Managed Self-Organization Control for Robust Wireless Sensor Networks

January 2013

Daichi KOMINAMI

Managed Self-Organization Control for Robust Wireless Sensor Networks

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2013

Daichi KOMINAMI

List of publication

Journal papers

1. Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi, "Energy-Efficient Receiver-Driven Wireless Mesh Sensor Networks," *Sensors*, vol. 11, no. 1, pp. 111–137, December 2010.

2. Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi, "Robust and Resilient Data Collection Protocols for Multihop Wireless Sensor Networks," *IEICE Transactions on Communications*, vol. E95-B, no. 9, pp. 2740–2750, September 2012.

3. Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi, "Controlled and Self-Organized Routing for Large-Scale Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, December 2012, accepted.

Refereed Conference Papers

1. Daichi Kominami, Masashi Sugano, Masayuki Murata, Takaaki Hatauchi, and Yoshikazu Fukuyama, "Performance Evaluation of Intermittent Receiver-Driven Data Transmission on Wireless Sensor Networks," in *Proceedings of the 6th International Symposium on Wireless Communication Systems (ISWCS 2009)*, pp. 141–145, September 2009.

2. Daichi Kominami, Masashi Sugano, Masayuki Murata, Takaaki Hatauchi, and Junichi Machida, "Energy Saving in Intermittent Receiver-Driven Multi-Hop Wireless Sensor Networks," in

Proceedings of the 3rd IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (IEEE SUTC 2010), pp. 296–303, June 2010.

3. Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi, "Robustness of Receiver-Driven Multi-Hop Wireless Network with Soft-State Connectivity Management," in *Proceedings of the 5th International Conference on Systems and Networks Communications (ICSNC 2010)*, pp. 46–51, August 2010.

4. Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi, "Controlled Potential-Based Routing for Large-Scale Wireless Sensor Networks," in *Proceedings of the 14th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (ACM MSWiM 2011)*, pp. 187–195, November 2011.

Non-Refereed Technical Papers

1. Daichi Kominami, Masashi Sugano, Masayuki Murata, Takaaki Hatauchi, Yoshikazu Fukuyama, and Tatsuya Shikura, "Evaluation of Intermittent Receiver-Driven Data Transmission on Wireless Sensor Networks," *Technical Report of IEICE (IN2008-155)*, vol. 108, no. 458, pp. 139–144, March 2009 (in Japanese).

2. Daichi Kominami, Masashi Sugano, Masayuki Murata, Takaaki Hatauchi, and Junichi Machida, "Performance Improvement by Collision Avoidance Mechanism in Receiver-Driven Multi-Hop Wireless Networks," *Technical Report of IEICE (AN2009-32)*, vol. 109, no. 247, pp. 65–70, October 2009 (in Japanese).

3. Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi, "Robustness of Receiver-Driven Multi-Hop Wireless Network with Soft-State Connectivity Management," *The Papers of Technical Meeting on Information Systems, IEE Japan (IS-10-038)*, pp. 81–86, May 2010 (in Japanese).

4. Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi, "Robustness of Intermittent Receiver-Driven Wireless Networks Against Fluctuations of Wireless Channel

Quality," *Technical Report of IEICE (AN2010-21)*, vol. 110, no. 129, pp. 63–68, July 2010 (in Japanese).

5. Daichi Kominami, Masashi Sugano, Masayuki Murata, and Takaaki Hatauchi, "Controlled Potential-Based Routing for Large-Scale Wireless Sensor Networks," *Technical Report of IEICE (AN2010-48)*, vol. 110, no. 377, pp. 25–30, January 2011 (in Japanese).

6. Daichi Kominami and Masayuki Murata, "A Design Approach for Controlled and Self-Organized Networks Focused on Control Timescale," *IEICE Technical Committee on Information Network Science (NetSci)*, August 2012 (in Japanese).

Preface

In recent years, due to advances in wireless and micro-electromechanical technologies, extremely small sensor nodes featuring wireless communication facilities have been developed, and as a result, wireless sensor networks have received considerable attention. In wireless sensor networks, a large number of low-cost sensor nodes with wireless communication capabilities collect various environmental data, such as temperature, light, pressure, humidity, movement, and noise, and they deliver such data toward data-sink nodes. Wireless sensor networks are particularly useful for a wide range of applications as they possess sensing capabilities without the need for implementing a centralized infrastructure. Thus, wireless sensor networks can be thought of becoming increasingly important in the future. Originally, its application is for monitoring a variety of environmental information; however, the needs have been diversified. The concept of "ubiquitous computing" is spread as one form of future networks, and among them, sensor nodes and actuators blending with circumstances provide useful information and services for humans. In order to achieve such networks, wireless sensor networks should play a greater role in observing and processing further more variety of information, and in interacting with environments and humans. However, many problems to be solved in wireless sensor networks still remains. In this thesis, we investigate energy-efficiency, robustness, scalability, and manageability issues for wireless sensor networks.

To begin with, we examine energy efficiency in wireless sensor networks, which consist of devices with limited energy resources. We focus on the sleep control in media access control (MAC) layer protocols, and in particular, we evaluate and improve the intermittent receiver-driven data transmission (IRDT) protocol, which aims at saving energy and achieving reliability. This protocol can save energy by allowing a wireless interface to sleep for a long time when there is no need for transmitting data. Communication between two nodes commences when a receiver node transmits its own identification and a sender node receives it. We clarify the performance characteristics of this protocol by comparing it with the famous two protocols, RI-MAC protocol and X-MAC protocol. Moreover, we improve the IRDT protocol by implementing proactive and reactive collision avoidance methods for control packets. We show that IRDT can offer greater reduction of the average energy consumption compared with RI-MAC and X-MAC, especially at small loads, and we demonstrate that IRDT with collision avoidance for control packets can attain higher performance than the original IRDT. This method ensures a high packet collection ratio and a lower average energy consumption than those of EA-ALPL and those of the original IRDT.

Robustness is one of the significant properties in wireless sensor networks because sensor nodes and wireless links are subjected to frequent perturbations. Once these perturbations occur, system performance falls into critical condition due to increases in traffic and losses of connectivity and reachability. Most of the existing studies on wireless sensor networks, however, do not conduct quantitative evaluation on robustness and do not discuss what brings in robustness. We define and evaluate robustness of wireless sensor networks and show how to improve them. We show that receiver-initiated MAC protocols, one of which is the IRDT protocol, are more robust than sender-initiated ones, and a simple detour-routing algorithm has much more robustness than the simple minimum-hop routing algorithm due to their memoryless property for the condition of communication.

The following part refers to improvement in scalability of wireless sensor networks. Much research on self-organization has been conducted toward this end. In self-organization schemes, entirely local information is used for decision-making by each node. This interaction among local-level components leads good scalability and robustness to the system. We propose a potential-based routing protocol as one type of self-organized routing protocols and show its scalability and robustness.

Since self-organized control is based on local interactions between system elements, it has high scalability and robustness; however, management of the whole system is very difficult. For example, desired behavior is not yet guaranteed in much larger networks based on pure self-organization. The controlled self-organization scheme has also been proposed from this perspective. Thus, we propose

a controlled potential-based routing protocol implementing a "controlled self-organization" scheme that also allows for external control. The scheme obtains close-to-optimal network behavior by the external control that controls a part of nodes in the network. We show that global traffic flow can be moderately controlled in a multi-sink large-scale sensor network. For example, traffic loads can be equalized among heterogeneously distributed sink nodes, and load balancing among the relay nodes based on remaining energy can bring an approximate four times extension of network lifetime.

Although there are many practical proposals on the scheme, no design approach for it has ever been investigated. At the last of the thesis, we propose and evaluate a design approach for realizing energy efficient, robust, scalable, and manageable networks based on controlled self-organization, paying attention to the control timescale.

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 Ph.D.

I am heartily grateful to the members of my thesis committee, Professor Koso Murakami and Professor Teruo Higashino of Graduate School of Information Science and Technology, Osaka University, and Professor Hirotaka Nakano of Cyber Media Center, Osaka University, for their multilateral reviews and perceptive comments.

Also, I would like to express my sincere appreciation for Professor Masashi Sugano of School of Knowledge and Information Systems, College of Sustainable System Sciences, Osaka Prefecture University. Without his continuous advices and supports, I would not have entered the Ph.D. program.

Furthermore, I must acknowledge Professor Naoki Wakamiya, Associate Professor Shin'ichi Arakawa, Associate Professor Go Hasegawa, Assistant Professor Yuichi Ohsita, Assistant Professor Yoshiaki Taniguchi, 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, Graduate School of Information Science and Technology, Osaka University. I am also thankful to my friends in the Ubiquitous Network

Laboratory, the Mobile Computing Laboratory, and the Information Sharing Platform Laboratory of Graduate School of Information Science and Technology, Osaka University. I learned a lot through a variety of discussions and interaction with them.

I cannot conclude my acknowledgement without expressing my thanks to my parents and family. Thank you for your giving me invaluable supports throughout my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Recent advances in wireless, micro-electromechanical, and battery technologies have made possible extremely small sensor nodes featuring wireless communication facilities, drawing considerable attention to wireless sensor networks [1]. In wireless sensor networks, many small, low-cost sensor nodes with wireless communication capabilities collect environmental data such as temperature, light, pressure, humidity, movement, and noise, and forward them toward data-sink nodes for human access. Wireless sensor networks provide sensing capabilities without a centralized infrastructure, making them useful for a wide range of applications and thus increasingly important. Original applications were for monitoring environmental information, but their applications have diversified. "Ubiquitous computing" [2] and "ambient intelligence" [3] are looked to as features of future networks, and sensor nodes and actuators provide these features. To achieve such networks, wireless sensors will play a greater role in collecting and processing information, and in interacting with environments and humans. However, critical technical problems remain. This thesis investigates energy efficiency, robustness, scalability, and manageability issues in wireless sensor networks.

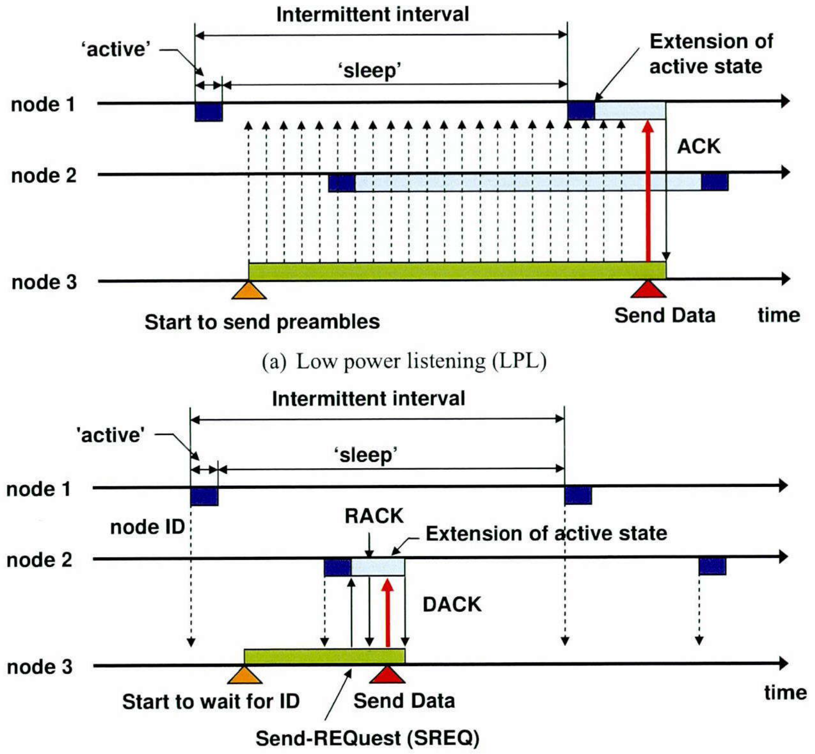
Energy Efficiency

A major problem in wireless sensor networks is the energy efficiency of sensor nodes with limited battery life. In this thesis, we place a primary emphasis on this aspect. Approaches to the improvement of energy efficiency include miniaturization of the sensor nodes, media access control (MAC) with sleep control, and multi-hop routing [4–12]. Here we discuss one of the MAC layer approaches, intermittent operation. Intermittent operation means wireless nodes sleep to save power and wake up periodically to transmit or receive packets. This can save energy because sleeping nodes consume considerably less energy than idling nodes [13]. In intermittent operation, nodes must control wake-up times (the 'intermittent interval') to communicate with each other.

Control methods for intermittent operation are classified into two types: synchronous [10–12] and asynchronous [6–9]. Synchronous methods use a beacon to synchronize between operations. Synchronization reduces energy consumption because the delay between waking up and data transmission states is shorter. The disadvantage is that regular beacon transmission consumes large amounts of energy, and can cause interference. Furthermore, all nodes must transmit at a fixed interval.

