+p)\right\}^{\frac{3}{2}}}{(1-\frac{D}{M})(1-q_w)-\delta} && (4.27) \end{aligned}$$

$c<0$ and $ab<0$ must hold for a real part of an eigenvalue to be always less than 0. When Equation (4.23) is satisfied and a fixed point exists, $0<1-q_s$ and $0<p+\frac{\bar{s}^2}{\bar{s}^2+\theta^2}$ hold. Therefore, the first condition $c<0$ is always met. In the same way, the second condition $ab<0$ is also met because of $0<1-q_w$, $0<1-q_s$, $0<\frac{2\bar{s}\theta^2}{(\bar{s}^2+\theta^2)^2}$, and $0<M-D-\bar{n}$. Thus a real part of an eigenvalue is always negative. Therefore, a state of the dynamical system does not diffuse out of the proximity of the fixed point.

4.4.4 Time Required to Recover from Individual Failures

We derive time T required for the number of workers to recover from individual failures by considering transient dynamics of the dynamical system after failures. Let us suppose that f percent of $\bar{n}$ workers die in a stable colony. f is specifically called a "failure rate". The recovery time T is defined as time required for deviation $e_n(t)=\bar{n}-n(t)$ to become as small as b ($b<f$) percent of $\bar{n}$ from failures. We consider failures occur at time 0. The deviation $e_n(0)$ soon after individual failures is identical to $\frac{f}{100}\times\bar{n}$. As described in Section 4.4.3, the deviation decreases as $e_n(t)=e_n(0)\times\exp^{\alpha_{min}t}$, where α_{min} is a damping factor which is derived by $\min Re\,\frac{1}{2}\times(c\pm\sqrt{c^2+4ab})$. Therefore, solving the equation $\frac{b}{100}\times\bar{n}=\frac{f}{100}\times\bar{n}\times\exp^{\alpha_{min}T}$, we can define the recovery time T as follows:

$$T=\frac{\log b-\log f}{\alpha_{min}}. \tag{4.28}$$

Figure 4.7 shows the dependence of the recovery time T on the failure rate f and the loss rate q. The quitting probability p, the threshold value θ, and the increasing rate δ are set at 0.1, 10, and 0.25, respectively. b in Equation (4.28) is set at 5. A redder color means longer recovery time. Contour

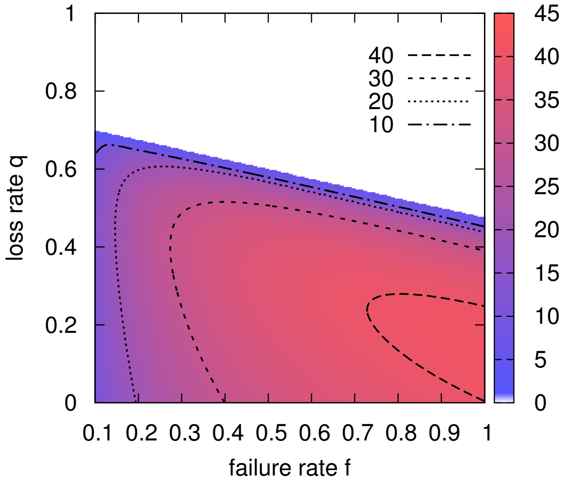


Figure 4.7: Influence of loss and failure on the recovery time

lines are shown for recovery time of 10, 20, 30, and 40, respectively. In the white area, a fixed point $[\bar{s}\ \bar{n}]^T$ does not exist. That is, the stimulus intensity continues to increase, and all individuals eventually become workers. Equation (4.23) means that a smaller p can reduce the invalid white area. As shown in the figure, an increase in the recovery time T is slow when the failure rate f increases. For example, we consider the case of $q = 0.2$ leading to the largest recovery time. When the failure rate increases from 0.1 to 1.0, the recovery time increases only by 3.6-fold. In addition, the contour lines shows that the gradient of the recovery time is smooth. These results imply that the recovery time does not diverge, and the response threshold model is valid in a hostile environment where individuals are prone to failures.

Figure 4.8 shows the influence of control parameters, i.e. the quitting probability p and the threshold value θ, on the recovery time T. The information loss rate q, the failure rate f, and the increasing rate δ are set at 0.1, 0.2, and 0.25, respectively. The figure shows that the threshold value θ does not have a visible influence on the recovery time for the fixed quitting probability p. In the response threshold model, the stimulus intensity must be large enough to have the sufficient number of workers depending on a threshold value. It means that the stimulus intensity is sufficiently large

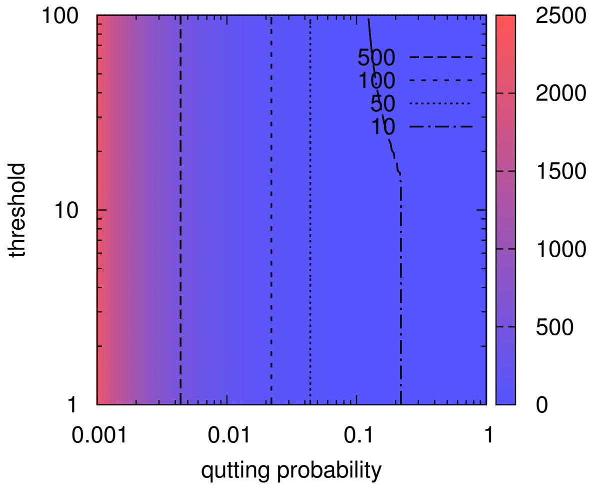


Figure 4.8: Influence of parameters p and θ on the recovery time

when individuals die after convergence. Therefore, even when individuals die with the probability of 0.2, the large stimulus soon changes idle individuals to workers. As a result, the recovery time is not affected by the value of the threshold θ.

On the contrary, the figure shows that the quitting probability p has an explicit influence on the recovery time for the fixed threshold value θ, and a smaller quitting probability p makes the recovery time longer. However it does not necessarily mean that the response threshold model is sensitive to the quitting probability p. A worker is engaged in a task for a longer period of time with a smaller p as explained in Section 4.2. Therefore, a reason for longer recover time with a larger p is that it takes longer for redundant workers to quit a task, while the sufficient number of workers satisfies the demand.

We can conclude that the control parameters θ and p do not have an influence on the time to recover from a lack of workers due to individual failures.

4.5 Simulation-based Evaluation

In Section 4.4, we investigated the robustness against failures of individuals under the assumption that individuals are fully connected with each other. Therefore, the death of an individual does not affect communication among other alive individuals. On the contrary, communication among a pair of nodes is mediated by intermediate nodes, i.e. multi-hop communication, in a general information network. As such, dead nodes would make a network disconnected and introduce isolated nodes. Isolated nodes cannot receive stimuli, and their states are excluded from derivation of $n(t)$ and $s(t)$. Furthermore, information about state and stimulus has a higher chance of loss due to multi-hop transmission. In this section, we verify the robustness of the response threshold model against individual failures in a lossy multi-hop network.

4.5.1 Network Control Model

We consider a hypothetical system consisting of homogeneous nodes capable of performing a task. A node has an unique address. One of nodes is appointed as a request node, which does not perform a task by itself. Instead, it asks other nodes to perform a task by flooding the stimulus intensity in the form of a “request”. A node memorizes an address of its neighbor node from which it receives a request first. As a response to the stimulus in a request, each node decides its state, that is, whether to perform the task or not, and notifies the request node of the decision by sending a “notification”. A notification is forwarded to the request node following the reversed path of the request. When a neighbor node is dead, from which a node received the corresponding request, a notification is silently discarded. Here nodes which perform the task are called “active nodes”. The other nodes are called “idle nodes”. Those nodes that cannot receive the stimulus due to loss of requests do not change their state. Additionally, those that cannot receive the stimulus for I_r (> 0) s change to idle nodes. A request node knows the number of active nodes from received notifications and derives the new stimulus intensity by using Equation (4.1). The stimulus intensity is again diffused to nodes. An interval of stimulus diffusion is I_c (> 0) s, which corresponds to a control interval of an application system.

While performing a task, each active node informs the request node of its result at regular

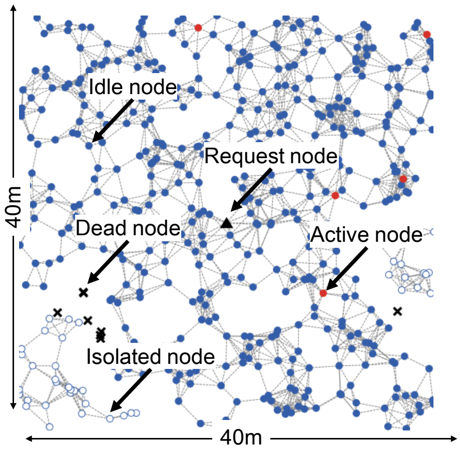


Figure 4.9: Snapshot of simulation experiments

intervals of I_d $(0 < I_d \leq I_c)$ s. We call a message for this a "report". A report message is forwarded following the reversed path of the last received request. When the designated neighbor node is dead, a node silently discards a report. For example, in a case of a monitoring application of a wireless sensor network, e.g. automatic meter reading, each sensor node periodically sends a report containing sensor reading, such as temperature and humidity, to a sink.

4.5.2 Simulation Setting

In simulation experiments, we randomly placed 500 nodes in a region of $40 \text{ m} \times 40 \text{ m}$. We additionally place a request node at the center of the region. An example of a generated topology is illustrated in Figure 4.9. A node can communicate with nodes within the proximity of 3 m. We assume that communication delay is negligible. There are three types of messages: request, notification, and report. Here we put an assumption that three messages are lost with the same probability q $(0 \leq q \leq 1)$ in a per-hop basis. A request node requires reports from 5 nodes every I_d s. Other parameters are summarized in Table 4.1.

In the following simulations, we try two scenarios where influence of node failures is different:

- Random failure scenario: Before we begin a simulation run, we randomly select and remove D $(0 \leq D \leq 500)$ nodes except a request node from an initial topology.