In asynchronous methods, nodes can communicate with other nodes at any time. There is therefore no traffic overhead for synchronization, reducing energy consumption and resulting in a highly scalable network. However, in these methods the sender node waits in an idle listening state until the receiver node awakens, which increases energy consumption in sender nodes. Long intermittent intervals can reduce node duty cycles and thus save energy, but this also increases the energy consumption of sender nodes. In terms of the overhead for synchronizing with other nodes, the latter is superior in terms of saving energy and enhancing scalability in systems with low packet generation rates. Here, we classify asynchronous control methods into two subtypes: sender-driven and receiver-driven. Classification depends on whether the sender or the receiver initiates communication. Message collisions must be controlled in both types, since nodes can initiate communication at any time.

The *low power listening* (LPL) protocol is a sender-driven asynchronous type of ad hoc network [8]. Figure 1.1(a) shows the basic intermittent operation of LPL (B-MAC [6]). Receiver



(a) Low power listening (LPL)

(b) Intermittent receiver-driven data transmission (IRDT)

Figure 1.1: Asynchronous MAC protocols with sleep control

nodes 1 and 2 intermittently check the channel state. If the channel is idle, they return to the 'sleep' state, and if it is busy, they enter the 'data wait' state. After receiving a data packet, nodes return an 'acknowledgement' message. For instance, when node 3 is ready to send data to node 1, it continuously sends preamble messages for a time period longer than the intermittent interval to activate the channel. After sending preamble messages, node 3 sends a data packet. However, there are a number of restrictions in this protocol. For example, when the intermittent interval is comparatively long, each sender node occupies the channel for a long time period while transmitting preamble messages, interfering with communication between neighboring nodes. Moreover, sender nodes transmit data packets to a specific node, reducing tolerance of node failures.

To overcome these drawbacks of LPL, Reference [14] proposes the *intermittent receiver-driven* *data transmission* (IRDT) protocol. IRDT is a receiver-driven MAC protocol, meaning communication between two nodes commences when the receiver node transmits its identifier (ID) to the sender node. IRDT addresses some of the restrictions of LPL. For example, it does not occupy the channel when the intermittent interval is long, and it can select as a receiver node a neighboring node from among multiple neighbors, thus constructing a mesh network at the MAC layer. In IRDT, receiver nodes periodically transmit small messages containing their ID (ID messages) as shown in Figure 1.1(b). Sender nodes wait for ID messages from receiver nodes, and after acquiring one return a send request (SREQ) message to establish a link. Note that IRDT have been developed as a protocol which has actually been implemented in meter products [14]. Furthermore, this protocol is proposed to IEEE 802.15 Task Group 4 as part of a standard protocol for smart meter systems [15]. We clarify the performance characteristics of IRDT in comparison with X-MAC and receiver-initiated MAC (RI-MAC) protocols through computer simulations.

Robustness

Sensor network robustness is a significant concern because sensor nodes and wireless links are subject to frequent failures due to harsh environmental conditions and energy depletion [16]. Robustness is the property of maintaining or recovering performance despite environmental variations, as illustrated in Figure 1.2. Environmental variation can entail changes of the route to the sink node, which can prevent end-to-end reachability and increase traffic load concentration. Without adequate robustness against environmental variation, severe perturbation of network conditions can reduce system performance to critical levels. Numerous approaches to optimizing sensor networks exist, but typically incur severe performance degradation after topological changes because they assume ideal situations. To solve these problems, mechanisms that monitor network conditions and leverage information on the network are effective. Paradis and Han [17] discuss various fault tolerant techniques for wireless sensor networks, but there have been few quantitative evaluations of robustness in wireless sensor networks.

We separate robustness into two properties, *robustness* and *resilience*, which respectively *maintain* and *recover* performance in the face of uncertain environmental variation. Unless otherwise

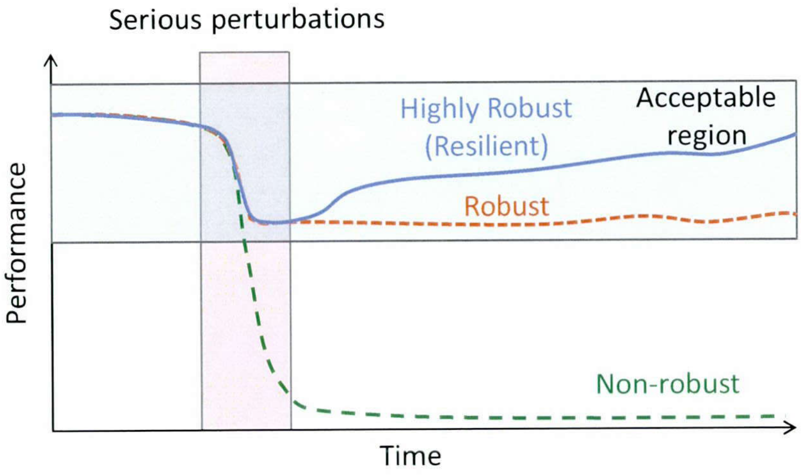


Figure 1.2: Robustness and resilience of system performance

stated, in discussions of robustness and resilience, "performance" refers to the packet delivery ratio, and we define these properties in a quantitatively evaluable form. "Environmental variation" includes abrupt increases of traffic load, random failure of sensor nodes, and sink node failure. We discuss how robustness and resilience are introduced and evaluate them with respect to MAC and routing layers in a sensor system.

Scalability

One challenge in creating wireless sensor networks is improvement of routing scalability [18]. When thousands of sensor nodes are present, the wireless channel is occupied by exchanges of routing information, which consumes considerable energy and bandwidth. Within such networks, it is impractical to give unique IDs to each node and exchange all routing information among them (unlike IP networks, which require arbitrary node access). Another scalability problem is that sink-node neighbors experience heavy loads, because many-to-one (or many-to-some) communication requires transmitting not only that node's data, but also forwarding data from neighbors.

Self-organization is expected to reduce the routing information exchanged throughout the network. In self-organization schemes, entirely local information is used for decision making by each node. Self-organization can provide good scalability, adaptability, and robustness [19], important properties for sensor networks. These properties arise through numerous interactions among local-level system components without external or centralized control processes. We therefore adopt potential-based routing [20–32] for self-organized routing. In potential-based routing schemes, nodes have a scalar value called "potential," and next hops are determined solely by the potential of a sensor node and its neighbors. A sensor node calculates its own potential from neighboring potentials, the number of hops to the sink node, or the remaining energy of itself or its neighbors. The smaller the hop count to the sink node, the lower the sensor node's potential value. Therefore, if a sensor node simply transmits data to a neighbor node with smaller potential than its own, the data will eventually reach the sink node.

To reduce load on neighboring sink nodes, multiple sink nodes are deployed across the network [33], and data obtained by the sink nodes are transmitted to a server. Users or applications can then access data from the server as necessary. Sensor nodes do not select a specific sink node as a destination; each node 'anycasts' its data. We apply potential-based routing to multi-sink sensor networks, which fortunately does not require special techniques. Once a potential field is adequately constructed, each node only has to forward data according to potentials and the data will eventually reach a sink node. Potential-based routing can thus be straightforward in multi-sink sensor networks.

Manageability

Practical realization of a self-organized network requires complicated emergent behavior to be manageable. However, decision-making based on local interactions in large systems results in emergent behavior, and precise management or control of such behavior is unrealistic. Thus, such a pure self-organization scheme has some problems because of its bottom-up design [34]:

- Guaranteeing optimal operation is difficult.
- Managing operations over the entire network is difficult.
- Convergence speed after an environmental change is slow.

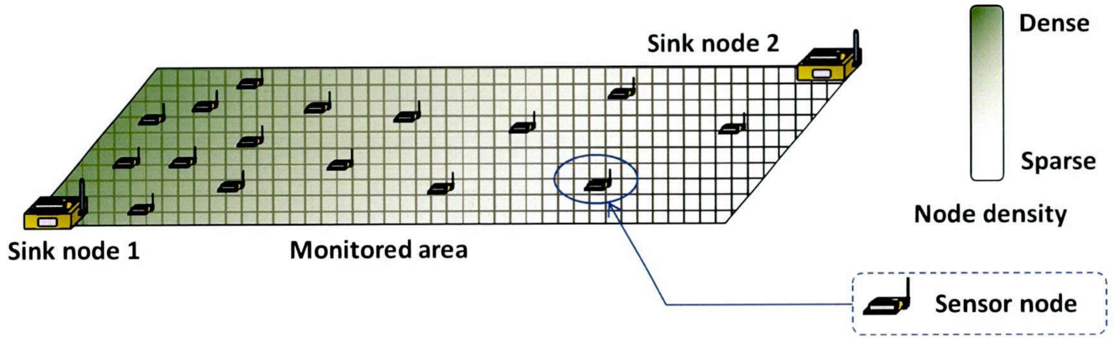


Figure 1.3: A serious imbalance of traffic. Heavy traffic around sink node 1 occurs because nodes are not aware of the irregular node-distribution density.

To solve these problems, Reference [35] proposes *controlled self-organization*. The authors of that paper suggest the use of an *observer/controller architecture*, where an *observer* and a *controller* are responsible for correcting system-level behavior. In controlled self-organization, an external observer and controller are responsible for 'external control,' guaranteeing that system behavior remains within constraints set by the system manager. The main task of the observer is to monitor system behavior by sampling information from a subset of system elements. The controller evaluates the system behavior reported by the observer and performs control actions that influence the system toward a given objective function. This observation/control loop is performed periodically to satisfy system goals. The *observer/controller* architecture is responsible for ensuring the desired behavior of the system, for guaranteeing high system performance, and for encouraging convergence of the system state, thus making the self-organized system *manageable* by controlling some of the self-organization components.

In the case of operation in self-organized routing, macroscale network problems cannot be considered because each node selects its next hop based on only local information. For example, excess concentrations of communication load induced by an irregular node-distribution density are difficult to alleviate (Figure 1.3). As a solution to these problems, we propose a controlled self-organization based routing protocol. We apply the controlled self-organization scheme to our potential-based routing, and thereby propose *controlled potential-based routing (CPBR)*.

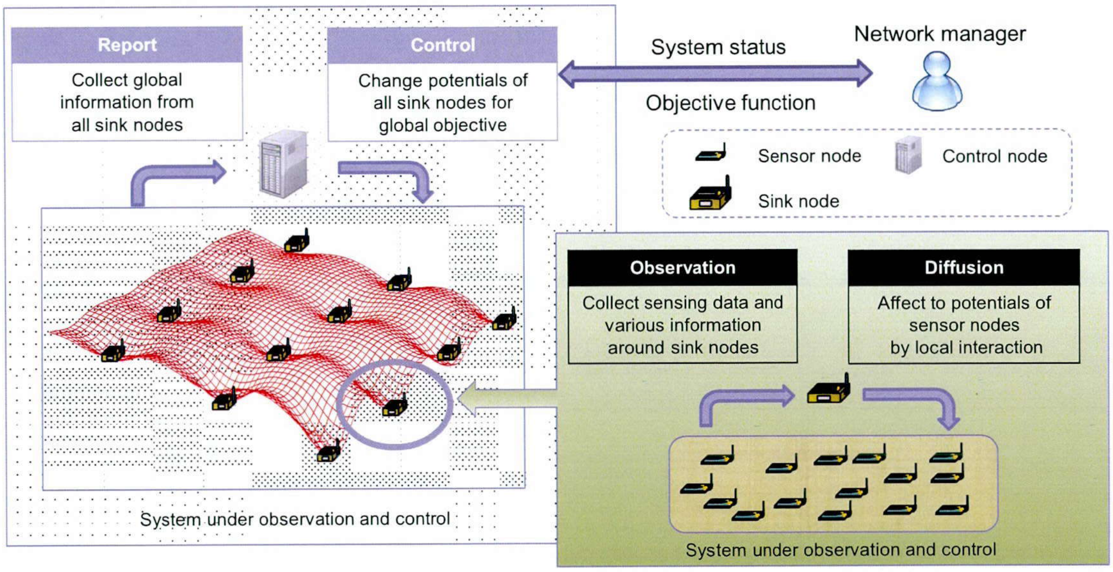

Report
Collect global information from all sink nodes
Control
Change potentials of all sink nodes for global objective
System status
Network manager
Objective function
Sensor node
Control node
Sink node
Observation
Collect sensing data and various information around sink nodes
Diffusion
Affect to potentials of sensor nodes by local interaction
System under observation and control
System under observation and control

Figure 1.4: Controlled potential-based routing (CPBR) architecture. Our architecture assumes that multiple sink nodes, which are powerful devices with sufficient energy supply, are placed at arbitrary positions within the network. Each can communicate with the control node (usually a high-end PC) through a high-speed wired or wireless connection (Ethernet, WiMAX, LTE, etc.). The control node changes sink-node potentials according to requests from the network manager, which is diffused over the entire network. Sink nodes report observation information to the control node.

Because potential-based routing performs self-organization, it retains the problems described above, such as communication-load concentrations caused by non-uniform sensor or sink node distributions. In CPBR, therefore, we introduce a control node (a *controller*). As shown in Figure 1.4, sink nodes monitor the network as *observers* and report observations to the control node. The control node adjusts sink-node potentials to construct desired potential fields according to the network manager's requests. We assume that multiple sink nodes are connected to the control node on a high-bandwidth wired or wireless network to enable periodic and instantaneous reports of various kinds of information, such as the number of received data packets and the remaining energy of neighbor nodes. The control node uses such information to adjust potentials so that a preferable potential field is constructed over the entire network, even as sensor nodes use local information to decide their own potential. The most significant difference between this and existing centralized systems is that controlled self-organization continues to function normally even if the controller is lost, albeit with a loss of near-optimality.

CPBR performance might prove inferior to routing with centralized control for optimizing performance. This is because self-organized methods make behavioral selections stochastically and locally, whereas centralized control can obtain theoretical upper limits of global performance. However, there exist scalability issues for centralized control, and recalculations for optimization are required whenever network conditions change. Wireless channel conditions fluctuate, and sensor nodes and links are prone to failure, so network conditions frequently change. Centralized control for optimization is therefore proper only under severely restricted conditions. Our CPBR cannot reach optimal performance, but it autonomously and adaptively approaches an optimal solution under various conditions. We show that the CPBR protocol autonomously and adaptively approaches an optimal solution under various conditions.

1.2 Outline of Thesis

An Energy-Efficient Receiver-Driven Data Transmission Protocol for Wireless Mesh Sensor Networks [36–40]

In Chapter 2, we address the importance of energy efficiency in wireless sensor networks, and evaluate and improve the IRDT protocol. We first examine the long-term operation of IRDT comparing energy consumption under conditions of low data incidence. We also improve IRDT by decreasing the incidence of control message collisions. Control message collisions are classified into two types: ID collisions, which occur between ID messages and other messages, and SREQ collisions, which occur between SREQ messages. Such collisions drastically reduce the performance of IRDT, and we discuss them in detail later in this chapter. We finally propose a simple and effective routing algorithm for mesh networks with IRDT, as well as novel improvement mechanisms for IRDT, and evaluate the impact of these improvements. Computer simulation shows that IRDT can reduce average energy consumption more than RI-MAC and X-MAC, especially under small loads. Simulation also demonstrates that IRDT with collision avoidance for control messages performs better than the original IRDT. This method ensures a packet collection ratio of more than 99% and an average energy consumption 38% lower than that of EA-ALPL and 90% lower than that of the original IRDT.

Robustness and Resilience in MAC and Routing Layer Protocols for Wireless Sensor Networks [41–44]

In Chapter 3, we discuss how robustness and resilience are introduced and improved in the MAC and routing layers of a sensor system. For the MAC layer, we focus on the difference between robustness of sender-initiated and receiver-initiated MAC protocols. We show that this difference is between the hard state and soft state [45,46], and that the latter has higher robustness. Moreover, we show that resilience in the MAC layer is obtained from the adaptive setting of appropriate duty cycles. For the routing layer, we address two points for robustness and resilience improvement: detour routing over a mesh network and management of routing tables. We demonstrate that soft-state management of routing tables has greater resilience than does hard-state management, and that robustness is enhanced by the existence of multiple candidates as next-hop nodes over a mesh sensor network. We show that receiver-initiated MAC protocols are more robust than sender-initiated ones, and computer simulation shows that a simple detour-routing algorithm has more than tripled robustness over the simple minimum-hop routing algorithm.

A Controlled and Self-Organized Routing Protocol for Large-Scale Wireless Sensor Networks [47–49]

Scalable and manageable properties are expected to be obtained by applying the controlled self-organization scheme to wireless self-organized sensor networks with multiple static sink nodes. In Chapter 4, we propose a scalable potential-based routing protocol based on the self-organization scheme, and apply the controlled self-organization scheme to our potential-based routing. An observer and a controller, which are assumed to connect with all sink nodes, are responsible for correcting system-level behavior. In the proposed routing, the external controller controls potentials of all sink nodes in the network. Moreover, we consider the properties of duty-cycle MAC protocols for potential-based routing protocols, and propose a simple but effective strategy for determining a next-hop node. This can elegantly perform local load balancing, and when used in combination with the controlled self-organization scheme, our proposed routing can attain global close-to-optimization of load balancing. Computer simulation shows that the proposed routing achieves traffic and energy-density balancing locally and globally. We also show that CPBR with potential control based on energy density can extend the time until the first node depletes its energy by 449%.

A Design Approach for Managed Self-organization Control Focused on Control Timescale for Future Wireless Sensor Networks [50–52]

Although controlled self-organization is important for realization of large-scale wireless sensor networks, the potential for unexpected situations due to simultaneous external and self-organized control remains poorly understood. Robustness to network topology change is also important for wireless sensor networks, where changes due to wireless channel conditions, node positions, and the number of nodes are commonplace. If communications protocols are not sufficiently flexible regarding environmental perturbations, various types of performance degradation may occur, such as data collection failures, data delivery delays, and increased energy consumption.

These perturbations and controls in each layer in the wireless sensor network architecture operate on widely different timescales. MAC layer protocols support one-hop communication, where data transmission takes a few milliseconds in most IEEE 802.15.4 sensor networks [53]. Energy efficient MAC protocols with sleep scheduling for prolonging network lifetime are often assumed in wireless sensor networks, which raises the lower limit of one-hop communication timescales due to the sleep cycles of tens of milliseconds to seconds [6, 7]. Routing layer protocols have to deal with topological changes to realize source-to-destination communications. In References [54, 55], static sensor nodes manage the network topology by using periodic HELLO messages every several tens of second. The timescale of the external control in controlled self-organization should be longer than that of the routing layer, because global behavior of a self-organized network arises as a result of that routing process. Thus, because these control timescales substantially differ, it is insufficient to discuss robustness within only one layer.

In Chapter 5, we propose a design approach for a scalable and robust network based on controlled self-organization, paying attention to the control timescale. We show that a design for robustness in only one layer cannot improve various types of perturbations that cause topological changes. Our study considers periodic environmental monitoring systems where sensor nodes deliver monitored data to multiple static sink nodes with CPBR. Then, we discuss how the timescale of control in the MAC, routing, and external control layers should be designed, and investigate these through computer simulation.

Finally, Chapter 6 concludes this thesis with directions for future work.

Chapter 2

An Energy-Efficient Receiver-Driven Data Transmission Protocol for Wireless Mesh Sensor Networks

2.1 MAC Layer Protocols with a Sleep Control Mechanism

In this section, we present some MAC protocols for intermittent asynchronous transmission and demonstrate the essential differences between sender-driven MAC and receiver-driven MAC.

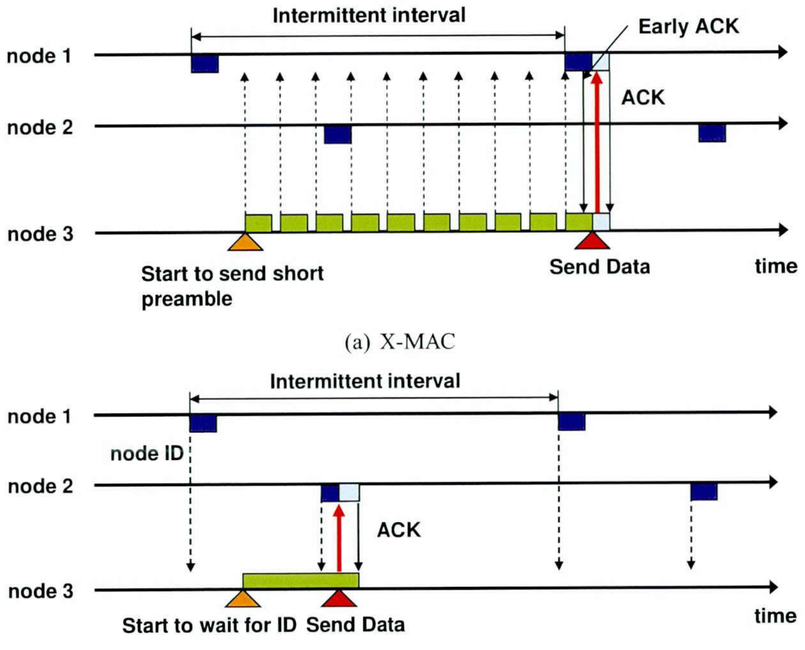
There are various approaches to media access control for intermittent asynchronous transmission. **B-MAC** [6] is the basis of LPL protocols as presented in Figure 1.1(a). In LPL, receiver nodes intermittently probe the state of the channel. As mentioned above, there are various problems associated with this LPL protocol; for instance, when the intermittent interval is comparatively long, each sender node occupies the channel by transmitting preamble messages for a period of time which is longer than the interval, thus interfering with any transmission from neighboring nodes. Moreover, the preamble messages transmitted from the sender consume the energy of unrelated receivers, which is known as "overhearing problem". Another problem is that each sender node has only one specific receiver.

Energy-aware adaptive low power listening (**EA-ALPL** [8]) is based on B-MAC. The procedure followed by receivers and senders in EA-ALPL is the same as the one shown in Figure 1.1(a), however, each node reconfigures its intermittent interval and adapts it to changes in traffic in order to attain higher energy efficiency. For high energy efficiency, the next hop selected by a sender node is the receiver which has the minimum hop count from the sink node. When there are multiple receiver candidates with minimum hop count, a sender node selects one of the most preferable nodes in accordance with the cost function of the intermittent interval and the sensing activity of neighboring nodes. The sensing activity is a Boolean variable, and it is determined by the sensing frequency of a node. In order to select a receiver, nodes regularly exchange information regarding the sensing activity and their own intermittent interval.

X-MAC [7] was designed to solve the overhearing problem of B-MAC. In order to prevent the preamble messages of the sender in B-MAC from occupying the channel, X-MAC continuously transmits short preamble messages to which the ID of the receiver is appended. The operation of X-MAC is shown in Figure 2.1(a). The receiver node replies with an early acknowledge (early ACK) message when the ID added to the short preamble corresponds to its own ID. The sender node transmits a data packet after receiving this early ACK and waits for the acknowledge message for the data. Receivers that detect unrelated short preambles can resume their state of sleep soon after the end of the reception. Thus, the overhearing problem generated by continuous transmission of preambles during intermittent intervals in B-MAC can be solved.

Although various receiver-driven asynchronous MAC protocols have also been proposed, most of them either assume that all nodes are active and can receive packets at any time, or that they use multi-channel access for transmitting packets [9,56,57]. In [56], receiver-driven media access control with a single channel, named "receiver initiated multiple access" (**RIMA**), is proposed. RIMA employs a collision avoidance handshake mechanism with CSMA/CA and obtains a reasonable throughput; however, this protocol does not use intermittent operation since it does not consider energy consumption.

In [9], two generic intermittent asynchronous MAC protocols are proposed, namely, Transmitter Initiated CyclEd Receiver (**TICER**) and Receiver Initiated CyclEd Receiver (**RICER**). The procedure of sending and receiving data in RICER is similar to that in IRDT, where receiver nodes periodically transmit ID messages. However, unlike the procedure in IRDT described in Section 2.2,



(a) X-MAC

(b) Receiver-Initiated MAC (RI-MAC)

Figure 2.1: Other MAC protocols with sleep control

a sender node in RICER transmits a data packet after obtaining an ID message. Furthermore, two channels are used for communication, and a sender uses only one receiver in RICER. In contrast, IRDT uses a single channel, which simplifies the implementation and ensures a highly reliable system. However, single-channel access causes control message collision.

Receiver-initiated MAC (**RI-MAC**) is also a receiver-driven MAC protocol, and thus it is similar to RICER [57]. In RI-MAC, a sender also transmits a data packet after receiving an ID message, however, RI-MAC uses a single channel for the transmission of packets (Figure 2.1(b)). In order to avoid message collisions, RI-MAC only uses collision detection and exponential backoff. Also, in terms of the routing algorithm, the authors of this protocol used minimum hop routing. IRDT uses an adaptive intermittent interval, whereas both RICER and RI-MAC use a fixed value for the intermittent interval. Such an adaptive interval can avoid message collisions and can attain higher performance. In this chapter, we propose a simple and effective routing algorithm for IRDT which is considered for mesh networks in an effective and efficient manner.

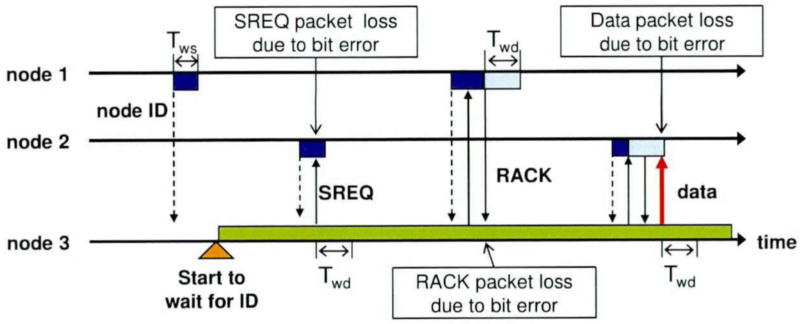

SREQ packet loss due to bit error
Data packet loss due to bit error
T_{ws}
T_{wd}
node 1
node ID
node 2
SREQ
RACK
data
node 3
time
Start to wait for ID
T_{wd}
RACK packet loss due to bit error
T_{wd}

Figure 2.2: T_{ws} and T_{wd} timers in IRDT

Here, an essential difference between IRDT and LPL is that nodes in intermittent operation mode transmit messages or listen to the channel, which can also be considered an essential difference between the sender-driven method and the receiver-driven method. In our previous research, we demonstrated the impact of this difference on the performance.