Table 4.1: Parameter setting

Notation	Description	Value
M	The number of nodes	500
θ	Hesitation to become an active node	1
p	Probability of becoming an idle node	0.01
I_c	Interval between two successive stimulus diffusion	1
I_d	Interval between two successive report	1
I_r	Timer for active nodes to change to idle nodes	10
δ	Increasing rate of stimulus intensity	0.01

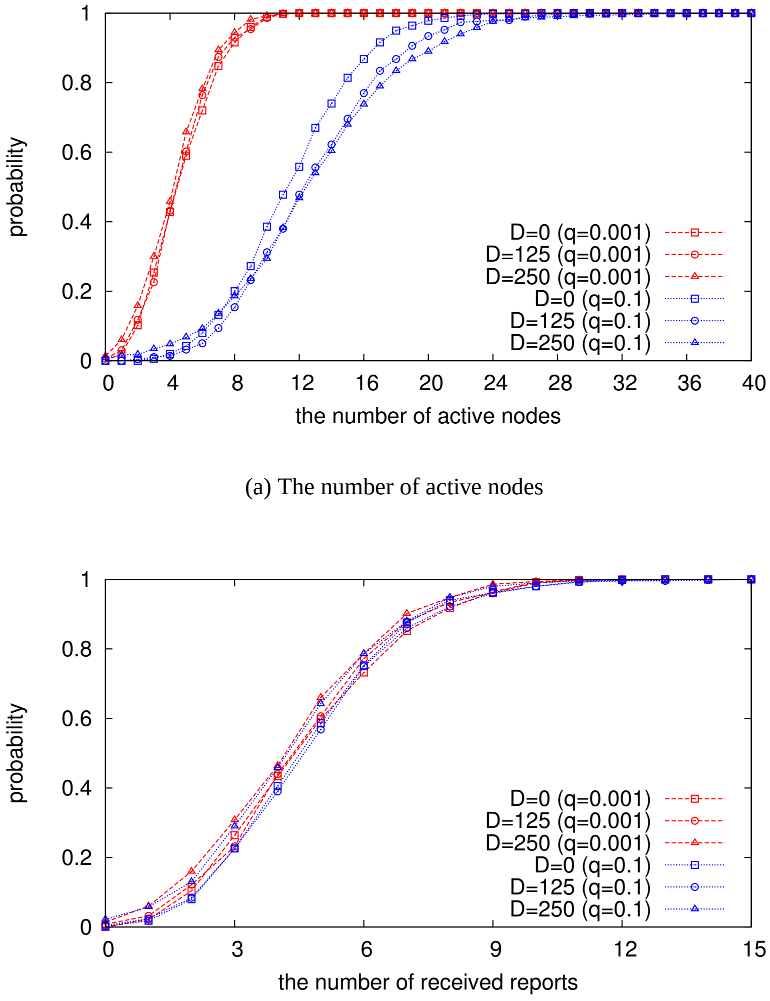
- Selective failure scenario: Before we begin a simulation run, we select and remove D nodes except a request node from an initial topology in descending order of a degree.

We conduct 500 simulation runs and use results at 500 s in the following evaluations. The initial state of nodes are idle. We evaluated six combinations of $D = 0$, 125, and 250 and $q = 0.1$ and 0.001.

4.5.3 Results of Random Failure Scenario

In Figure 4.10, we draw the cumulative probability distribution of the number of active nodes and the number of received reports in the random failure scenario. Figure 4.10(a) shows that the number of active nodes increases as the loss rate q becomes higher while the loss rate q does not affect the number of received reports as shown in Figure 4.10(b). The loss of notifications leads to the increase in the stimulus intensity because the request node estimates the number of active nodes to be smaller than the actual one. By being stimulated by stronger stimulus, a larger number of nodes become active nodes as the loss rate q becomes higher.

In addition, we can also find that the number of active nodes with the loss rate $q = 0.1$ is larger than that with the loss rate $q = 0.01$. In the response threshold model-based network control, requests, notifications, and reports are forwarded in a multi-hop manner. When there are dead nodes, the number of hops from a node to a request node increases. As a result, the probability of loss of messages increases. Consequently the stimulus intensity with $q = 0.1$ becomes larger than that with $q = 0.01$, and it makes more active nodes.



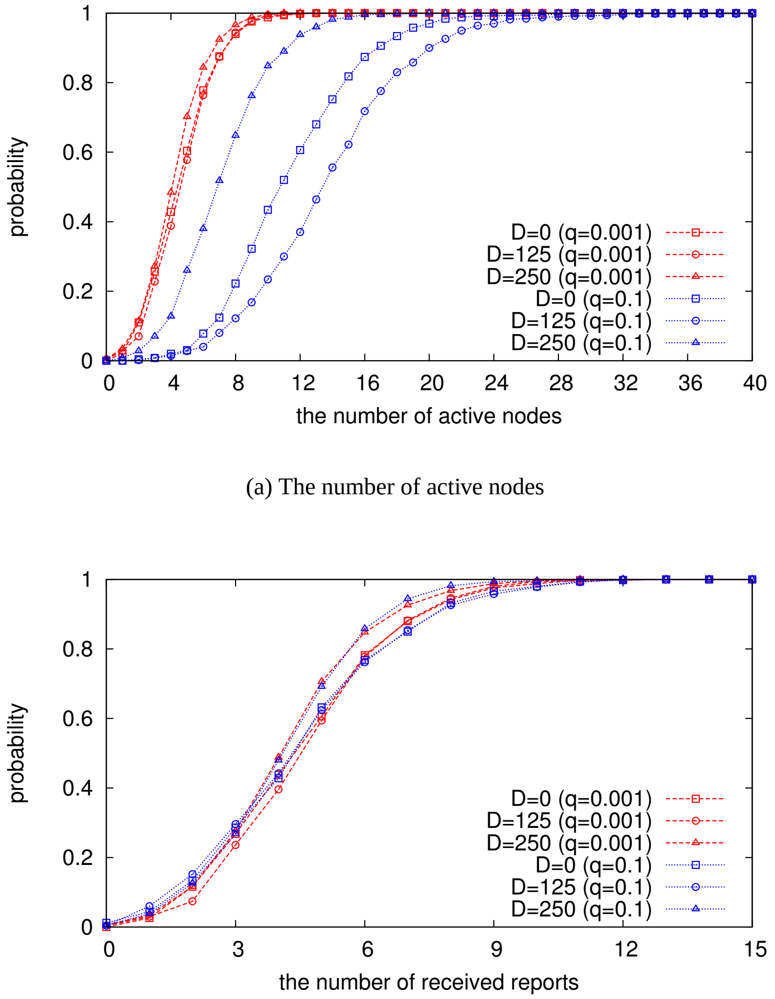
(a) The number of active nodes

(b) The number of received reports

Figure 4.10: Influence of random node failures

4.5.4 Results of Selective Failure Scenario

In Figure 4.11, we draw the cumulative probability distribution of the number of active nodes and the number of received reports in the selective failure scenario. From Figure 4.11(a), we observe that the number of active nodes with $D = 250$ is the smallest in the case of $q = 0.1$. It is quite different from the random failure scenario.



(a) The number of active nodes

(b) The number of received reports

Figure 4.11: Influence of selective node failures

In the response threshold model-based network control, a request is diffused by flooding. As more dead nodes exist, the number of paths from the request node to a node becomes smaller because the network becomes more sparse. This causes difficulty for a node to receive a request. This makes an idle node more keep its state and an active node more become an idle node. The number of paths to a node located far more away from the request node specifically decreases because of difficulty in taking a detour. Therefore, a node near the request node is apt to become an active node in response to increase in the stimulus intensity. Smaller hop length of a node nearer the request node makes the probability that its notification or report drops lower. This contributes to the stimulus intensity calculated by the accurate number of active nodes, and the stimulus intensity becomes smaller than other simulation settings. Then, the number of active nodes becomes smaller while the number of reports is kept as shown in Figure 4.11(b).

4.6 Summary

In this chapter, we analyzed characteristics of the response threshold model. First we built an analytical model which takes into account individual failures and information loss. Next we showed that the response threshold model is highly robust as much as the number of workers can recover without divergence of the recovery time even if all workers die. Then we investigated the robustness in a multi-hop network and revealed that the number of workers is adaptively adjusted depending on a network topology where random or selective failures occur.

Chapter 5

Design Policy for Bio-inspired Network Control to Achieve High Adaptability

Bio-inspired network control is a promising approach for realizing adaptive network controls. It relies on a probabilistic mechanism composed of positive and negative feedback that allows the system to eventually stabilize on the best solution. As a result, the temporary failure to function if the solution fails due to environmental fluctuation cannot be avoided. To suppress this possibility, we need to prevent the system from stabilizing on specific solutions. This causes the system to continually search for alternative solutions. However, most bio-inspired network controls are not designed with this issue in mind. In this chapter, we propose a thermodynamics-based design policy that allows systems to retain an appropriate degree of randomness depending on the degree of environmental fluctuation, which prepares the system for the occurrence of environmental fluctuation. Furthermore, we verify the design policy by using an attractor selection model-based multi-path routing to run simulation experiments.

5.1 Introduction

For information and communication networks to serve as an indispensable part of the infrastructure for secure, dependable, and comfortable society, they must be more robust against ever-increasing size, dynamic changes, and complexity [71]. In recent years, many researchers have focused on interdisciplinary approaches to spark innovative ideas. In particular, they have been actively working on network controls that are inspired by biological behavior, and many published chapters support the usefulness of such systems [66, 67, 72].

A bio-inspired network control relies on a probabilistic mechanism composed of positive and negative feedback that allows searching for better solutions. On finding better solutions, the system more aggressively selects in the direction of those solutions as a result of its positive feedback. The system eventually stabilizes on the best solution by minimizing its negative feedback [62]. This type of mechanism will need to search for a solution again in the case where the known solution fails due to environmental fluctuations, such as node failures and link failures. Therefore, it is not possible to avoid a temporary loss of function. This process implies that it is not always reasonable for the network control to select the solution that is optimum at a particular time. To prepare for the occurrence of environmental fluctuation, it is quite important that such systems also select other solutions in addition to the optimum one. However, most existing bio-inspired network controls are not designed with this issue in mind.

Let us assume that the tendency for bio-inspired network controls to select the optimum solution is measured by *ordering energy*. The ordering energy relates to their potential performance, which we can potentially obtain in environments that have no fluctuation. We also assume that the tendency for them to select other solutions in addition to the optimum solution is measured by *disordering energy*. Disordering energy is related to performance degradation caused by environmental fluctuation [73]. Using ordering energy and disordering energy, we can describe an appropriate design for the steady state. When a system is deployed in a stable environment, its steady state should be designed to have high ordering energy. In contrast, for systems in dynamic environments, the design should prevent its performance from degrading, and therefore should cause the steady state to have low disordering energy [74]. As a consequence, we need to design systems to balance between their ordering energy and their disordering energy according to the degree of environmental fluctuation as a means of preparing the systems for the occurrence of environmental fluctuation.

To give a quantitative interpretation of ordering and disordering energy, we focus on thermodynamic free energy, which indicates the state of a natural substance. Thermodynamics says that free energy A is formulated by equation $A = E - T \times S$, where A, E, T, and S are the free energy, internal energy, temperature, and entropy, respectively. The details are described in Section 5.3, but we describe the model briefly here. Internal energy E corresponds to ordering energy. The product $T \times S$ corresponds to disordering energy. Assuming that temperature T does not change, this equation implies that the temperature T can be used to determine which change is effective in changing free energy A, internal energy E or entropy S. From the standpoint of designs for bio-inspired network controls, we can obtain a design for balancing the ordering energy with the disordering energy by regarding temperature T as the degree of environmental fluctuation, which allows us to prepare systems for its occurrence. The obtained design contributes to making performance when the fluctuation occurs higher than other designs.

In this chapter, we establish and verify a thermodynamics-based design policy for network controls that will cause them to reach the steady state appropriate to the degree of environmental fluctuation. For this purpose, we first explain a steady state of network controls in terms of thermodynamics. Next we preliminarily investigate an appropriate steady state, which depends on the given degree of environmental fluctuation. Then, we build an analytical network control and formulate and analyze its thermodynamic state values. From the analytical results, we obtain a thermodynamics-based design policy. Finally, we verify the design policy by using an attractor selection model-based multi-path routing [75] as an existing control.

The rest of this chapter is organized as follows. First, in Section 5.2, we describe related work. Next, we describe the interpretation of network controls from the perspective of thermodynamics in Section 5.3. Then, in Section 5.4, we explain preliminary simulation results. We build and analyze an analytical model of a network control, and we establish a design policy in Section 5.5. In Section 5.6, we verify the design policy. Finally, we conclude this chapter.

5.2 Related Work

Researchers are focusing on network controls based on biological self-organization (hereinafter, BSON) in order to realize more robust communication networks. For instance, ant colony optimization [8], which is inspired by the foraging behavior of ants, has been applied to routing algorithms [76], and pulse-coupled oscillation [9], which is inspired by the synchronization behavior of fireflies, has been applied to time synchronization algorithms [77]. Useful global behaviors arise from direct and indirect interactions among nodes. However, this bottom-up approach can lead to difficulty in designing or optimizing the global behaviors of BSON controls.

To optimize a BSON control, many researchers analyze the influence of its control parameters on its characteristics (such as optimality and robustness) through simulation mathematical analysis [65, 78]. However, each analysis is typically focused on a specific BSON control. As a consequence, there is little general knowledge that can be used in optimizing existing or future BSON controls. Some other studies have tried to quantitatively define characteristic values (see, for example, [79, 80]). Such general definitions are helpful during the design and optimization of existing and future BSON controls. However, it is typically supposed that each element's behavior can be represented by a simple stochastic automaton. This makes it difficult to define the characteristic values of a BSON control that has complicated behaviors. As another approach, some argue that the system should be constricted in a top-down manner to realize the desired behaviors of BSON controls. For this style of approach, the self-organizing behavior is elucidated by focusing on an underlying thermodynamic principle [81, 82]. However, that work does not provide us with a concrete method for designing and optimizing BSON controls. Still other researchers have focused on directly controlling BSONs [83, 84], aiming to achieve the desired behavior by controlling a part of the elements. The research along this vein, however, targets improvements in transient characteristics, such as convergence speed. It does not consider how steady states should emerge to achieve sufficient robustness and performance under a given set of network conditions.

In conclusion, there is no general method to design or optimize a BSON control while taking into account the assumed network conditions.

Table 5.1: Interpretation from the thermodynamic perspective

	Description in terms of thermodynamics
Internal energy E	Energy that can potentially be extracted
Entropy S	Difficulty of extracting energy
Temperature T	Average energy of particles
Free energy A	Energy that can actually be extracted

5.3 Thermodynamic Interpretation

By analyzing a network control from the perspective of its free energy, we can discuss the balance between the system's ordering energy and its disordering energy according to the degree of environmental fluctuation. In the following subsections, we briefly introduce a thermodynamic model and interpret the state of a network control from the thermodynamic perspective.

5.3.1 Thermodynamics

Natural substances stabilize in the state with the minimum free energy [41]. Free energy is generally formulated by the following equation:

$$A = E - T \times S, \tag{5.1}$$

where A, E, T, and S are the free energy, internal energy, temperature, and entropy, respectively. Collectively, these values are "thermodynamic state values". Internal energy E is the stored energy, which we can potentially extract as heat or work, and corresponds to the ordering energy. Entropy S measures the difficulty of extracting energy from the substance. A higher entropy S means that it is more difficult to transform the internal energy E to heat or work. Temperature T measures the average energy of particles of the substance. When the temperature T is higher, the substance includes particles with higher energy. From the definitions of temperature T and entropy S, their product $T \times S$ measures the energy that cannot be extracted from the substance; this corresponds to the disordering energy. From Equation (5.1), this implies that the free energy A is the energy that we can actually extract from the substance. Thus, natural substances become stabilized in a

Table 5.2: Interpretation from the perspective of network controls

	Description in terms of network controls
Internal energy E_{nw}	Performance that can potentially be obtained
Entropy S_{nw}	Difficulty of maintaining performance
Temperature T_{nw}	Frequency of environmental fluctuation
Free energy A_{nw}	Performance that can actually be obtained

state where change from internal energy to heat or work do not occur. The descriptions above are summarized in Table 5.1.

5.3.2 Network Controls

A good state (e.g., a routing table with the shortest paths) of network controls emerges via direct or indirect interactions among nodes. Let us assume here that when a network control has higher internal energy E_{nw}, it can potentially achieve higher performance. Note here that the term “potentially” means in an ideal environment, where no fluctuation occurs. For the sake of simplicity, the internal energy E_{nw} is called “optimality”. We assume that a network control has higher entropy S_{nw} when its performance is more susceptible to environmental fluctuation. Therefore, the rate of performance degradation for the given degree of fluctuation corresponds to the system’s entropy. For the sake of simplicity, entropy S_{nw} is called “robustness” (note that in this chapter small S_{nw} means high robustness). We assume that a network control has a higher temperature T_{nw} when the environment changes more frequently. Therefore, we can quantify the temperature by node failure rate, link error rate, and so on. We find that the product $T_{nw} \times S_{nw}$ corresponds to the performance decrease caused by the fluctuation; this follows from the interpretations of S_{nw} and T_{nw}. We also find that free energy A_{nw} corresponds to the performance actually achieved in a fluctuating environment, as shown in Equation (5.1). Thus, we should design network controls to have higher free energy. The interpretations given above are summarized in Table 5.2.

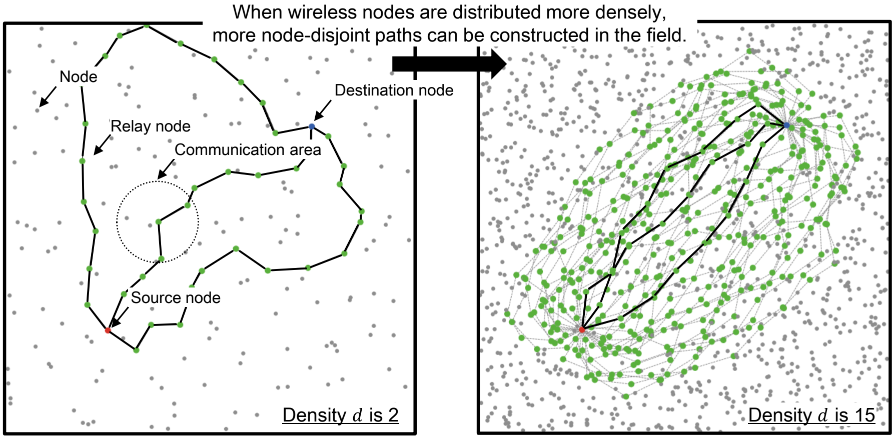

When wireless nodes are distributed more densely,
more node-disjoint paths can be constructed in the field.
Node
Destination node
Relay node
Communication area
Source node
Density d is 2
Density d is 15

Figure 5.1: Influence of node density d on node-disjoint paths

5.4 Preliminary Investigation

We preliminarily investigate an appropriate design for a network control that accounts for the degree of environmental fluctuation. For this purpose, taking multi-path routing [85] as an example of a network control, we conduct simulation experiments to investigate the influence of a design whose optimality and robustness are different components of performance.

5.4.1 Simple Model of a Multi-path Network

We here consider a simple model of a multi-path network. In this model, many wireless nodes are randomly distributed in the field, and each wireless node can communicate with other wireless nodes that are within a certain radius. Some node-disjoint paths are constructed between a pair consisting of a source node and a destination node.

More specifically, wireless nodes are randomly distributed in a field of size $10\ \mathrm{m} \times 10\ \mathrm{m}$. To distribute the nodes uniformly, we partition the field into blocks of size $1\ \mathrm{m} \times 1\ \mathrm{m}$ and randomly allocate d (≥ 1) wireless nodes in each of the blocks. The constant d is called the "node density". In total, there are $100d$ wireless nodes in the field. In addition to these $100d$ wireless nodes, we place two more wireless nodes, one at $(2.5, 2.5)$ and one at $(7.5, 7.5)$. We call the former wireless node the "source node" and the latter wireless node the "destination node". Two examples of this wireless network are illustrated in Figure 5.1.

Each wireless node can communicate with other wireless nodes whose position is within a circle of radius $1\ \mathrm{m}$. A wireless node succeeds in sending a packet to a neighboring wireless node with probability $1 - l_i$, where the constant l_i $(0 \leq l_i \leq 1)$ is the probability that a packet to node i is dropped due to environmental fluctuation. Let us assume a simple model of environmental fluctuation. In this model, the source of environmental fluctuation is at position $(5, 5)$, in the center of the field. Environmental fluctuations affect the probability l_i for wireless node i according to the following formula:

$$l_i = q_{max} \times \exp\left[-\frac{(x_i - 5)^2 + (y_i - 5)^2}{2 \times \sigma^2}\right]. \tag{5.2}$$

In this equation, the position of node i is denoted by (x_i, y_i). The constant q_{max} $(0 \leq q_{max} \leq 1)$ sets the degree of environmental fluctuation. As this constant is larger, the environment more frequently fluctuates. The constant σ^2 (> 0) denotes the extent of the area affected by environmental fluctuation. As the constant q_{max} becomes larger, environmental fluctuation occurs more frequently. As the constant σ^2 becomes larger, a wider area is affected by environmental fluctuation.

We search for all node-disjoint paths between the source node and the destination node. We use the shortest three paths as “path candidates”. When the source node sends a packet to the destination node, it probabilistically selects a path from among the path candidates. The probability that the source node selects path j is given by a set of constants $p_{j \in \{1,2,3\}}$ $(0 \leq p_j \leq 1)$ in advance. A packet is iteratively forwarded to the destination node along the selected path.

5.4.2 Path Candidates with Different Characteristics

We prepare two types of path candidates, which have different characteristics: hop length and susceptibility to environmental fluctuation. For this purpose, we investigate topologies with different node densities d.

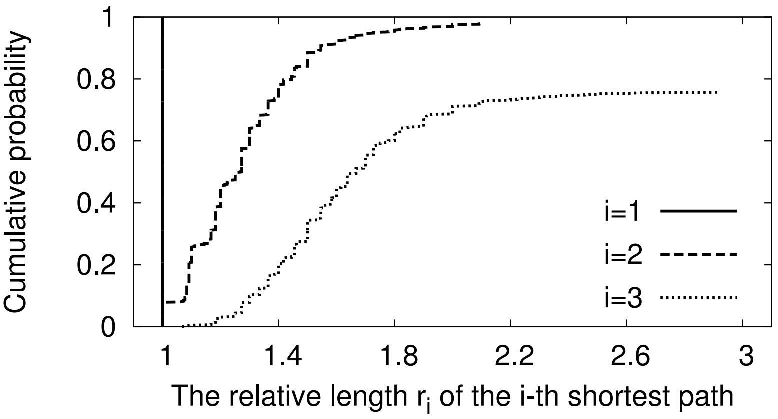
The left panel of Figure 5.1 is a sample of node-disjoint paths when nodes are sparsely distributed ($d = 2$). The right is a sample of node-disjoint paths when nodes are densely distributed

($d = 15$). In these figures, black lines denote the path candidates, that is, the three shortest node-disjoint paths. Gray lines denote other node-disjoint paths. Red, blue, and green nodes are the source, destination, and relay nodes, respectively. When the node density d is 2, there is a small number of node-disjoint paths. These paths detour, and the path candidates are long paths and cover a wide area. When node density d is 15, there are a large number of node-disjoint paths. These paths lie near the shortest path, and here they are as short as the shortest path.