2.2 Intermittent Receiver-Driven Data Transmission

2.2.1 MAC Protocol

In IRDT, each receiver sends its own ID to inform other nodes that they are ready to receive a data packet. A sender node waits for a receiver ID, and when it acquires an ID from an appropriate receiver, it establishes a link with it by returning an SREQ message. After obtaining an acknowledge message for SREQ (RACK), the sender transmits a data packet and terminates the communication upon receipt of an acknowledge message for the data (DACK). Carrier sense multiple access with collision avoidance (CSMA/CA) is used for sending messages. However, especially when a node transmits an ID message or an SREQ message, it terminates the transmission of those messages if the channel condition is busy. If the channel is idle, it transmits an ID message or an SREQ message after a random backoff period. Otherwise, when it transmits a data packet, a RACK message, or a DACK message, a binary exponential backoff mechanism is utilized.

Here, all nodes contain two timers, which are set immediately before starting to wait for an SREQ message, a RACK message, a data packet, or a DACK message. T_{ws} is the time allocated for waiting for an SREQ message following the transmission of an ID message. Furthermore, T_{wd} is the time allocated for waiting for a data packet, a RACK message, and a DACK message. After the transmission of a RACK message, an SREQ message, or a data packet, respectively, as shown in Figure 2.2. If a time T_{ws} passes before receiving an SREQ message after the transmission of an ID message, the receiver node enters sleep mode, as shown in the figure. The receiver node also enters sleep mode if the period T_{wd} before receiving a data packet after transmitting a RACK message extends beyond a certain limit. On the side of sender nodes, if a RACK message and a DACK message are not received from the receiver after a lapse of T_{wd}, they begin to wait for reception of another appropriate ID message. Note that, for the CSMA/CA backoff algorithm, T_{ws} is shorter than T_{wd}.

The decision of the sender regarding whether to send an SREQ message is taken on the basis of its routing protocol. In this way, a sender node can select a receiver node flexibly, which can enhance the communication reliability and save considerable amounts of energy. Therefore, in the routing layer, the routing protocol should be designed to use multiple receiver nodes in a flexible and efficient manner. A specific example is shown in Figure 1.1(b), where receiver nodes 1 and 2 are in intermittent operation mode. Sender node 3 checks the ID received from node 2 and accepts node 2 as an appropriate receiver.

2.2.2 Routing Protocol

The routing protocol of IRDT is based on the distance vector routing protocol. All nodes have routing tables and a routing function for deciding on the transmission of an SREQ message.

A routing table contains hop counts from the node which has created the table to all nodes in the network. In order to create its own routing table, each node must exchange its table with its neighbors. In IRDT, all nodes periodically wake up and wait for ID messages for a short period of time, which, however, is longer than the intermittent interval. When a node receives an ID message within this period, it registers on its routing table that the hop count to the sender of the ID is one. We refer to this interval as ‘sampling interval’ (denoted by T_{si}).

The routing algorithm for IRDT must be based on multi-hop routing, and therefore each node

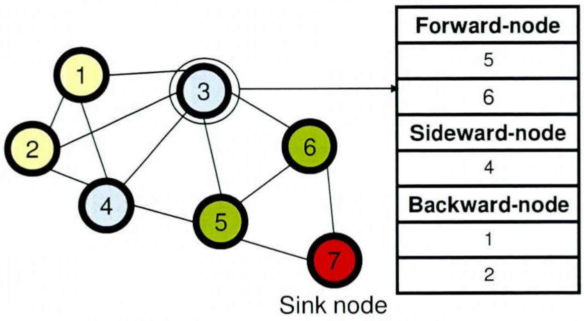


Figure 2.3: Classification of neighboring nodes at node 3

conducts the relay processing of the packet. Although minimum hop routing is preferable for the purpose of minimizing energy consumption, in some situations nodes cannot utilize the optimal routing due to poor radio wave conditions or failure of certain nodes. Therefore, for higher flexibility, the routing algorithm considers alternatives to the minimum hop route. Here, we define forward nodes, sideward nodes, and backward nodes. A node whose hop count from the sink node is H classifies its adjacent nodes as shown below.

Forward nodes: Adjacent nodes whose hop count from the sink node is $H - 1$.

Sideward nodes: Adjacent nodes whose hop count from the sink node is H.

Backward nodes: Adjacent nodes whose hop count from the sink node is $H + 1$.

Figure 2.3 shows an example of this classification of neighboring nodes.

The routing function is a logic function that utilizes a routing table. Sender nodes decide whether to return an SREQ message in accordance to this function, an example of which is shown in Figure 2.4. The function in Figure 2.4 assumes the minimum hop routing; however, detours are also used when the condition of sideward relay is satisfied.

Here, we define communication failure as a situation in which the sender cannot obtain a RACK and a DACK from the receiver. For minimum hop routing with detours, the sender node prefers forward nodes as receivers, and sideward nodes are selected if communication with all forward nodes fails. In order to prevent routing loops, all data packets have a time to live (TTL) field. The

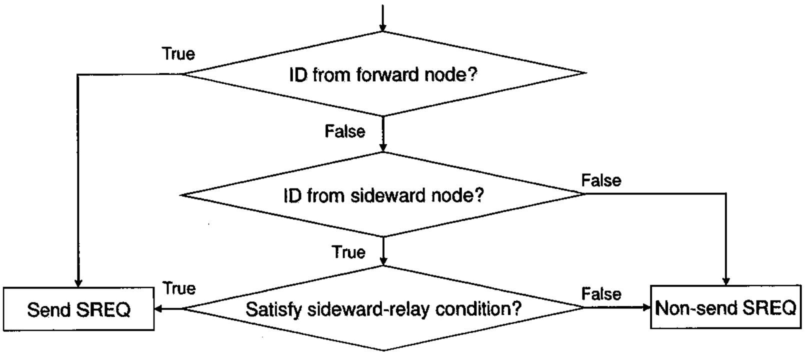


Figure 2.4: An example of a routing function

TTL is decremented by one only when a receiver node has received a data packet from a sender node. When the TTL becomes zero and the receiver is not the destination of the data packet, the data packet is discarded. No sender node selects a sideward node or a backward node if this results in loss of data packets due to the TTL mechanism.

2.3 Control Packet Collision

In this section, we discuss the control message collision problem in IRDT together with some novel approaches to resolving it. One problem related to IRDT is collisions between ID messages and other messages, as well as collisions between SREQ messages, which we refer to as 'ID collisions' and 'SREQ collisions', respectively. All nodes send ID messages periodically, and therefore ID messages can collide with other messages. Regarding SREQ collisions, the sender node returns an SREQ message when an ID message from a forward node arrives, as described in Section 2.2.2. Thus, if more than one sender receives an ID from a forward node, the sender nodes return SREQ messages simultaneously, the messages collide with each other. In this case, the sender nodes remain awake in wait for another ID, and as a result their energy consumption increases. Furthermore, SREQ collisions are in danger of recurring at nodes that are the only forward nodes for their backward nodes. For example, this 'recurring SREQ collision' often occurs at the sink node, which is the only forward node for its neighbor nodes. After an SREQ collision occurs at the sink node, more

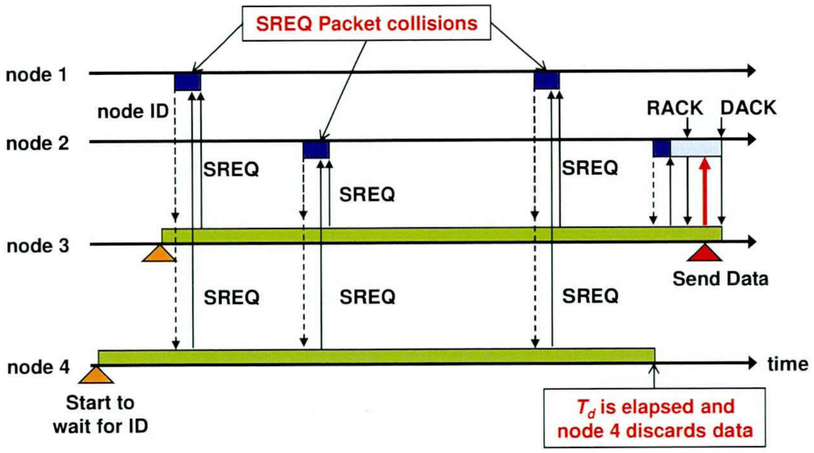


Figure 2.5: Recurring SREQ collisions; A major cause of excessive energy consumption in sender nodes

than one neighbor node still contains data packets. This causes another SREQ collision following the ID transmission by the sink node. Due to the scheduled timer for discarding data (set to T_d) built into all nodes, recurring SREQ collisions eventually cease. Since the sender continues waiting for an ID message until the sender receives a DACK message from a receiver, recurring SREQ collisions lead to large energy consumption, as shown in Figure 2.5. For the above reasons, a reduction of the respective rates of ID and SREQ collisions (collectively denoted as "control message collisions") is meaningful in terms of energy efficiency.

Next, we describe the influence of the intermittent interval on the probability of message collisions, as well as the procedure for determining a proper intermittent interval which decreases this probability. Changing the intermittent interval affects the following two aspects:

1. Probability of SREQ collisions

   This is the probability with which multiple nodes return SREQ messages simultaneously immediately after a receiver node sends an ID message. Since SREQ collisions are caused by data packet congestion, a longer intermittent interval increases this probability. If SREQ collisions occur, the energy consumption of the sender nodes increases due to retransmissions. Furthermore, such SREQ collisions can occur repeatedly.

2. Probability of ID collisions

   This probability corresponds to the likelihood that ID messages sent periodically by all neighboring nodes collide with SREQ or data packets. It is clear that a shorter intermittent interval increases this probability. As in the case of SREQ collisions, retransmissions increase energy consumption.

We propose three methods for resolving the control message collision problem, namely, reactive and proactive control of the intermittent interval and data aggregation. A protocol using the reactive method starts avoiding SREQ collisions soon after the first SREQ collision occurs. The advantage of this method is adaptability to changes in the network topology and the packet generation rate. In comparison, in the proactive method, the optimal intermittent interval which minimizes the sum of the respective probabilities for SREQ collisions and ID collisions is obtained analytically, where each node knows its own traffic load. We refer to this intermittent interval as the "proper interval" (denoted as T*). Finally, data aggregation can be used to decrease the number of data packet transmissions for each node, which can decrease the probability of SREQ collisions.

2.3.1 Collision Avoidance with Reactive Interval Setting

SREQ collisions are caused by two factors, one of which is the disagreement between the transmission capacity and the load of a node. The maximum number of packets that a node can receive per unit time corresponds to the number of IDs the node sends per unit time. Therefore, as the intermittent interval of a node is shortened, the amount of data that a node can receive increases. When the load exceeds the processing performance of the node, multiple SREQ messages are sent, and collision occurs. Accordingly, in the reactive method, each node sets its ID transmission interval dynamically. Nodes determine that their loads are high when collisions are detected while they are waiting for an SREQ message. In this case, they set their own intermittent intervals to T_{min}. If an SREQ collision is not detected, the nodes gradually increase their intermittent interval to T_{max} at increments of T_i after every transmission of an ID in order to reduce the duty cycle (Figure 2.6). Regarding T_{max} and T_{min}, although a longer T_{max} decreases the duty cycle of the node, it affects its neighbors by increasing the interval of waiting for an ID. In contrast, while a shorter T_{min} improves

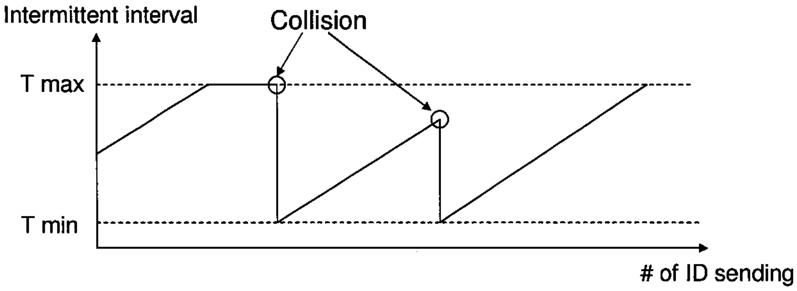


Figure 2.6: Dynamic control of the intermittent interval

the transfer performance, it interferes with communication between other nodes.