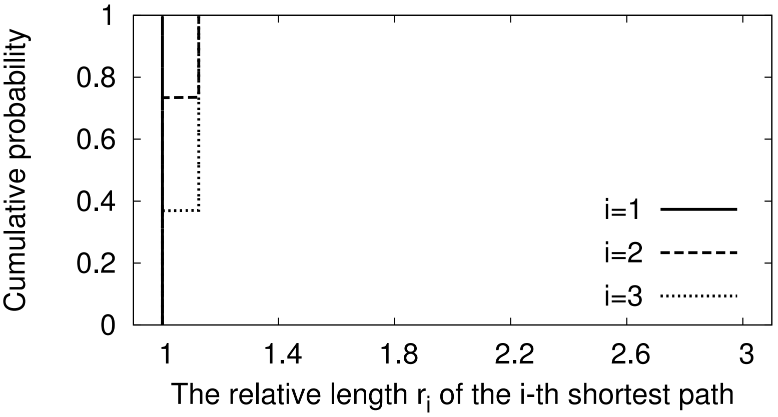
Figure 5.2 shows the relative lengths $r_j\ (\geq 1)$ of the j shortest paths. Here the relative length r_j is defined as $r_j = n_j/n_1$, where $n_j\ (\geq 1)$ is the hop length of the jth shortest path. For each node density, we randomly generate $1,000$ topologies and calculate r_j for all path candidates and topologies. Figures 5.2(a) and 5.2(b) are the cases for node density d at 2 and 15, respectively. In these figures, the x-axis corresponds to the relative length r_j, and the y-axis corresponds to the cumulative incidence of successful transmission in the trials.

Figure 5.2(a) shows that the cumulative probability does not always reach 1 when the node density d is 2, that is, in the sparse case. This means that it becomes more difficult to find suitable detours because the wireless nodes are more sparsely distributed. There are at least two successful paths in 980 samples out of $1,000$ samples. However, half of them are 1.2 times longer than the shortest one. In the worst case, they are 2.1 times longer. In 757 of $1,000$ samples, we can obtain three paths. The middle-length paths are 1.5 times longer than the shortest path. The longest paths are 3.0 times in the worst case. Thus, the number of detours is small in the sparse case. However, the detours lie in a wide area far from the shortest path, and the path candidates are not simultaneously affected by environmental fluctuation.

In contrast, Figure 5.2(b) shows that the cumulative probability always reaches 1. This result means that detours always exist because nodes are more densely distributed. In 70% of all samples, the hop length of the second-shortest path is equal to the hop length of the shortest path length. At the worst case, it is only 1.25 times the length. In 40% percent of all samples, the hop length of the third-shortest path is the same as the hop length of the shortest path. In the worst case, it is only 1.25 times the length. Thus, more detours exist in the dense case. However they lie in a narrow area near the shortest path. As a result, the path candidates are simultaneously affected by environmental fluctuation.



(a) Node density $d = 2$



(b) Node density $d = 15$

Figure 5.2: Influence of node density d on the three shortest node-disjoint paths

From the above results, the path candidates consist of paths that take a roundabout route when the node density d is small, such as when $d = 2$. In such cases, each path is not simultaneously affected by environmental fluctuation. However, the alternate paths are longer than the shortest path. In contrast, when the node density d is large, such as when $d = 15$, the path candidates include paths of equal length, but these paths are apt to be simultaneously affected by environmental fluctuation.

In the next subsection, we investigate an appropriate design for multi-path routing that depends on the degree of environmental fluctuation q_{max}.

5.4.3 Design Depending on Degree of Fluctuation

We discuss an appropriate design for multi-path routing depending on the degree q_{max} of environmental fluctuation. For this purpose, we conduct simulation experiments with three designs that have different levels of optimality and robustness (as defined in Section 5.3.2) and examining different node densities d. The source node can avoid selecting a path susceptible to environmental fluctuation by making the robustness higher. However, it becomes difficult for the source node to select the shortest path when its optimality is lower, even when the path would not be affected by environmental fluctuation.

Rule–A maximizes optimality without concern for robustness. For the rule, probabilities p_1, p_2, and p_3 are set at 1, 0, and 0, respectively. Rule–B lowers the optimality from Rule-A but raises the robustness. For the rule, probabilities p_1, p_2, and p_3 are set at 0.7, 0.15, and 0.15, respectively. Rule–C further lowers the optimality and further raises the robustness from Rule–B. For the rule, probabilities p_1, p_2, and p_3 are set at 0.4, 0.3, and 0.3, respectively. In all cases, the constant σ^2 is 1, the node density d is 2 or 15, and the degree q_{max} of environmental fluctuation ranges from 0 to 1 in step sizes of 0.1.

Figure 5.3 shows the simulation results. In the figure, the x-axis corresponds to the degree q_{max} of environmental fluctuation, and the y-axis corresponds to fitness. Here, fitness is defined as n_1/n_j when the destination node succeeds in receiving a packet using path j and as 0 otherwise. We choose topologies with 3 or more node-disjoint paths. We send $1,000$ packets, and we calculate the average fitness by using the $1,000$ generated samples. For the sake of simplicity, we use the term “fitness” for the average of fitness as defined above.

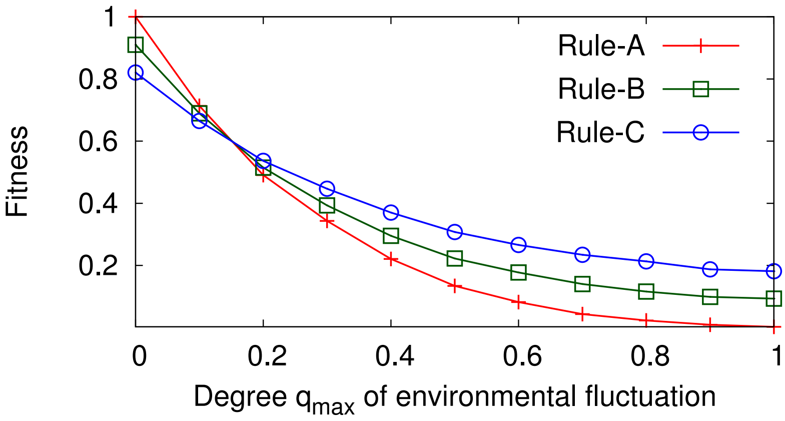
First, we investigate the case with $d = 2$ where paths are not simultaneously affected by environmental fluctuation but the lower-ranked paths are longer than the shortest path. Figure 5.3(a) shows that Rule–A is the most effective to obtain the highest fitness when the degree q_{max} of environmental fluctuation is near 0. By prioritizing increased optimality, Rule–A sacrifices its robustness. This leads to difficulty in maintaining fitness as the degree of fluctuation q_{max} increases. At a certain point, the fitness achieved under Rule–A is exceeded by the fitness under Rule–B and Rule–C. Specifically, when the degree q_{max} is about 0.15, the fitness under Rule-A drops to the fitness achieved with under other rules. As the degree of q_{max} becomes higher, the difference becomes much clearer.

Next, we investigate the case when $d = 15$ where path candidates are equally short but the path candidates are apt to be simultaneously affected by environmental fluctuation. Figure 5.3(b) shows that a higher degree q_{max} leads to the deterioration of fitness, regardless of the chosen rule, but Rule–A always results in the highest fitness. This is because the path candidates are spatially close each other, and so they are apt to be simultaneously affected by environmental fluctuation. Thus, each path is disconnected at a similar rate. In such cases, the path candidates are prepared just only considering the increase in the optimality. These cases do not produce an effect of the disordering energy. As a result, an increase in the robustness does not contribute to keeping good fitness against an increase in the degree of fluctuation q_{max}. The detail about this will be discussed in the next section, but it is here better to prioritize optimality, that is, ensuring that the source node selects the shortest path. Rule–A, which has the highest optimality, therefore succeeds in obtaining the highest fitness.

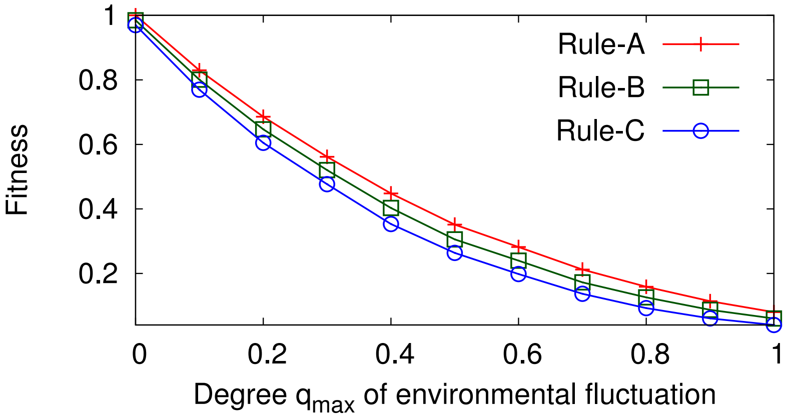
These results imply that we should design multi-path routing so that higher robustness is sought when the path candidates include paths insensitive to environmental fluctuation and paths that are likely to work in a more frequently fluctuating environment. In the next section, we establish a design policy for network control from the perspective of thermodynamic free energy.

5.5 Free Energy-based Design Policy

We establish an appropriate policy for designing a network control system that balances optimality against robustness according to the degree of environmental fluctuation, which we assume will occur. For this purpose, we first abstract the features of network control. Next, we build an analytical



(a) Node density $d = 2$



(b) Node density $d = 15$

Figure 5.3: Impact of environmental fluctuation on the average fitness

model. Then, we formulate that analytical model by a free-energy model. Finally, we analyze the degree of free energy and construct a design policy for the network control.

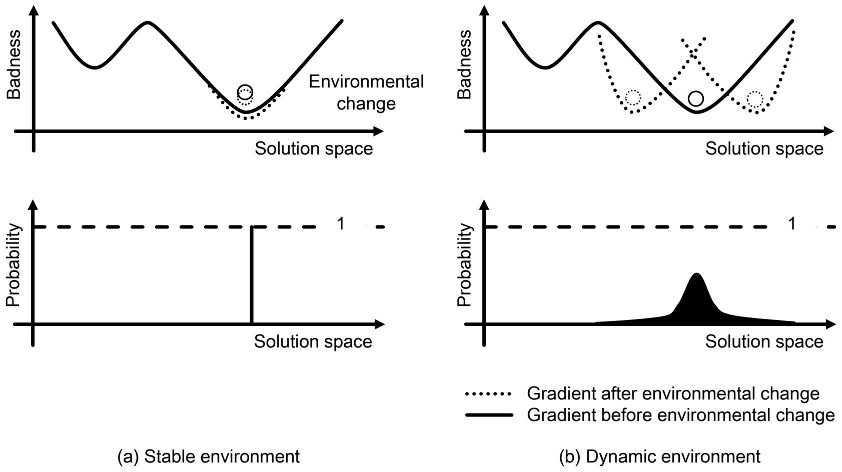


Figure 5.4: Abstract image of bio-inspired network controls

5.5.1 Abstraction of Network Control

Figure 5.4 is an abstract image of a network control. In the figure, the x-axis corresponds to the solution space, which is searched for a better solution, and the y-axis corresponds to the goodness of the discovered solution. The solution's goodness is hereinafter called its "gain". Each black circle denotes a state of the network control at a certain time. Taking multi-path routing as a network control, a solution x corresponds to a path, and its gain corresponds to the shortness of the path. Solution affected by environmental fluctuation correspond to paths disconnected by link errors, node failures, and so on.

The network control eventually stabilizes on the solution with the highest gain at a certain time. In environments without fluctuation, the gradient of gain does not change. When this is the case, it is the best action for the network control to stay on the solution, as shown in Figure 5.4 (a). However, actual networks are typically affected by environmental fluctuation, and so the gradient of gain changes dynamically. To suppress the influence of environmental fluctuation, the network control must avoid stabilizing on the solution with the highest gain at a particular moment in time. It is important that the network control select additional solutions, even if those have lower gain at a certain time, as shown in Figure 5.4 (b).

5.5.2 Analytical Model of Abstract Network Control

We describe an analytical model of the abstract network control. In this analytical model, the gain of a solution x is given by a Gaussian function $g(x) = g_{max} \times \exp[-x^2/2\sigma_2^2]$ when the solution x is not affected by environmental fluctuation. The gain is 0 when the solution x is affected by environmental fluctuation. Here, the coefficient g_{max} $(0 \leq g_{max} \leq 1)$ sets a maximum value for the gain. A solution x is affected by environmental fluctuation per unit time with probability $q(x) = q_{max} \times \exp[-x^2/2\sigma_3^2]$. Thus, the solution x is, on average, affected by environmental fluctuation at intervals of $\exp[x^2/2\sigma_3^2]/q_{max}$. Here, coefficient q_{max} $(0 \leq q_{max} \leq 1)$ indicates the maximum degree of environmental fluctuation. The solution x is selected with probability $p(x) = \exp[-x^2/2\sigma_1^2]/\sqrt{2\pi\sigma_1^2}$. The descriptions of these parameters are summarized in Table 5.3.

5.5.3 Definition of Free Energy of Analytical Model

In Section 5.3, we stated that, in a fluctuation environment, the free energy A is identical to the performance G that is actually achieved by the network control. For the analytical model, the performance G can be formulated as the following equation:

$$\begin{aligned} G &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma_1^2}} \times e^{-\frac{x^2}{2\sigma_1^2}} \times \Bigg\{ g_{max} \times e^{-\frac{x^2}{2\sigma_2^2}} \times \\ &\qquad \left(1 - q_{max} \times e^{-\frac{x^2}{2\sigma_3^2}}\right) + 0 \times q_{max} \times e^{-\frac{x^2}{2\sigma_3^2}} \Bigg\} dx \\ &= g_{max} \times \sqrt{\frac{\sigma_2^2}{\sigma_2^2 + \sigma_1^2}} - q_{max} \times \\ &\qquad g_{max} \times \sqrt{\frac{\sigma_2^2\sigma_3^2}{\sigma_2^2\sigma_3^2 + \sigma_1^2\left(\sigma_2^2 + \sigma_3^2\right)}}. \end{aligned} \tag{5.3}$$

Here, the first term on the right side is the maximum performance that can be achieved in an environment without fluctuation. This term corresponds to optimality. The second term indicates performance degradation due to environmental fluctuation. This term corresponds to robustness (see Section 5.3.2 for the precise definitions of optimality and robustness). The aim of this chapter is to

Table 5.3: Parameters of analytical model

Symbol	Description
q_{max}	Maximum degree of environmental fluctuation
g_{max}	Maximum gain of solution
σ_1^2	Randomness in selecting a solution
σ_2^2	Abundance of good solutions
σ_3^2	Extent of influence of environmental fluctuation

present an appropriate design for a network control that accounts for the degree of environmental fluctuation, with the intent of preparing the network control for fluctuation. To meet this objective, we regard the coefficient q_{max} as the expected degree of environmental fluctuation. In the following, we discuss a design appropriate for a network control that accounts for environmental fluctuation.

5.5.4 Design Policy Depending on Degree of Fluctuation

Maximizing the performance G is identical to balancing the robustness, which corresponds to the second term, with the optimality, which corresponds to the first term, for a given degree q_{max} of environmental fluctuation. To achieve this, we first describe how to maximize the optimality. Then, we describe how to maximize the robustness. Finally, we consider an appropriate balance between the optimality and the robustness according to the expected degree q_{max} of environmental fluctuation.

- **For maximizing optimality**

The first term on the right side of Equation (5.3) indicates the maximum value of the performance G and is identical to the performance achieved by the network control in an environment without fluctuation. Therefore, we adjust the variance σ_1^2 depending on the path candidates, which depend on the variance σ_2^2, so as to maximize the optimality. To do this, the variance σ_1^2 must fall when the variance σ_2^2 does. This suggests that the network control will more aggressively select solutions with larger gain as the solution candidates include more solutions with smaller gain.

- **For maximizing robustness**

The second term on the right side of Equation (5.3) indicates the degradation of the performance G due to environmental fluctuation. Larger values for this term imply weaker robustness against environmental fluctuation. Therefore, to maximize the robustness, we need to minimize the second term for the solution candidates by parameter tuning. To minimize the second term, we reduce the variance σ_2^2 as the variance σ_3^2 grows. If we cannot reduce σ_2^2 due to restrictions on the network environment (e.g., such as those discussed for the network topology and node density d in Section 5.4), then σ_1^2 should be made larger. These relations suggest that we should prepare solution candidates having lower gain because the influence of environmental fluctuation will be relatively suppressed in those solutions. Furthermore, when it is not possible to prepare these solutions, it is better for the network control to increases its randomness in selecting solutions so that selecting a solution susceptible to environmental fluctuation becomes less likely.

- **For balancing robustness with optimality**

We regard Equation (5.3) as a one-dimensional function of the degree q_{max} of environmental fluctuation. Then, we expect that an appropriate balance between the robustness and the optimality exists and depends on the degree q_{max}. As examples, when the degree q_{max} is small, such as when $q_{max} = 0$, it is more effective to prioritize maximizing optimality, that is, to make the variance σ_1^2 smaller. When the degree q_{max} is large, such as when $q_{max} = 1$, it is more effective to prioritize maximizing robustness, that is, to make the variance σ_1^2 larger. These cases imply that we must appropriately choose a balance between optimality and robustness, and this balance will depend on the expected degree of environmental fluctuation.

Using numerical examples, we first show that the degree q_{max} of environmental fluctuation affects which should be prioritized for maximization: optimality or robustness. Figure 5.5 shows the numerical examples. In the figure, the x-axis corresponds to the degree $q_{max}$of environmental fluctuation, and the y-axis corresponds to the performance G as obtained from Equation (5.3). Here, the degree g_{max}, the variance σ_2^2, and the variance σ_3^2 are set at 0.5, 30, and 15, respectively. As the parameter setting for the optimality to be prioritized, the variance σ_1^2 is set at 10. As a parameter setting for the robustness to be prioritized, the variance σ_1^2 is set at 30. The figure depicts the results. From this figure, we see that the robustness should be prioritized when $q_{max} > 0.6$. This shows

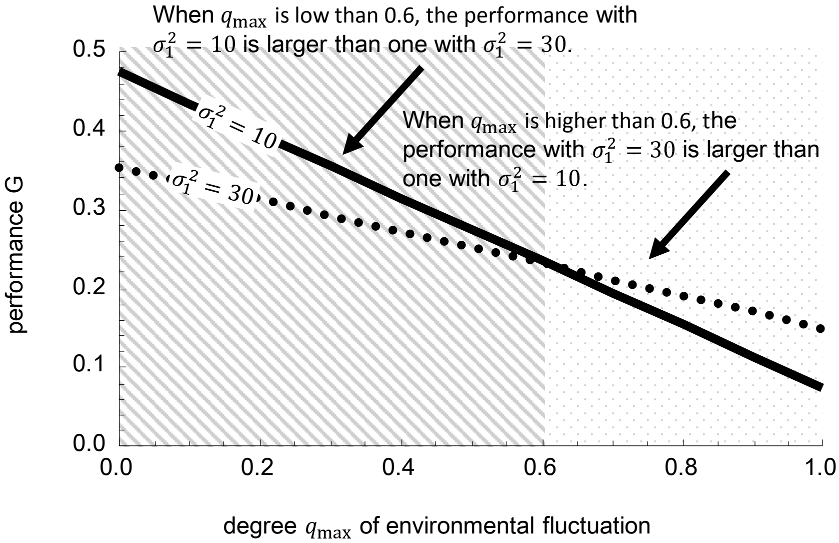
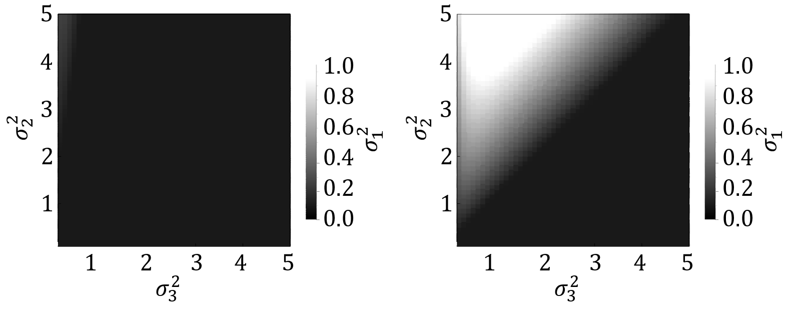


Figure 5.5: Tradeoff between robustness and optimality

that the appropriate balance depends on the degree q_{max} of environmental fluctuation.

Next, we investigate the appropriate balance according to the degree q_{max} of environmental fluctuation and the solution candidates. For this purpose, we numerically derive the variance σ_1^2 at which the performance G is maximized when the parameters σ_2^2, σ_3^2, g_{max}, and q_{max} are fixed. For the testing, the variances σ_2^2 and σ_3^2 range from 0.1 to 5.0 in steps of 0.1. The coefficient g_{max} is taken as 1, and the coefficient q_{max} is taken as 0.125 or 0.5. Altogether, this results in $5,000$ sets of parameter values. Figure 5.6 shows the results. Similarly, the results for the maximum performance G are shown in Figure 5.7. In each figure, the x-axis is the variance σ_3^2, and the y-axis is the variance σ_2^2. Shading intensity indicates the variance σ_1^2 (Figure 5.6) or the performance G (Figure 5.7). In each figure, lighter shades indicate higher values.

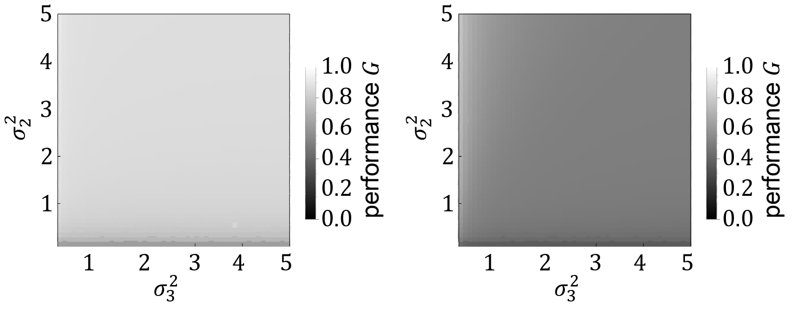
Figure 5.6(a) shows that shading for a wide range of gradients is darker when the degree q_{max} is set at 0.125. This result suggests that the maximization of the optimality in case of $q_{max} = 0.125$ is prioritized so as to maximize the performance G. In contrast, the shading is lighter when the degree q_{max} is 0.5, as shown in Figure 5.6(b). This result suggests that when $q_{max} = 0.5$, robustness is more heavily prioritized. In the analytical model, the robustness is sacrificed by enhancing the optimality. To achieve a balance between robustness and optimality when q_{max} is 0.5, it is better



(a) $q_{max} = 0.125$

(b) $q_{max} = 0.500$

Figure 5.6: Optimal value of variance σ_1^2



(a) $q_{max} = 0.125$

(b) $q_{max} = 0.500$

Figure 5.7: Optimal value of the performance G

that the variance σ_1^2 is set below the maximum value of 5; here, it is set at 1. Additionally, we can notice a trend in which the shading becomes lighter when the variance σ_2^2 is higher than the variance σ_3^2. This results implies that the optimality needs to be prioritized if the solution candidates include solutions that have higher gain and are not susceptible to environmental fluctuation. In contrast, we can also see a trend in which the shading becomes darker when the variance σ_2^2 is less than the variance σ_3^2. The solution-selection rule does not increase the performance G when almost all solution candidates are susceptible to environmental fluctuation. In this case, it is better to prioritize optimality in order to maximize the performance G, such as Rule–A in Figure 5.3(b).