The other factor is the priority of forward nodes as receivers. As described in Section 2.2.2, when a sender node receives an ID message from its forward node, it transmits an SREQ message. Therefore, when more than one hidden node is ready to send data to the same receiver, whenever the receiver transmits an ID, an SREQ collision occurs. In addition, even if there are no hidden nodes, SREQ messages will collide if they are transmitted simultaneously. At nodes which are the only forward nodes for a large number of sender nodes, such as the sink node, SREQ collisions occur repeatedly, as mentioned before in the section regarding recurring SREQ collisions. In order to solve this problem, it is necessary for sender nodes to ignore the ID message of their forward nodes in a random fashion. Therefore, if a node fails to transmit a packet to all its forward nodes, which is a situation described as 'communication failure' in Section 2.2.2, it ignores IDs from the forward nodes with a fixed probability denoted by P_f.

As P_f becomes larger, sender nodes tend to transmit data packets to sideward nodes. Thus, a large P_f leads to an increase in both the number of data relays and the period of waiting for ID messages from sender nodes. We utilize the concept of disregarding ID messages with a certain probability for selecting the appropriate P_f. Although this additional process cannot prevent initial collisions, once a collision occurs, each sender node autonomously avoids further collisions.

2.3.2 Collision Avoidance with Proactive Interval Setting

Analytical Derivation of the Probability of Control Message Collision

In analyzing the probability of control message collision, we introduce the following assumptions.

- All nodes possess complete information about the network topology and contain a static routing table based on this information. Here, we use the topology shown in Figure 2.7, where node R is a sink node. Thus, the forward node of node A is node R, and its sideward nodes are node B and node C.

- Each sensor node generates a data packet in accordance to a Poisson process with intensity λ, and subsequently sends the data to the sink node. In addition, when nodes forward data, they always select forward nodes, and any forward node is equally likely to be chosen as the receiver.

- When message collisions occur, the receiver of the messages always discards all messages involved in the collision.

- Each node sends ID messages at a regular intermittent interval denoted as T. Moreover, all nodes perform the "clear channel assessment" (CCA) procedure when sending any type of message. Neither ID messages nor SREQ messages are transmitted if the CCA has indicated that the wireless channel is busy. If the wireless channel is idle, nodes transmit an ID or an SREQ message after a random backoff period of time. After it is ensured that receivers can obtain SREQ messages correctly, collisions between data, RACK and DACK messages and other messages occur less frequently. However, if a collision occurs, a receiver must wait for the following ID message, which increases the total amount of time spent by the affected sender node in waiting for an ID message. Therefore, data packets, RACK messages and DACK messages are transmitted by using binary exponential backoff in order to prevent collisions with other messages (especially ID messages).

From the above assumptions, we can calculate $G(R)$, which is the approximate average number of data packets received by node R in one second. $G(R)$ depends on the number of backward nodes

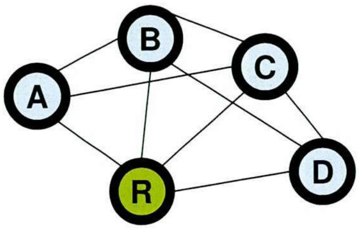


Figure 2.7: A simple network example for collision analysis

for node R and its traffic load. Here, we define $N_b(R)$ as the set of backward nodes of node R and $|N_f(n)|$ as the number of forward nodes of node n. The probability with which a node (denoted as n) selects node R as its receiver is $\frac{1}{|N_f(n)|}$, and therefore $G(R)$ is expressed as follows:

$$G(R) = \sum_{n \in N_b(R)} \frac{1}{|N_f(n)|} \{G(n) + \lambda\}. \tag{2.1}$$

SREQ collisions occur when two or more neighboring nodes send SREQ messages simultaneously. We assume that all nodes use the CSMA/CA mechanism, which can reduce the number of SREQ collisions.

However, SREQ collisions can still occur, unless there are no hidden nodes, since SREQ messages can be returned at once. In the CSMA/CA mechanism with exponential backoff, the number of time slots chosen at random by each node is 2^{BE}, where BE is a moderate integer value. If the wireless channel is idle, the sender node transmits an SREQ message (or an ID message) after a CCA and a random backoff period, as described in Section 2.2.1. In this regard, a time slot with a range of 2^{BE} is utilized for the random backoff period. Here, we assume that node R receives the same number of data packets from each of its backward nodes. Therefore, the probability with which a node returns an SREQ message upon receiving an appropriate ID can be expressed as $1 - e^{-G_b(R)T}$, where $G_b(R)$ is $\frac{G(R)}{|N_b(R)|}$. Furthermore, the probability with which the node does not return an SREQ message can also be expressed as $e^{-G_b(R)T}$. P_{SREQ}, which is the probability with which SREQ collisions occur, is also the probability with which at least two neighboring nodes of node R receive a data packet. However, the CSMA/CA mechanism cannot avoid SREQ collisions when node R sends an ID message. Thus, P_{SREQ} can be calculated as follows:

$$P_{SREQ} = 1 - \sum_{k=0}^{|N_b(R)|} C(R,k)e^{-(|N_b(R)|-k)G_b(R)T}(1 - e^{-G_b(R)T})^k, \tag{2.2}$$

where $C(R,k)$ indicates the number of combinations of k different nodes out of $N_b(R)$, which addresses the hidden node problem under CSMA/CA. Here, we consider only the case where k is less than three because the term $e^{-(|N_b(R)|-k)G_b(R)T}(1 - e^{-G_b(R)T})^k$ is exceedingly small and can be ignored for large k. $C(R,k)$ is defined as follows:

$$C(R,k) = \begin{cases} 1 & (k=0) \\ |N_b(R)| & (k=1) \\ \frac{2^{BE}-1}{2^{BE}}h(R) & (k=2), \end{cases} \tag{2.3}$$

where $h(R)$ is the number of couples of nodes out of $N_b(R)$ in relation to the number of hidden nodes.

Next, we target collisions of ID messages at node R. A collision of ID messages occurs when ID messages are sent by the neighbors of node R while node R is receiving an SREQ message or a data packet. Note that it is not necessary to consider the backoff time slot of CSMA/CA as discussed in P_{SREQ} since ID messages are rarely transmitted simultaneously by multiple nodes. Here, we define $H(R)$ as the average number of hidden nodes for node R for the time when node R is receiving SREQ message or data packet. $H(R)$ is represented as follows:

$$H(R) = \frac{1}{|N_a(R)|} \sum_{n \in N_a(R)} h(R,n), \tag{2.4}$$

where $N_a(R)$ is the set of adjacent nodes for node R, $|N_a(R)|$ is the number of elements of $N_a(R)$ and $h(R,n)$ is the number of hidden nodes for node n included in $N_a(R)$.

The average interval for receiving ID messages while node R is receiving SREQ message or data packet can be computed as $\frac{T}{H(R)}$ because $H(R)$ nodes can send ID messages even while node R is receiving other messages. Here, we define T_r as the reception time for SREQ message and data packet, in which case the probability of ID collisions, denoted as P_{ID}, is expressed as follows:

$$P_{ID} = \frac{T_r H(R)}{T}. \tag{2.5}$$

Procedure for Determining the Proper Transmission Interval

In order to determine the proper transmission interval, we modify Equation (2.2). Equation (2.2) shows the probability with which an SREQ collision occurs when an ID message is sent by node R, and Equation (2.5) shows the probability with which an ID collision occurs when node R receives an SREQ or a data packet. Therefore, we introduce P'_{SREQ} (the product of P_{SREQ} and $(G(R)T)^{-1}$), which corresponds to the probability with which an SREQ collision occurs when receiving an SREQ or a data packet (Equation (2.6)).

$$P'_{SREQ} = \frac{1 - \sum_{k=0}^{2} C(R,k) e^{-(2-k)G_b(R)T}(1 - e^{-G_b(R)T})^k}{G(R)T}. \tag{2.6}$$

Then, we can obtain T^* by minimizing P_{CTRL}, which is the probability of control message collisions, as follows:

$$P_{CTRL} = P'_{SREQ} + P_{ID}. \tag{2.7}$$

Unfortunately, an explicit expression of T^* which minimizes Equation (2.7) cannot be given; instead, we can compute the approximate value of T^* by calculating the minimum value of the sum and subsequently computing T^* every 10 ms in the semi-open interval (0.0 s, 2.0 s].

Figure 2.8 shows the results of the analysis and simulation of control message collisions for the network topology shown in Figure 2.7, where $\lambda = 0.024$, $BE = 3$ and the error bar corresponds to the 95% confidence interval. From the results shown in Figure 2.8, it can be concluded that the analysis and the simulation of both P_{ID} and P_{SREQ} correspond rather well, which indicates that our analysis is correct. However, for P_{SREQ}, as the intermittent interval becomes longer, the simulation results indicated superior performance than the analytical results due to the assumption

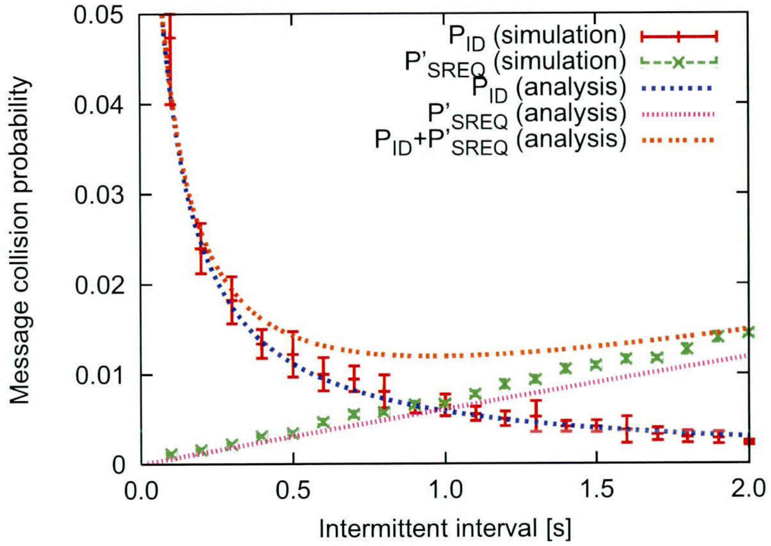


Figure 2.8: Probability of control message collisions

that CSMA/CA can always prevent message collisions, except in the presence of hidden nodes. In fact, CSMA/CA cannot completely avoid message collisions even when two nodes are hidden with respect to each other. Also, SREQ collisions tend to occur more often as more backward nodes contain data packets. Therefore, when the packet generation rate is high, SREQ collisions occur more frequently. In an actual multi-hop network, a node sends data packets not only to forward nodes, but also to sideward nodes and backward nodes since P_{SREQ} in an actual network is difficult to estimate. Moreover, the actual average number of data packets received in one second increases due to retransmissions.

2.3.3 Collision Avoidance with Data Aggregation

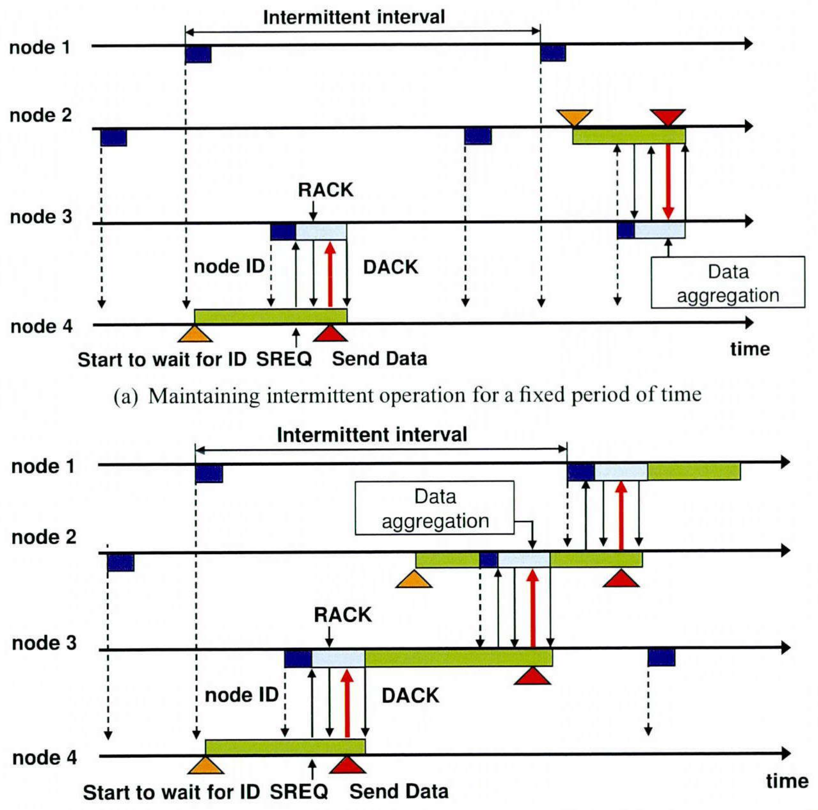
Data aggregation can reduce the number of data packet transmissions for each node. We assume that when a node aggregates m data packets, the size of the data packet increases m times, and the number m is appended to the ID messages in order to inform the receiver nodes about the identity of the sender node. Therefore, a larger m effectively decreases $G(R)$ in Equation (2.6), and P_{SREQ} also decreases. Unfortunately, it increases T_r in Equation (2.5) as well as P_{ID}. We present this trade-off in the following section.