5.6 Verification of Design Policy

We verify the free-energy-based design policy for network controls. For this purpose, we introduce an existing system for multi-path routing and identify the parameter that changes its balance between optimality and robustness. Then, we investigate the appropriate balance between the characteristics. Finally, we verify the design policy described in the previous section.

5.6.1 Example of Existing Multi-path Routing

We take an attractor selection model for multi-path routing [44, 75] as an example of an existing system for multi-path routing.

5.6.1.1 Mechanism of Path Selection

We make two assumptions. The first is that K node-disjoint paths are constructed between a source node and a destination node in advance. The second is that the source node knows the addresses of all wireless nodes on all node-disjoint paths. A path $j \in \{1, 2, \cdots, K\}$ has a state value m_j (≥ 0). At an interval of I_c, the source node selects the path with the maximum state value from among all paths. The state value m_j evolves according to the following equation:

$$\dot{m_i} = \left(\frac{\beta \times \alpha^{\gamma} + \frac{1}{\sqrt{2}}}{1 + \max_{1 \leq j \leq K} m_j^2 - m_i^2} - m_i \right) \times \alpha + \eta_i. \tag{5.4}$$

Here, there are K attractors that depend on which of the state values is the largest. The coefficient β (> 0) sets the maximum depth of attractors, and the coefficient γ (> 0) sets the magnitude of attraction by attractors. The term η_i is a random value with mean 0 and variance 1 (i.e., stochastic noise). The variable α $(0 \leq \alpha \leq 1)$ is the goodness of the multi-path routing. The details are described later, but, broadly, the variable α becomes larger as the source node more stably selects a shorter and better connected path. We hereinafter call this variable “activity”.

5.6.1.2 Derivation of Activity

The source node calculates the activity α based on the path length selected by the source node and its connectivity.

To obtain a path's connectivity, the source node observes its connectivity in a periodic manner. For this purpose, the source node sends an "observation packet" at intervals of I_s along the path selected by the above-mentioned mechanism of path selection. In each observation packet, the source node stores a list of the addresses of wireless nodes on the path. The relay nodes iteratively forward the packet by following the list. When the destination node receives the packet, the destination node sends a "notification packet" back to the source node. This notification packet is iteratively forwarded along the reversed path of the corresponding observation packet. If the source node does not receive the notification message before it sends the next observation packet, the source node assumes that the path was disconnected by environmental fluctuation. Even if the source node later receives the notification packet, the packet will be dropped.

From the results of observation, the source node updates the activity α at an interval of I_a as follows:

$$\dot{\alpha} = \delta \times \left(\frac{L_{min}}{L_{now}} - \alpha \right). \tag{5.5}$$

In this equation, the coefficient δ is a smoothing coefficient; L_{min} (> 0) is the minimum hop length of connected paths within the last I_h observations; and L_{now} (> 0) is the hop length of the last-observed path. Note here that L_{min}/L_{now} is 0 when the last-observed path was not connected.

5.6.1.3 Behavior of Multi-path Routing

The source node sends a "data packet" at intervals of I_d [s] along the selected path, which is determined by the attractor selection mechanism. As the source node selects a worse path, such as a longer path or a disconnected path, the activity α gradually decreases. Simultaneously, the relative influence of the noise term η_i in Equation (5.4) on the change of the state value m_i becomes larger. The magnitude relationship of the state values is apt to be changed by the noise term, and so the source node selects a path more randomly in its search for a better path. As the source node finds better paths, the activity α gradually becomes larger. Simultaneously, the relative influence of the first term of Equation (5.4) on the change of the state value m_i becomes larger. The state value of the found path increases, and the other state values decrease. Eventually, the magnitude relationship of the state values becomes stable, and the source node selects a specific path more stably.

5.6.1.4 Parameter for Balance between Robustness and Optimality

In the attractor selection model of multi-path routing, the parameter β affects which of optimality and robustness is prioritized. Specifically, as parameter the β becomes smaller, the source node becomes more likely to stabilize on only the shortest path. Therefore, we can regard the maximization of optimality as the minimization of the parameter β. In the other direction, as the parameter β becomes larger, the source node becomes more likely to stabilize on a path without regard to its hop length, increasing the chance that communication between the source node and the destination node will be maintained. This is because the source node can easily stabilize on connected paths regardless of hop length. Thus, we can regard the maximization of robustness as the maximization of the parameter β. In the following, we therefore call the parameter β the “balance parameter”.

5.6.2 Simulation Experiment

We verify the free-energy-based design policy through simulation experiments using the attractor selection model of multi-path routing. For this purpose, we derive an appropriate balance between robustness and optimality by investigating the approach-balancing parameter β. Comparing the results obtained by simulation with those described in Section 5.5.4, we verify the design policy.

5.6.2.1 Simulation Setting

Using the simple model of a multi-path network described in Section 5.4, we prepare a wireless network and path candidates. The parameter values are set as follows. We evaluate two node densities, $d = 2$ and $d = 15$. In the case of node density $d = 2$, there are path candidates that are not simultaneously affected by environmental fluctuation, but almost all of the path candidates

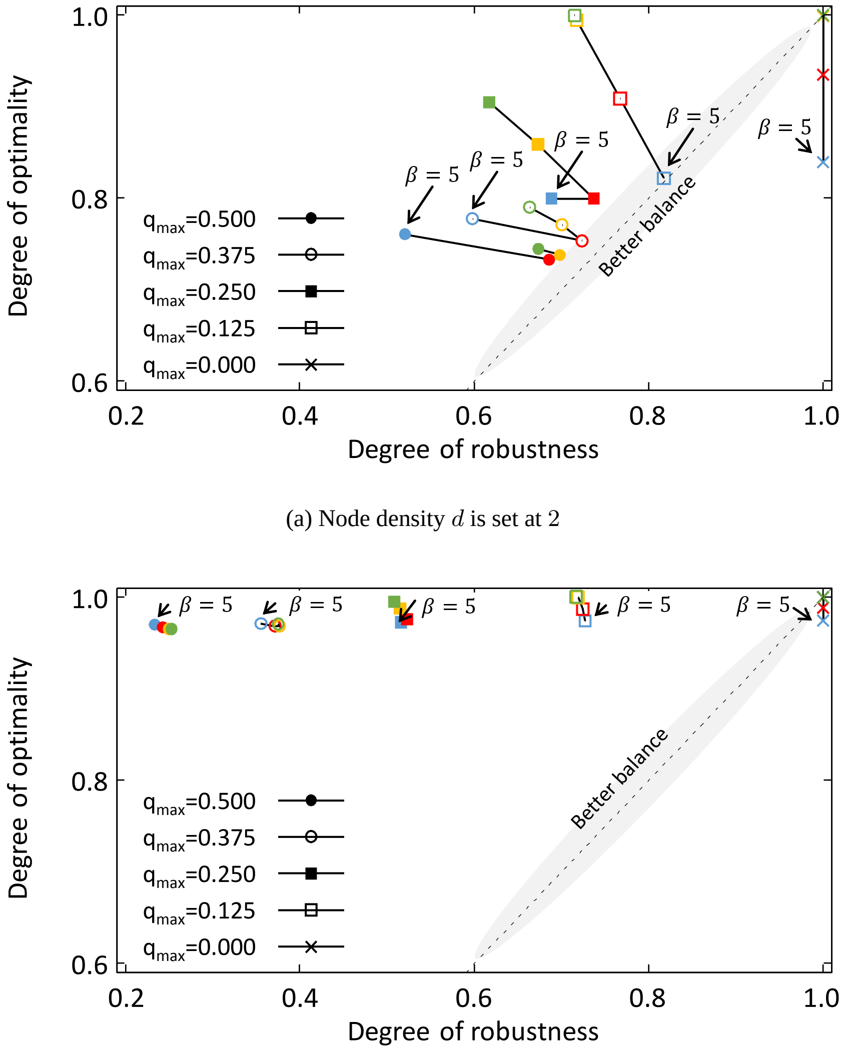
Table 5.4: Parameter setting

Symbol	Value	Description
I_c	1 [s]	Interval between path selection
I_s	1 [s]	Interval for observing connectivity
I_a	1 [s]	Interval for updating activity
I_d	1 [ms]	Interval between sending of data packets
I_h	10 packets	Length of history
γ	1.0	Magnitude of attraction
δ	0.1	Smoothing coefficient

are longer than the shortest path. In the other case ($d = 15$), there are path candidates as short as the shortest path, but the path candidates are simultaneously affected by environmental fluctuation. The degree q_{max} of environmental fluctuation ranges from 0.0 to 0.5 at intervals of 0.125. The variance σ^2 is set at 1. The balance parameter β ranges from 5 to 20 at intervals of 5. As the balance parameter β becomes larger, the maximization of the robustness becomes more prioritized. The other parameters are set as shown in Table 5.4. The initial state vector $\vec{m}$ is set at $\vec{m} = (\beta, 0, 0)$. The initial activity α is set at 1. In this parameter setting, the shortest path is stably selected by the source node just after a simulation begins. The following simulation results are the average values from across 500 simulation runs, where the duration of a single run is $1,000$ [s].

5.6.2.2 Simulation Result

Figure 5.8 shows the results. In the figure, the x-axis corresponds to the ratio of the number of data packets received by the destination node relative to the number of data packets sent by the source node during a simulation run. This ratio measures the robustness, which here means the degree to which communication is maintained despite environmental fluctuation. We hereinafter call this ratio the “degree of robustness”. The y-axis corresponds to the relative length of the path used to send the data packet to the destination node. The relative length is defined as n_j/n_1, where n_j is the hop length of path j, and path 1 is the shortest path. This value corresponds to the optimality, which measures the quality of communication. This value is hereinafter called the “degree of optimality”. The balance between optimality and robustness is taken as better when the degree of robustness is



(a) Node density d is set at 2

(b) Node density d is set at 15

Figure 5.8: Influence of balance parameter β on balance between robustness and optimality

closer to the degree of optimality and both are higher. The balance parameter β is distinguished in the figure by symbol color. The blue, red, orange, and green symbols denote results for $\beta = 5$, 10, 15, and 20, respectively.