Here, we demonstrate the strong effect of data aggregation with sideward nodes. Although data transmissions toward sideward nodes increases both the number of data relays and the consumption of energy, when using data aggregation, relay with sideward nodes is more effective since data aggregation with both sideward and forward nodes greatly decreases $G(R)$. SREQ collisions can occur between two or more nodes even if they are not hidden. This occurs when the random numbers for two nodes selected through the binary exponential backoff mechanism coincide . For example, if node 3 and node 4 in Figure 2.5 are not hidden nodes, an SREQ collision might occur. However, if data aggregation at these nodes is performed well, only one node contains the aggregated data packet, and no SREQ collision occurs. Moreover, data aggregation can resolve recurring SREQ collisions which occur when there is only one forward node, such as a sink node. In our previous research, we demonstrated that these repeated SREQ collisions cause an increase in energy consumption. If IRDT does not use data aggregation, repeated SREQ collisions continue to occur until the sending time expires. Specifically, when data aggregation is possible, the priority of the forward nodes is extended to sideward nodes which contain data packets. Whether sideward nodes receive data packets can be determined by adding this information to the ID messages.

We limit the size of the aggregated data packets for the reasons noted above, namely, a large value of m increases both P_{ID} and the channel occupation time. We insert the number m into the ID messages in order to inform the receiver nodes about it, which can also be used to provide information about whether sideward nodes receive data packets. The use of this information prevents the data packet size from exceeding m times the original data size as a result of aggregation.

Here, two methods can be used to add the functionality of data aggregation to IRDT:

1. Maintaining intermittent operation for a fixed period of time: Sender nodes immediately begin to wait for an ID message in IRDT when they receive or generate a data packet. At that time, data aggregation can be achieved by continuing their intermittent transmission of ID messages in order to receive data packets until the end of the fixed period of time without waiting for an ID message, as shown in Figure 2.9(a). The node begins to wait for an ID message when the size of the aggregated data packet reaches a certain predetermined size or



(a) Maintaining intermittent operation for a fixed period of time

(b) Maintaining intermittent transmission of ID messages while waiting for an appropriate ID

Figure 2.9: Data aggregation procedures in IRDT

a certain period of time passes.

2. Maintaining intermittent transmission of ID messages while waiting for an appropriate ID: In the current implementation of IRDT, the node which contains a data packet does not send an ID message, although it is waiting for ID messages from other nodes. In order for sender nodes to receive data packets while waiting for an ID message, they alternate the processes of transmitting ID messages and waiting for an appropriate ID message, as shown in Figure 2.9(b). When they receive an SREQ message, they perform data aggregation, and when they receive an ID message from an appropriate receiver, they cease the aggregation and transmit an SREQ message.

The first method decreases the data transmission frequency through aggressive data aggregation, while the second method aggregates data without increasing the delay time. In this chapter, we focus on the first method in order to achieve higher energy efficiency.

2.4 Simulation Results

In this section, we evaluate and compare the performance of IRDT, RI-MAC and X-MAC by using computer simulation. Also, we clarify the impact of collision avoidance for control messages. We devised a large-scale sensor network system composed of a large number of nodes as an application of the proposed method to our further studies. However, the ns-2 simulator, which is the most general simulation tool, does not scale well for such sensor networks, as discussed in [58]. Therefore, we prepared an event-driven simulation program written in Visual C++ for this experiment. Evaluation by using a general simulator that scales well for sensor networks is under consideration. Here, we use the network model shown in Figure 2.10, in which one sink node and 49 sensor nodes are deployed over 400x400 m^2. In this figure, the sink node is represented as a square, and other shapes denote sensor nodes. The communication range of each node is 100 m, and the sensor nodes shown in the figure with the same shape and color have the same number of hops from the sink node. When modeling the network, we used the following assumptions:

- Static network topology
- A disk model is used in order to abstract away from any fluctuations in wireless communication
- The capture effect is not considered

In order to examine the impact of collision avoidance for control messages, we assume that the network topology is static. Regarding the model of communication between nodes, we employ the disk model, where the strength of the radio signals does not deteriorate, and unless message collisions occur, a transmitted message is assumed to be received for certain by the nodes within the communication range. In addition, our evaluation is performed on with conservative settings for

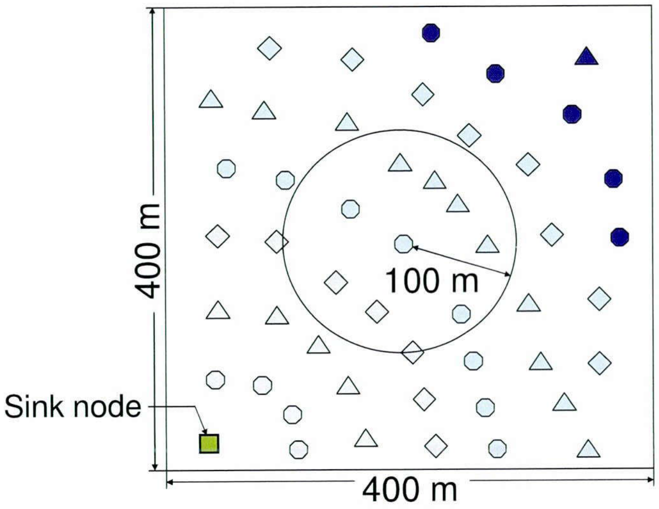


Figure 2.10: 50-node network for evaluation of IRDT

the message collision model in which both messages are always discarded if a message collision occurs while a message is being received.

Note that when another wireless communication model is utilized, the value of T^* is varied with time, and therefore nodes should frequently exchange information about the network topology for the purpose of calculating $G(R)$. Also, regarding the capture effect, even though the value of P_{SREQ} appears to decrease slightly, SREQ collisions are of intrinsic importance in IRDT.

In our simulations, sensor nodes other than the sink node in the network generate data packets according to a Poisson process. Each sensor node transmits data to the sink node through a multi-hop relay, where the routing algorithm for IRDT in the simulation is described in Section 2.2.2. Here, data is collected after completion of the exchange of routing tables. Each node conducts CSMA/CA in order to avoid collisions with other messages. Before a node transmits an ID message or an SREQ message, it performs a clear channel assessment (CCA). If the channel is busy, it does not transmit a message. In the case of other types of message transmission, a node performs up to five attempts for binary exponential backoff of CSMA/CA. The initial size of the contention window is set to W_{min} and incremented up to W_{max}. All nodes use a data discard timer for preventing repeated SREQ collisions from occurring, where the timer is set to T_d. The parameters are set as

Table 2.1: Parameter settings for basic performance evaluation

Parameter	Value
Simulation time	6 h
Transmission speed	100 kbps
Communication range	100 m
T_d	5 s
T_{si}	300 s
TTL	$H + 3$
T_{ws}	2 ms
T_{wd}	10 ms
Contention window size (W_{min})	3
Contention window size (W_{max})	5
Current consumption (TX)	20 mA
Current consumption (RX)	25 mA
Current consumption (Sleep)	0 mA
Message size (ID, SREQ)	24 bytes
Message size (DATA)	128 bytes
Message size (RACK, DACK)	22 bytes

shown in Table 2.1. In particular, the TTL is set to $H + 3$ (H is the number of hops from the sink node) since extra relays increase the energy consumption.

We investigated the message collection ratio, that is, the number of packets received at the sink node divided by the total number of generated packets. We also investigated the energy consumption of the node with the heaviest load, which is determined by the maximum energy consumption, as well as the average energy consumption for all nodes when the packet generation rate (the number of data packets generated at each node per 1.0 s) is changed. Here, we use the term 'performance' to indicate the packet collection ratio, the maximum energy consumption, and the average energy consumption.

2.4.1 Basic Performance

The performance of all methods is examined for the topology shown in Figure 2.10. In order to investigate the basic performance, the intermittent interval is set to a constant value which is the same for all nodes. Although shorter intermittent intervals are important for improving the performance in IRDT, extremely short intervals cause frequent transmission of IDs, which appears to interfere with other communication. Therefore, we examine the basic performance in the case where the intermittent interval is set to 0.1 and 1.0 s. We clarify the performance characteristics of IRDT by comparing them with those of RI-MAC and X-MAC. In IRDT, each node transmits an ID message and waits for an SREQ message. The time for ID transmission is 1.92 ms, T_{ws} is set to 2 ms, and in X-MAC each node periodically waits for 4 ms for a short preamble. In addition, X-MAC and RI-MAC use minimum hop routing, where sender nodes select one receiver node out of the neighboring nodes with minimum hop count from the sink node.

Packet Collection Ratio

The collection ratio is shown in Figure 2.11. In case the intermittent interval is set to 0.1 s, highly frequent ID transmissions interfere with the communication of other nodes in IRDT and RI-MAC. However, the collection ratio is comparatively high (always over 98%) since T_d is much longer than 0.1 s, which increases the chance for retransmission. In contrast, at an intermittent interval of 1.0 s, IRDT can attain a collection ratio of almost 100% when the packet generation rate is low, although the collection ratio decreases to less than 45% at relatively high packet generation rates. This result can be explained with SREQ collisions and the repeated SREQ collisions mentioned in Section 2.3.2. As the intermittent interval becomes longer, these collisions increase further, and the collection ratio for high packet generation rates at 1.0 s results in lower values of the collection ratio. Also, in RI-MAC, data packets collide with each other, and the packet collection ratio decreases as the packet generation rate increases. In this case, owing to the detour routing, IRDT can attain a higher packet collection ratio in comparison to RI-MAC.

In X-MAC, the collection ratio is lower than that in IRDT since the sender nodes transmit preamble packets without considering their receivers. If a sender node cannot obtain an early ACK, it transmits preambles throughout T_d, which interferes with other communication. Thus, it can be said that X-MAC is clearly disadvantageous for retransmission in the MAC layer. However, unlike IRDT, in X-MAC a short intermittent interval does not interfere with other communication since each node periodically inspects the condition of the channel. Therefore, X-MAC can reduce the

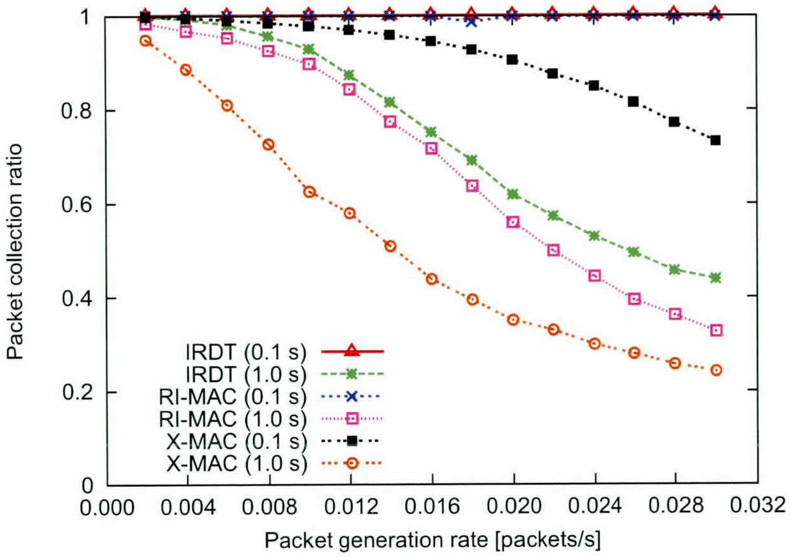


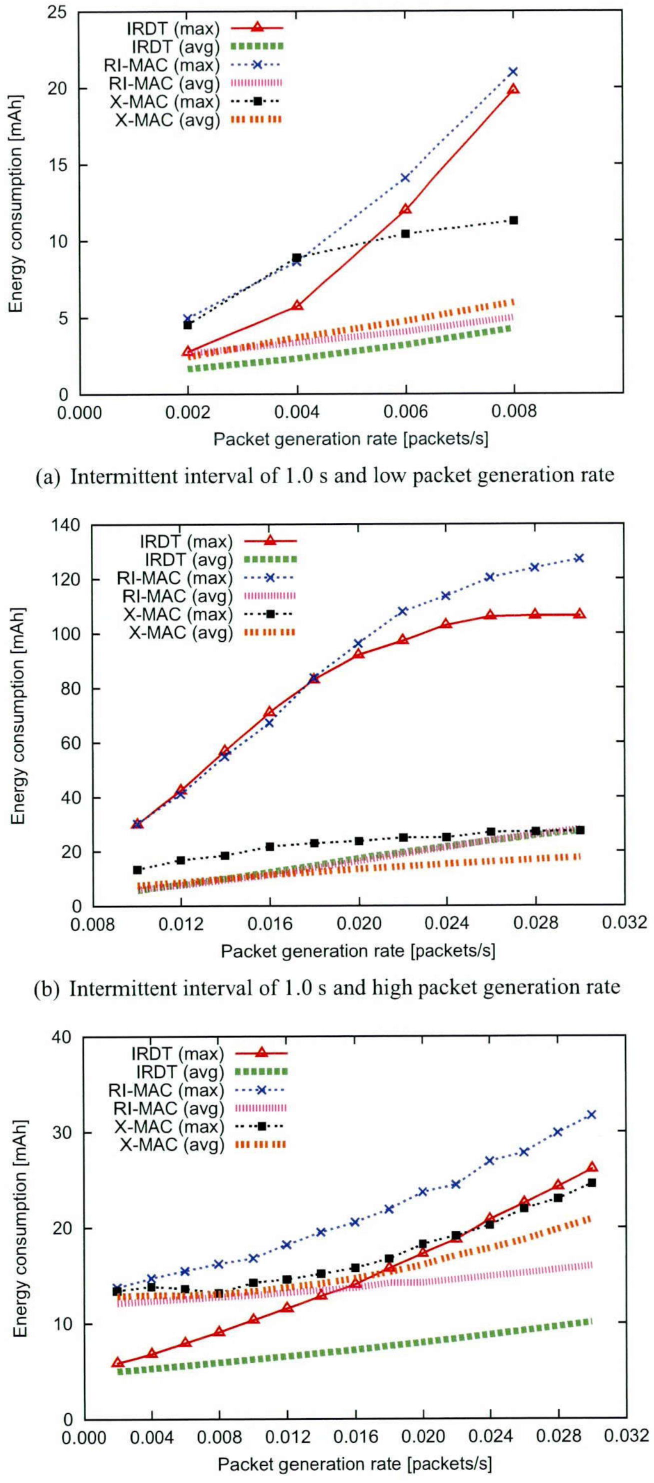
Figure 2.11: Basic performance; packet collection ratio

length of the intermittent intervals, and as a result it can achieve a higher collection ratio.

Energy Consumption

We examine the average energy consumption and the maximum energy consumption for all nodes (Figure 2.12).

In a comparison between IRDT and X-MAC at a low packet generation rate, when the intermittent interval is 1.0 s, the average energy consumption for IRDT is 33% lower than that of X-MAC since in IRDT there can be more than one receiver, as shown in Figure 2.12(a). In intermittent operations, more energy is consumed when sender nodes wait for the receiver, and using multiple receivers can reduce this waiting time. In comparing IRDT and RI-MAC, it is found that the energy efficiency of IRDT is higher due to the use of SREQ messages. Since the data packet size is larger than the SREQ message size, when a receiver obtains a data packet and detects bit errors in RI-MAC after an ID transmission, the wasted energy is greater than that of SREQ collisions in IRDT. Also, in both IRDT and RI-MAC, the neighboring nodes of the sink node consume large amounts of energy since SREQ (or data) collisions occur more frequently at the sink node, which prolongs



(a) Intermittent interval of 1.0 s and low packet generation rate

(b) Intermittent interval of 1.0 s and high packet generation rate

(c) Intermittent interval of 0.1 s

Figure 2.12: Basic performance; energy consumption

the idle time for listening for senders (Figure 2.12(b)). Thus, the energy consumption of the neighboring nodes of the sink node (IRDT (max)) grows rapidly in accordance with the increase of the packet generation rate when the intermittent interval is 1.0 s. Similarly, the energy consumption increases at nodes whose receivers experience frequent collisions of SREQ messages. In X-MAC, procedures for collision avoidance are not used, with the exception of CSMA/CA. Therefore, a short intermittent interval is necessary in order to achieve a higher collection ratio, although this prolongs the total idle listening time.

When the intermittent interval is 0.1 s, the maximum energy consumption in the case of IRDT does not grow considerably due to the smaller number of SREQ collisions (Figure 2.12(c)), and this is the same in the case of RI-MAC. The consumption of energy for both RI-MAC and X-MAC is higher than for IRDT. In RI-MAC, nodes wait for a data packet after sending an ID message during T_{wd}. This entails higher energy consumption than for IRDT, which uses T_{ws}. In addition, energy is consumed by overhearing a short preamble or a data packet in X-MAC. Also, in X-MAC, each node attempts to transmit a short preamble message without considering the state of the receivers, which results in data retransmissions and consequently increases the network-wide energy consumption.

2.4.2 Effects on Collision Avoidance for Control Messages

Reactive and Proactive Setting of the Intermittent Interval

At this stage, we introduce a method for SREQ collision avoidance (as described in Section 2.3.1 and 2.3.2) to IRDT and show the strong effects of this method. For the evaluation of this method for SREQ collision avoidance, we assume that exchanges of routing tables are not considered and all nodes have correct routing tables. The reactive setting of the intermittent interval is shown in Figure 2.6, and its parameters are shown in Table 2.2. T_{max} is set by assuming continuous operation of about several years, and a short T_{min} is set in order to reduce SREQ collisions. After the interval becomes T_{min}, it is increased in steps of T_i at every transmission of an ID message. On the other hand, the proactive method uses T^*. Here, as previously discussed in Section 2.3.2, each node can obtain the approximate value of T^* by calculating the minimum value of Equation (2.7).

By avoiding control message collisions, a higher collection ratio and lower energy consumption

Table 2.2: Parameter settings for reactive setting of the intermittent interval

Parameter	Value
T_{max}	1.5 s
T_{min}	0.1 s
T_i	10 ms
P_f	50%

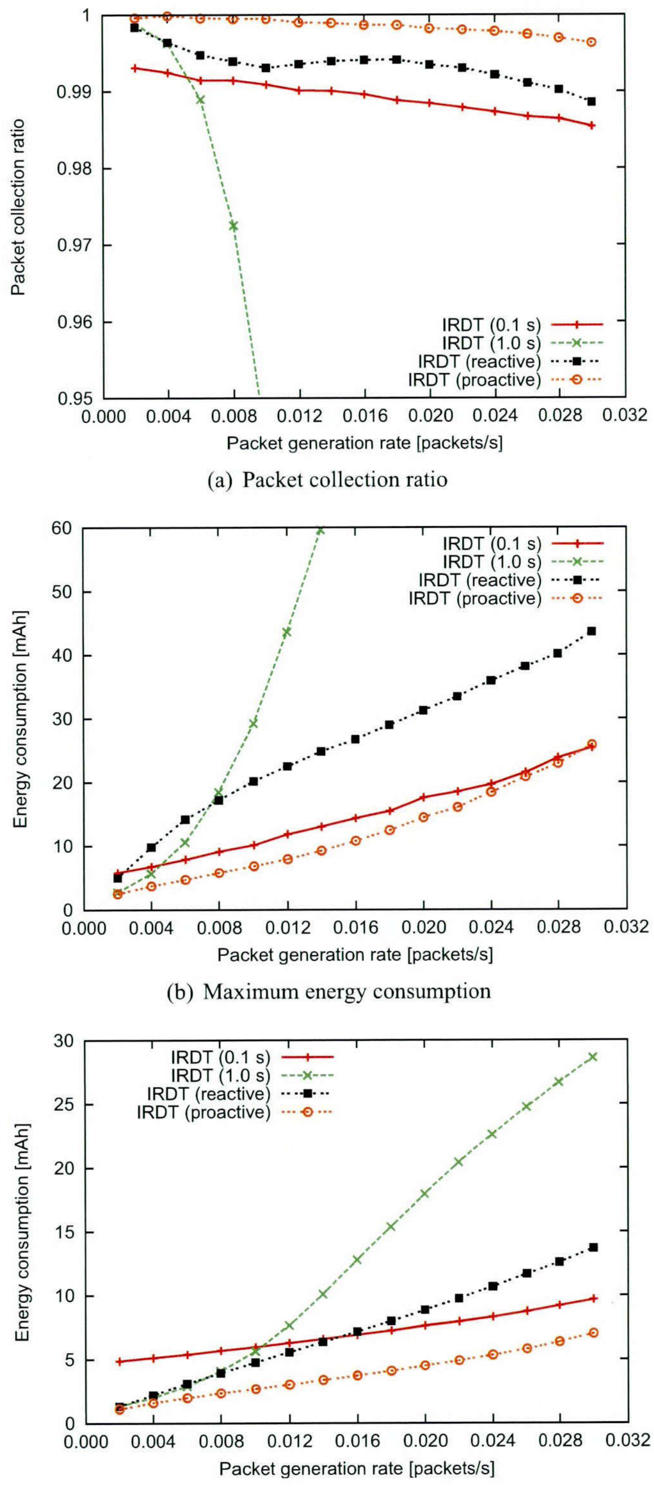
are achieved. In particular, the collection ratio in the proactive method is over 99.5% even when the packet generation rate is 0.030 (Figure 2.13(a)). This result indicates that IRDT can perform efficiently even at comparatively high packet generation rates.

Regarding the maximum energy consumption, its increase can be suppressed by using the proactive method with an interval of T*, as shown in Figure 2.13(b), due to the prevention of control message collisions. Although the reactive method can also reduce energy consumption, except in the case of a packet generation rate of 0.002, it consumes larger amounts of energy than the original IRDT with the 0.1 s interval since the reactive mechanism attempts to avoid collisions after at least one collision has occurred. If SREQ collisions tend to occur in the neighboring nodes of the sink, for example, if there is a large number of such nodes, the improved IRDT is more effective even than the original at nodes adjacent to the sink. Additionally, preventing recurring SREQ collisions and shortening the ID waiting time can decrease energy consumption.

An intermittent interval of T* results in a 50% reduction of the maximum energy consumption as compared with the reactive setting of the intermittent interval at a packet generation rate of 0.002. Although a 40% reduction in energy consumption is also achieved at a packet generation rate of 0.030, with the proactive method the consumption of energy is as high as with the original IRDT with an interval of 0.1 s.

Since the shorter intermittent interval derived from T* yields greater chances of receiving data packets, this leads to implosion of the traffic. Therefore, a load balancing mechanism is necessary in order to reduce the maximum energy consumption, and this issue is investigated in our other research [59].

Regarding the average energy consumption, when packets are generated infrequently, both the



(a) Packet collection ratio

(b) Maximum energy consumption

(c) Average energy consumption

Figure 2.13: Improved performance of IRDT using reactive and proactive setting of the intermittent interval

proactive and the reactive method suppress the increase in energy consumption, while at intermittent intervals of higher frequency, the reactive method consumes more energy than the original IRDT with an interval of 0.1 s. In addition, a reduction of 15% and 48% in average energy consumption is attained when the packet generation rate is 0.002 and 0.030, respectively.

Data Aggregation

The performance of IRDT with the data aggregation function is shown in Figure 2.14, where the number in the label denotes how many data packets can be included in a single aggregated data packet. Immediately after the reception or generation of data, each node waits for 5.0 s for aggregation without forwarding. When the intermittent interval is 1.0 s, the packet collection ratio increases with data aggregation (up to two data packets), after which it deteriorates with aggregation of three or more data packets. At an intermittent interval of 0.1 s, data aggregation always decreases the collection ratio since large data packets are likely to collide with ID messages. Moreover, the loss of aggregated data packets greatly decreases the collection ratio. In summary, our conclusion on the collection ratio is that aggregation of up to two data packets is effective in terms of avoidance of SREQ collisions, while aggregation of three or more packets is disadvantageous.

The maximum and the average energy consumption in all cases other than '0.1 s (3)' decreases as the number of aggregated data packets increases [Figure 2.14(b) and 2.14(c)]. However, when the packet generation rate is low, data aggregation seldom occurs during the waiting time of 5.0 s, and the energy efficiency does not increase considerably.

Note that the increase in average energy consumption for the '0.1 s (3)' case indicates that the increase in retransmissions due to ID collisions increases the number of data retransmissions everywhere in the network. For aggregation of up to three data packets when the packet generation rate is 0.030, a reduction in the maximum energy consumption of 83% and a reduction in the average energy consumption of 77% can be attained at an intermittent interval of 1.0 s. Moreover, the respective reduction of the maximum and the average energy consumption is 60% and 10% at an interval of 0.1 s. These improvements are achieved in particular by forwarding data to sideward nodes, which effectively suppresses SREQ collisions in nodes adjacent to the sink.