First, we discuss the simulation results when we can obtain path candidates that are not simultaneously affected by environmental fluctuation, although they include longer paths. Figure 5.8(a) shows that it is better that optimality is more prioritized when environmental fluctuation occurs less frequently. From this figure, we can see that a parameter value of $\beta = 15$ achieves the best balance between optimality and robustness when the degree q_{max} of environmental fluctuation is 0.5. This result agrees with the suggestion that the maximization of robustness is prioritized, that is, the balance parameter β is best set at larger values when environmental fluctuation occurs more frequently. However, it is not always better to increase β. In this simulation setting, the best value for β is not the maximum value of 20; it is 15. As the degree q_{max} becomes smaller than 0.5, the best value for the balance parameter β becomes smaller. This implies that it is good to prioritize optimality more strongly as the degree q_{max} becomes smaller. Here, when the degree q_{max} is set at 0.375 and 0.25, the setting the parameter β to 10 achieves the best balance. In addition, setting β to 5 achieves the best balance when the degree q_{max} is set at 0.125. However, there is an inconsistency in this figure. The best value for β is 20 when the degree q_{max} is set at 0, that is, when environmental fluctuation does not occur. In this simulation setting, just after the simulation begins, the source nodes stably selects the shortest path. When the degree q_{max} is set at 0, the shortest path is always connected. In this case, the best balance between optimality and robustness is achieved by the source node continuing to select the shortest path until the simulation finishes. Thus, the best value for β is 20, the maximum possible value in this simulation setting; however, this depends on the initial setting of the simulation.

Next, we discuss the simulation result when we can obtain a set of path candidates that include equally short paths as the shortest path but candidates are simultaneously affected by environmental fluctuation. The simulation results are shown in Figure 5.8(b). This figure shows that the value of the balance parameter β does not affect the balance between optimality and robustness, and so no single appropriate balance exists. When the node density d is 15, the path candidates lie in a narrow area near the shortest path, and each path is disconnected with similar frequency. A rule for path selection therefore does not contribute to maintaining communication between the source node and the destination node. That is, we cannot improve robustness by adjusting the balance parameter β. Therefore, there is no best balance between robustness and optimality.

In conclusion, the balance parameter β should be set at a larger value for the network control to be equipped with the ability to endure more frequent environmental fluctuation, but this is helpful only when the path candidates include paths not simultaneously affected by environmental fluctuation. This suggestion agrees with the free-energy-based design policy described in the previous section.

5.7 Summary

We formulated and analyzed a free-energy model of network control. Then, we established a design policy based on the analytical results. The obtained policy suggests that network control should be designed to improve its robustness in cases where it is deployed in more dynamic environments and has solution candidates that are insusceptible to environmental fluctuation.

Chapter 6

Conclusion

A bio-inspired network control is a promising approach to realize resilient wireless sensor and actuator networks (WSANs). In this thesis, we took two steps in order to realize resilient WSANs. The first step is that we realize bio-inspired robust mechanisms for WSANs. The second one is that we propose a design policy to equip a bio-inspired WSAN control with the appropriate adaptability depending on its design requirements.

In Chapter 2, we first studied the robustness of bio-inspired WSAN controls against location error of nodes and shape error of sensing areas caused by environmental noise. For this purpose, we proposed and verified the attractor selection model-based coverage control. Most of existing coverage protocols require accurate information about the location, sensing area, and sensing state of neighbor nodes. Therefore, they suffer from the errors leading to degradation of coverage and redundancy of active nodes. In addition, they introduce communication overhead leading to energy depletion. On the other hand, our proposal enables autonomous decision on nodes without accurate location information and communication with neighbor nodes by adopting the attractor select model, the nonlinear mathematical model of adaptive behavior of biological systems to dynamically changing environment. Through simulation experiments, we showed that our proposal can achieve the sensing ratio of up to 0.98 and prolong the life time of the network up to 6-fold by comparison with CCP. In addition, it was also shown that our proposal is superior to CCP when the maximum location error is larger than 6 m. Results in this chapter support that the bio-inspired

WSAN controls are equipped with the high robustness against the errors.

Next, we focused on the robustness of bio-inspired WSAN controls against parameter setting in Chapter 3. For this purpose, we proposed and verified the response threshold model-based device assignment control. The existing mechanisms suffer from difficulty in designing an appropriate set of rules with fine-tuned parameters. Our proposal realizes the fully distributed and self-organizing device assignment mechanism by adopting the response threshold model, which imitates division of labors in a colony of social insects. The deterministic and complicated rules are not required, and an appropriate device assignment emerges as a consequence of autonomous decision of individual nodes. Through simulation, we confirmed that our proposal accomplishes as effective device assignment as an existing deterministic mechanism and is less sensitive to parameter setting errors. Results obtained from this chapter show that bio-inspired WSAN controls are robust against parameter setting.

Then, in Chapter 4, we studied the robustness of bio-inspired WSAN controls against node failures in lossy environment. For this purpose, we analyzed the robustness of the response threshold model against individual failures in lossy environment. The original response threshold model as a mathematical model of biological behavior is built under an assumption that there is no dead individual and no information loss. It is not clear yet how individual failures and information loss affect the steady and transient states. Therefore, we built its analytical model and conducted mathematical analysis in order to show the robustness against individual failures in lossy environment. Through analysis and simulation experiment, we confirmed that the number of workers recovers from individual failures even if dead and isolated individuals exist. This implies that the robustness originates from abundance of states that can be taken. Moreover, we also confirmed that the recovery time does not diverge as more individuals die. Results in this chapter imply that bio-inspired WSAN controls are robust against node failures.

Finally, we established and verified the design policy for a bio-inspired WSAN control to have the sufficient adaptability depending on the expected degree of environmental fluctuation in Chapter 5. We focused on the fact that the goodness of its state can be analyzed in terms of thermodynamic free energy. Therefore, we formulated and analyzed the free energy of a bio-inspired WSAN control. We established the design policy based on the analytical results. The obtained policy suggests that it is better that the randomness increases as the network control is expected to work in environment with more frequent fluctuation if the solution candidates include those which are insusceptible to environmental fluctuation and have higher gain.

Throughout this thesis, we confirmed that bio-inspired WSAN controls contribute to realizing resilient WSANs. However, it is also shown that bio-inspired WSAN controls are not necessarily versatile. From simulation results, we clearly find that there is a turning point where their superiority or inferiority to conventional network controls are reversed. As our future work, we need a method to clarify an application area of a bio-inspired network control by identify its turning point. In addition, origins of resilience of biological systems are not well solved, and the recent technical progress advances investigation into unknown mechanisms, which have not been studied in detail. Thus, there is room to obtain novel ideas and inspirations and realize more resilient WSANs by collaborating with researchers in diverse fields. We hope that this thesis expresses possibility of bio-inspired WSAN controls to researchers all over the world and contributes to accelerating research and development.

Bibliography

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, pp. 351 – 367, Oct. 2004.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, Dec. 2002.

[4] T. S. Rappaport, *Wireless communications: principles and practice*, vol. 207. Prentice Hall PTR New Jersey, 1996.

[5] J. Steffan, L. Fiege, M. Cilia, and A. Buchmann, "Towards multi-purpose wireless sensor networks," in *Proceedings of the International Conference on Systems Communications*, pp. 336–341, Aug. 2005.

[6] A. Majeed and T. A. Zia, "Multi-set architecture for multi-applications running on wireless sensor networks," in *Proceedings of the 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 299–304, Apr. 2010.

[7] Y. Yu, L. J. Rittle, V. Bhandari, and J. B. LeBrun, "Supporting concurrent applications in wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 139–152, Nov. 2006.

[8] M. Dorigo, G. D. Caro, and L. M. Gambardella, “Ant algorithms for discrete optimization,” *Artificial life*, vol. 5, pp. 137–172, Apr. 1999.

[9] R. E. Mirollo and S. H. Strogatz, “Synchronization of pulse-coupled biological oscillators,” *SIAM Journal on Applied Mathematics*, vol. 50, pp. 1645–1662, Dec. 1990.

[10] A. M. Turing, “The chemical basis of morphogenesis,” *Philosophical Transactions of the Royal Society of London. Series (B): Biological Sciences*, vol. 237, pp. 37–72, Aug. 1952.

[11] E. Bonabeau, F. Henaux, S. Guerin, D. Snyers, P. Kuntz, and G. Theraulaz, “Routing in telecommunications networks with ant-like agents,” *Intelligent Agents for Telecommunication Applications*, vol. 1437, pp. 60–71, May 1998.

[12] K. Hyodo, N. Wakamiya, E. Nakaguchi, M. Murata, Y. Kubo, and K. Yanagihara, “Reaction-diffusion based autonomous control of wireless sensor networks,” *International Journal of Sensor Networks*, vol. 7, pp. 189–198, May 2010.

[13] Y. Taniguchi, N. Wakamiya, and M. Murata, “A traveling wave based communication mechanism for wireless sensor networks,” *Journal of Networks*, vol. 2, no. 5, pp. 24–32, 2007.

[14] A. Tyrrell, G. Auer, and C. Bettstetter, “On the accuracy of firefly synchronization with delays,” in *Proceedings of the 1st International Symposium on Applied Sciences on Biomedical and Communication Technologies (ISABEL)*, pp. 1–5, Oct. 2008.

[15] J. Nishimura and E. J. Friedman, “Robust convergence in pulse-coupled oscillators with delays,” *Physical Review Letters*, vol. 106, p. 194101, May 2011.

[16] C. S. Holling, “Resilience and stability of ecological systems,” *Annual review of ecology and systematics*, vol. 4, pp. 1–23, Nov. 1973.

[17] T. Iwai, N. Wakamiya, and M. Murata, “Error-tolerant and energy-efficient coverage control based on biological attractor selection model in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 1–14, Feb. 2012.

[18] T. Iwai, N. Wakamiya, and M. Murata, “Bio-inspired autonomous and adaptive coverage control for wireless sensor networks,” in *Proceedings of the 2nd International Workshop on Sensor Networks and Ambient Intelligence (SeNAmI)*, Sept. 2009.

[19] T. Iwai, N. Wakamiya, and M. Murata, “Error-tolerant coverage control based on bio-inspired attractor selection model for wireless sensor networks,” in *Proceedings of the 10th International Conference on Computer and Information Technology (CIT)*, June 2010.

[20] T. Iwai, N. Wakamiya, and M. Murata, “Proposal and evaluation of attractor selection-based coverage control in wireless sensor networks,” *Technical Report of IEICE (NS2009-120)*, vol. 109, pp. 1–6, Dec. 2009.

[21] T. Iwai, N. Wakamiya, and M. Murata, “Error-tolerant and energy-efficient coverage control based on attractor selection model for wireless sensor networks,” in *Proceedings of IEICE Society Conference*, Sept. 2010.

[22] M. Cardei and J. Wu, “Coverage in wireless sensor networks,” *Handbook of Sensor Networks*, pp. 422–433, 2004.

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

[24] B. Yener, M. Magdon-Ismail, and F. Sivrikaya, “Joint problem of power optimal connectivity and coverage in wireless sensor networks,” *Wireless Networks*, vol. 13, pp. 537–550, Aug. 2007.

[25] R. Zheng, G. He, and X. Liu, “Location-free coverage maintenance in wireless sensor networks,” Tech. Rep. Tech. Rep. UH-CS-05-15, Depertment of Computer Science, University of Houston, July 2005.

[26] T. Iwai, N. Wakamiya, and M. Murata, “Response threshold model-based device assignment for cooperative resource sharing in a WSAN,” *International Journal of Swarm Intelligence and Evolutionary Computation*, vol. 1, pp. 1–12, Apr. 2012.

[27] T. Iwai, N. Wakamiya, and M. Murata, “Proposal for dynamic organization of service networks over a wireless sensor and actuator network,” in *Proceedings of the 2nd International Conference on Ambient Systems, Networks and Technologies (ANT)*, Sept. 2011.

[28] T. Iwai, N. Wakamiya, and M. Murata, “Proposal of autonomous task allocation for dynamic formation of service networks over wireless sensor and actuator networks,” *Technical Report of IEICE (NS2010-219)*, vol. 110, pp. 323–328, Mar. 2011.

[29] T. Iwai, N. Wakamiya, and M. Murata, “A self-organization based device assignment mechanism for cooperative resource sharing in a wireless sensor and actuator network,” *Technical Report of IEICE (IN2012-8)*, vol. 112, pp. 37–42, Apr. 2012.

[30] E. H. Jung and Y. J. Park, “TinyONet: A cache-based sensor network bridge enabling sensing data reusability and customized wireless sensor network services,” *Sensors*, vol. 8, pp. 7930–7950, Dec. 2008.

[31] E. Avilés-López and J. García-Macías, “TinySOA: a service-oriented architecture for wireless sensor networks,” *Service Oriented Computing and Applications*, vol. 3, pp. 99–108, June 2009.

[32] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom)*, pp. 56–67, Aug. 2000.

[33] T. Iwai, N. Wakamiya, and M. Murata, “Characteristic analysis of model of division of labors for self-organizing network control,” *Nano Communication Network (submitted for publication)*, Dec. 2014.

[34] T. Iwai, N. Wakamiya, and M. Murata, “Analysis of response threshold model and its application for self-organizing network control,” in *Proceedings of the 7th International Workshop on Self-Organizing Systems (IWSOS)*, May 2013.

[35] T. Iwai, N. Wakamiya, and M. Murata, "Characteristic evaluation of response threshold model for self-organizing network control," in *Proceedings of IEICE General Conference*, Sept. 2012.

[36] T. Iwai, D. Kominami, M. Murata, and T. Yomo, "Free energy-based design policy for robust network control against environmental fluctuation," *The Scientific World Journal (submitted for publication)*, Dec. 2014.

[37] T. Iwai, D. Kominami, M. Murata, and T. Yomo, "Thermodynamics-based entropy ajustment for robust self-organized network controls," in *Proceedings of the 38th International Computers, Software, and Applications Conference (COMPSAC)*, July 2014.

[38] T. Iwai, D. Kominami, M. Murata, and T. Yomo, "Thermodynamic principle-based strategy to achieve balance between robustness and performance for self-organized network controls," in *Proceedings of the 8th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, Sept. 2014.

[39] T. Iwai, D. Kominami, and M. Murata, "Thermodynamics-based coordinated control for self-organizing information networks," in *Proceedings of IEICE General Conference*, Mar. 2014.

[40] T. Iwai, D. Kominami, M. Murata, and T. Yomo, "Interpretation of self-organized network controls in terms of thermodynamics," in *Proceedings of Workshop on IEICE Technical Committee on Information Network Science (NetSci)*, pp. 1–4, Aug. 2014.

[41] I. M. Klotz and R. M. Rosenberg, *Chemical thermodynamics; basic theory and methods*. Wiley, June 2008.

[42] J. Chen and X. Koutsoukos, "Survey on coverage problems in wireless ad hoc sensor networks," in *Proceedings of IEEE South East Conference*, pp. 22–25, Mar. 2007.

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

[44] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo, “Adaptive response of a gene network to environmental changes by fitness-induced attractor selection,” *PloS ONE*, vol. 1, pp. 1–10, Dec. 2006.

[45] J. Wang, R. K. Ghosh, and S. K. Das, “A survey on sensor localization,” *Journal of Control Theory and Applications*, vol. 8, pp. 2–11, Feb. 2010.

[46] K. Leibnitz, N. Wakamiya, and M. Murata, “A bio-inspired robust routing protocol for mobile ad hoc networks,” in *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN)*, pp. 321–326, Aug. 2007.

[47] J. Lu, L. Bao, and T. Suda, “Probabilistic self-scheduling for coverage configuration in wireless ad-hoc sensor networks,” *International Journal of Pervasive Computing and Communications*, vol. 4, pp. 26–39, Mar. 2008.

[48] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, “Impact of radio irregularity on wireless sensor networks,” in *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 125–138, June 2004.

[49] Crossbow Technology, “MICAz Datasheet.” http://www.xbow.com.

[50] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh, “Simulating the power consumption of large-scale sensor network applications,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 188–200, Nov. 2004.

[51] H. B. Lim, M. Iqbal, and T. J. Ng, “A virtualization framework for heterogeneous sensor network platforms,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 319–320, Nov. 2009.

[52] N. Mohamed and J. Al-Jaroodi, “A survey on service-oriented middleware for wireless sensor networks,” *Service Oriented Computing and Applications*, vol. 5, pp. 71–85, June 2001.

[53] C. Frank and K. Romer, “Algorithms for generic role assignment in wireless sensor networks,” in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 230–242, Oct. 2005.

[54] C. Frank and K. Romer, “Solving generic role assignment exactly,” in *Proceedings of the 20th International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 25–29, Apr. 2006.

[55] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” *Wireless Networks*, vol. 5, pp. 481–494, Sept. 2002.

[56] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J. L. Deneubourg, “Adaptive task allocation inspired by a model of division of labor in social insects,” in *Proceedings of the 1st International Conference on Biocomputing and Emergent Computation (BCEC)*, pp. 36–45, Jan. 1997.

[57] H. M. N. D. Bandara, A. P. Jayasumana, and T. H. Illangasekare, “Cluster tree based self-organization of virtual sensor networks,” in *Proceedings of the International Workshops on Wireless Mesh and Sensor Networks (WAMSNet)*, pp. 1–6, Nov. 2008.

[58] N. Hoeller, C. Reinke, J. Neumann, S. Groppe, D. Boeckmann, and V. Linnemann, “Efficient XML usage within wireless sensor networks,” in *Proceedings of the 4th International Conference on Wireless Internet (WICON)*, pp. 1–10, Oct. 2008.

[59] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, 1999.

[60] A. Varga, “The OMNeT++ discrete event simulation system,” in *Proceedings of the 15th European Simulation Multiconference (ESM)*, pp. 319–324, June 2001.

[61] M. Sasabe, N. Wakamiya, M. Murata, and H. Miyahara, “Media streaming on P2P networks with bio-inspired cache replacement algorithm,” in *Proceedings of the 1st International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT)*, pp. 380–395, May 2004.

[62] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron, and S. Camazine, “Self-organization in social insects,” *Trends in Ecology and Evolution*, vol. 12, pp. 188–193, May 1997.

[63] A. Duarte, I. Pen, L. Keller, and F. J. Weissing, “Evolution of self-organized division of labor in a response threshold model,” *Behavioral Ecology and Sociobiology*, vol. 66, pp. 947–957, Mar. 2012.

[64] J. F. G. Auchmuty and G. Nicolis, “Bifurcation analysis of nonlinear reaction-diffusion equations I. evolution equations and the steady state solutions,” *Bullebtin of Mathematical Biology*, vol. 37, pp. 323–365, Aug. 1975.

[65] W. J. Gutjahr, “A graph-based ant system and its convergence,” *Future Generation Computer Systems*, vol. 16, pp. 873–888, June 2000.

[66] C. Zheng and D. Sicker, “A survey on biologically inspired algorithms for computer networking,” *IEEE Communications Surveys Tutorials*, vol. PP, pp. 1–32, Jan. 2013.

[67] Z. Zhang, K. Long, J. Wang, and F. Dressler, “On swarm intelligence inspired self-organized networking: its bionic mechanisms, designing principles and optimization approaches,” *IEEE Communications Surveys Tutorials*, vol. PP, pp. 1–25, July 2013.

[68] T. H. Labella and F. Dressler, “A bio-inspired architecture for division of labour in SANETs,” *Advances in Biologically Inspired Information Systems*, vol. 69, pp. 209–228, Dec. 2007.

[69] P. Janacik, T. Heimfarth, and F.-J. Rammig, “Emergent topology control based on division of labour in ants,” in *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 733–740, Apr. 2006.

[70] F. Szidarovszky and T. Bahill, *Linear system theory, second edition*. CRC Press, 1997.

[71] T. Hasegawa, “A survey of the research on future internet and network architectures,” *IEICE Transactions on Communications*, vol. E96-B, pp. 1385–1401, June 2013.

[72] S. Balasubramaniam, K. Leibnitz, P. Liò, D. Botvich, and M. Murata, “Biological principles for future internet architecture design,” *IEEE Communications Magazine*, vol. 49, pp. 44–52, July 2011.

[73] F. Heylighen, “The science of self-organization and adaptivity,” *The Encyclopedia of Life Support Systems*, vol. 5, no. 3, pp. 253–280, 2001.

[74] Y. Jin and B. Sendhoff, “Trade-off between performance and robustness: An evolutionary multiobjective approach,” in *Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, vol. 2632, pp. 237–251, Apr. 2003.

[75] K. Leibnitz, N. Wakamiya, and M. Murata, “Biologically inspired self-adaptive multi-path routing in overlay networks,” *ACM Communications*, vol. 49, pp. 62–67, Mar. 2006.

[76] G. D. Caro, “Ant colony optimization and its application to adaptive routing in telecommunication networks,” *Ph.D. Thesis in Applied Sciences, Polytechnic School, Université Libre de Bruxelles, Brussels, Belgium*, Sept. 2004.

[77] R. Pagliari and A. Scaglione, “Scalable network synchronization with pulse-coupled oscillators,” *IEEE Transactions on Mobile Computing*, vol. 10, pp. 392–405, Mar. 2011.

[78] C. E. Torres, L. F. Rossi, J. Keffer, K. Li, and C. C. Shen, “Modeling, analysis and simulation of ant-based network routing protocols,” *Swarm Intelligence*, vol. 4, pp. 221–244, July 2010.

[79] R. Holzer and H. de Meer, “Quantitative modeling of self-organizing propertie,” in *Proceedings of the 3rd International Workshop on Self-Organizing Systems (IWSOS)*, pp. 149–161, Dec. 2009.

[80] R. Holzer and H. de Meer, “Methods for approximations of quantitative measures in self-organizing systems,” in *Proceedings of the 5th International Workshop on Self-Organizing Systems (IWSOS)*, pp. 1–15, Feb. 2011.

[81] M. Jacyno and S. Bullock, “Energy, entropy and work in computational ecosystems: a thermodynamic account,” in *Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems (ALIFE)*, pp. 274–281, July 2008.

[82] H. V. D. Parunak and S. Brueckner, “Entropy and self-organization in multi-agent systems,” in *Proceedings of the 5th International Conference on Autonomous Agents (AGENTS)*, pp. 124–130, May 2001.

[83] J. Branke, M. Mnif, C. Müller-Schloer, H. Prothmann, U. Richter, F. Rochner, and H. Schmeck, “Organic computing – addressing complexity by controlled self-organization,” in *Proceedings of the 6th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*, pp. 185–191, Nov. 2006.

[84] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Controlled and self-organized routing for large-scale wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 10, no. 13, pp. 1–27, 2013.

[85] C. Lim, S. Bohacek, J. P. Hespanha, and K. Obraczka, “On the effectiveness of proactive path-diversity based routing for robustness to path failures,” in *Proceedings of the 7th International IFIP–TC6 Networking Conference on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet (NETWORKING)*, vol. 4982, pp. 574–585, May 2008.