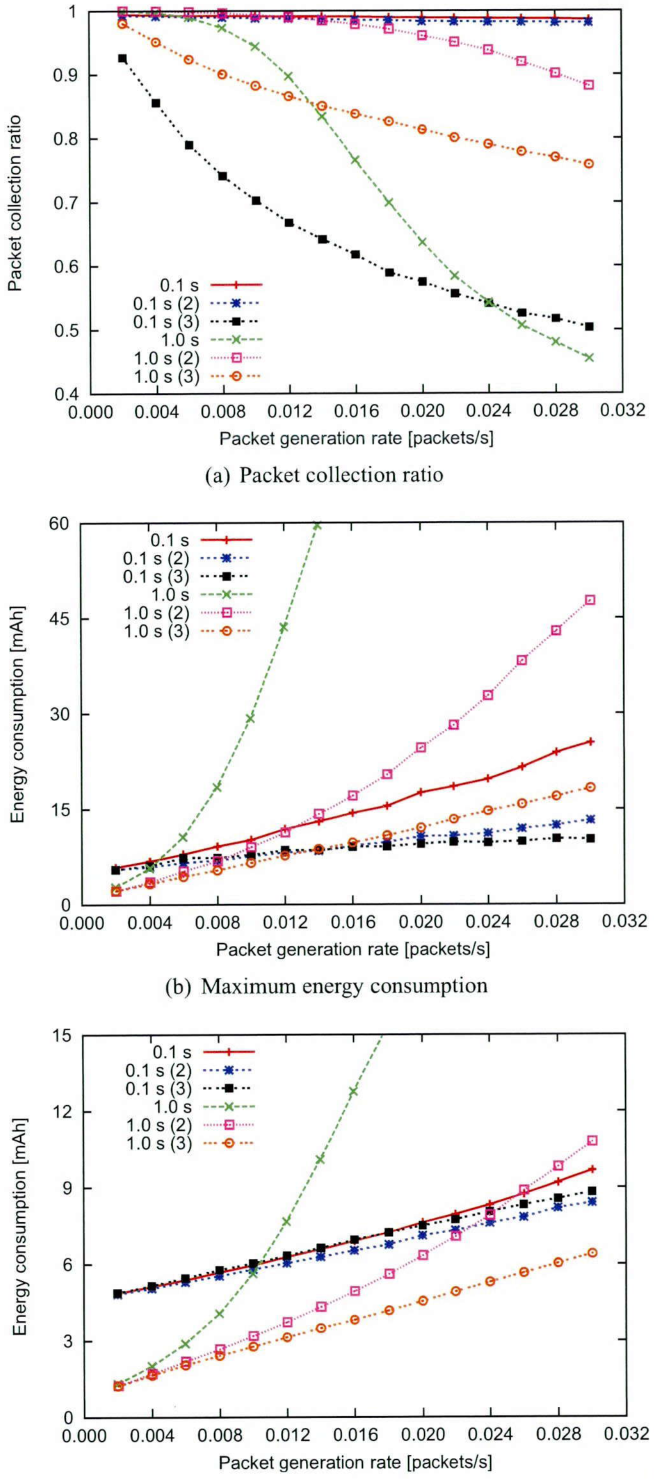
Combinations of Intermittent Interval Setting and Data Aggregation

We compare the performance of IRDT with both the proactive collision avoidance method and data aggregation with that of EA-ALPL [8] as described in Figure 2.15, where data aggregation is limited to two data packets to prevent the packet collection ratio from decreasing. To conduct a fair comparison, EA-ALPL also uses data aggregation and an appropriate intermittent interval which minimizes the energy consumption (although it does not minimize message collisions). However, due to the MAC layer protocol (B-MAC) of EA-ALPL, the intermittent interval is limited to 8 values (10, 20, 50, 100, 200, 400, 800, 1600 ms) [6]. Therefore, out of these eight values, EA-ALPL selects the value that is closest to the appropriate interval.

The results show that IRDT attains a higher collection ratio than EA-ALPL. In addition, IRDT has lower maximum and average energy consumption at all times, as seen in Figure 2.15(b). Specifically, the maximum and the average energy consumption at a packet generation rate of 0.002 can be reduced by 61% and 38%, respectively, although those at a packet generation rate of 0.030 can be reduced by only 0.1% and 45%, respectively. Moreover, a 90% reduction of the maximum energy consumption and an 84% reduction of the average energy consumption is achieved as compared with the original IRDT at an intermittent interval of 1.0 s. It is important to lower the maximum energy consumption for long-term operation of the network, and in this regard the avoidance of control message collisions is highly efficient.

2.5 Summary

In this chapter, we studied the basic performance characteristics of the receiver-driven asynchronous system IRDT. We also investigated the relation between control message collisions and the intermittent interval and examined the efficacy of two simple settings of the intermittent interval and data aggregation in a comparison between IRDT, RI-MAC, and X-MAC, which is a sender-driven asynchronous system, by constructing a computer simulation. As a result, a reduction of 33% in the average energy consumption was achieved with IRDT as compared with RI-MAC and X-MAC. Furthermore, as compared with the original IRDT, the maximum energy consumption was reduced by 90%, and the average energy consumption was reduced by 84%.

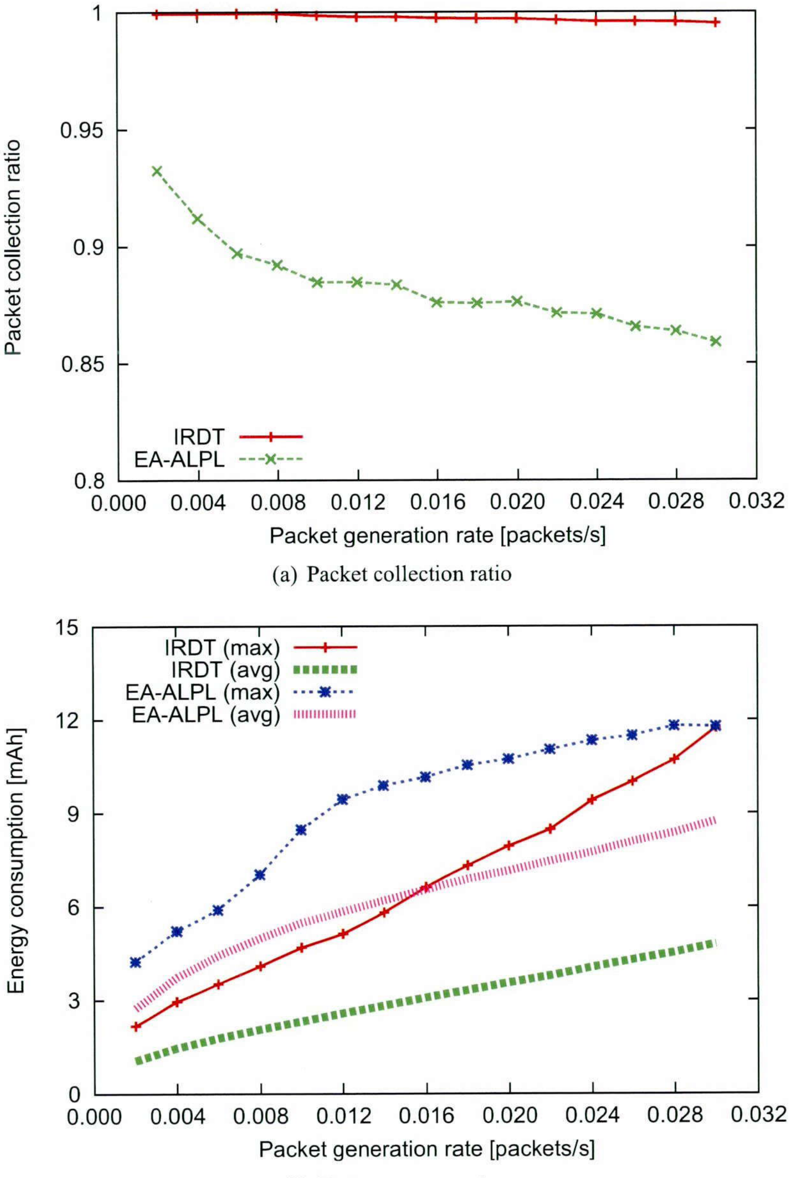


(a) Packet collection ratio

(b) Maximum energy consumption

(c) Average energy consumption

Figure 2.14: Improved performance of IRDT using data aggregation



(a) Packet collection ratio

(b) Energy consumption

Figure 2.15: Improved performance of IRDT using data aggregation and T^*

Chapter 3

Robustness and Resilience in MAC and Routing Layer Protocols for Wireless Sensor Networks

3.1 Quantative Definitions of Robustness and Resilience

We define robustness and resilience as the properties that “maintain” and “recover” performance in the face of unexpected environmental variations, respectively. In this section, we intuitively propose quantitative expressions for robustness and resilience based on Figure 1.2 and discuss how to improve them.

Suppose that measures of network performance, such as the packet delivery ratio, the average end-to-end delay, or the total energy consumption, are linearly related to time. Such assumptions are beyond question when a system is operating ideally, and of course, when measurement results between regular time intervals are constant. Explicitly, robustness is the property that reduces instability in those constants immediately before and after variations, and resilience is the property with which that constant values are recovered immediately after variation to the previous stable values. Here, we define robustness and resilience (denoted by R_b and R_s respectively) according to the following expressions:

$$R_b = \frac{|\overline{C_{before}} - \overline{C_{after}}|}{\overline{C_{before}}}, \tag{3.1}$$

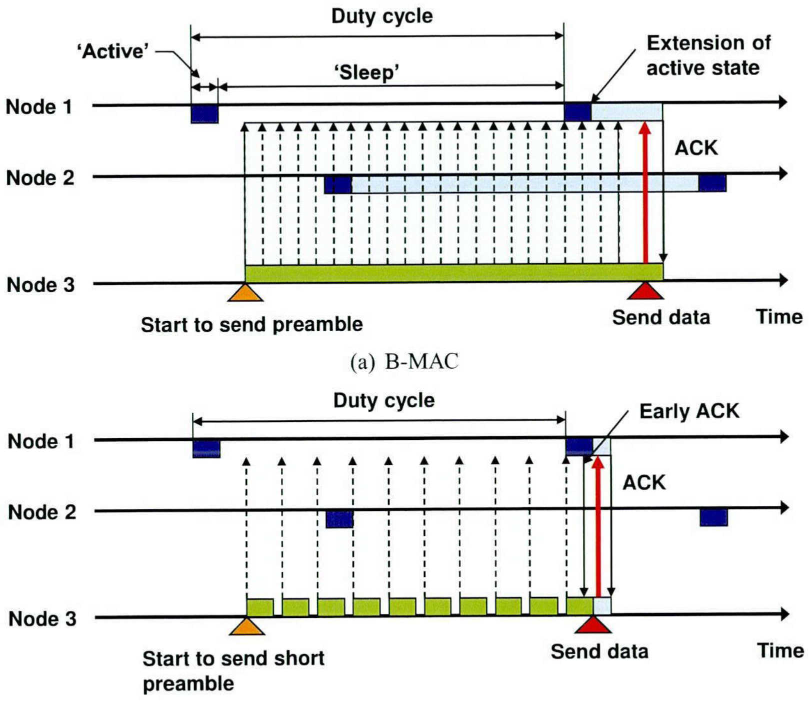
$$R_s = T_{recovery} - T_{variation}, \tag{3.2}$$

where $\overline{C_{before}}$ and $\overline{C_{after}}$ are the short-time average performance immediately before and after environmental variation, respectively; $T_{variation}$ is the time at which the environmental variation occurs; and $T_{recovery}$ is the R %-recovery time after the variation (for constant R). Specifically, R_b is the relative change in performance immediately before and after a variation, and R_s is the time that elapses between the occurence of the variation and recovery of the performance to R % of that immediately before the variation. Clearly, from these definitions, smaller values of R_b and R_s imply greater robustness and resilience of network performance.

In order for improvement of robustness, retransmission mechanisms are of important. In the MAC layer, the one-to-one message retransmission advances robustness of the data delivery, and the routing layer can enhance robustness by utilizing alternative and detour paths. These mechanisms keep the packet delivery ratio stable and some time-to-live (TTL) metrics curb a rapid increase of the delay time and the energy consumption. In order to increase resilience, mechanisms that monitor network conditions and operate adaptively to the conditions are essential. In the MAC layer, there exists an appropriate duty cycle by which high data delivery ratio and low energy consumption are attained. Great resilience is obtained by setting a suitable duty cycle for a node adaptively. Resilience in the routing layer is acquired by grasping exact route information, so highly-frequent exchanges of route information are indispensable factor.

3.2 Robustness and Resilience in MAC Protocols

Considerable importance is placed on the energy efficiency of MAC layer protocols in sensor networks [60], and many duty-cycle MAC protocols have been proposed [6–8, 11, 12, 57, 61]. Therefore, we examine duty-cycle MAC protocols in this thesis. Power-saving operation in duty-cycle MAC protocols is based on the fact that sleeping nodes consume significantly less energy than



(a) B-MAC

(b) X-MAC

Figure 3.1: Sender-initiated MAC protocols

idling nodes [13]. However, since nodes turn off their wireless interfaces, the nodes must control their wake-up timings in order to communicate with other nodes. According to whether the sender or receiver initiates communications, duty-cycle MAC protocols are respectively classified into two types: sender-initiated [6–8, 11, 12] and receiver-initiated MAC protocols [57, 61]. In the subsequent sections, we describe both sender-initiated and receiver-initiated MAC protocols, and show that the difference between these two types is essentially between hard- and soft-states. Furthermore, the "soft-state", which is often referred to in network protocol designs [62–65], is important for robustness improvement.

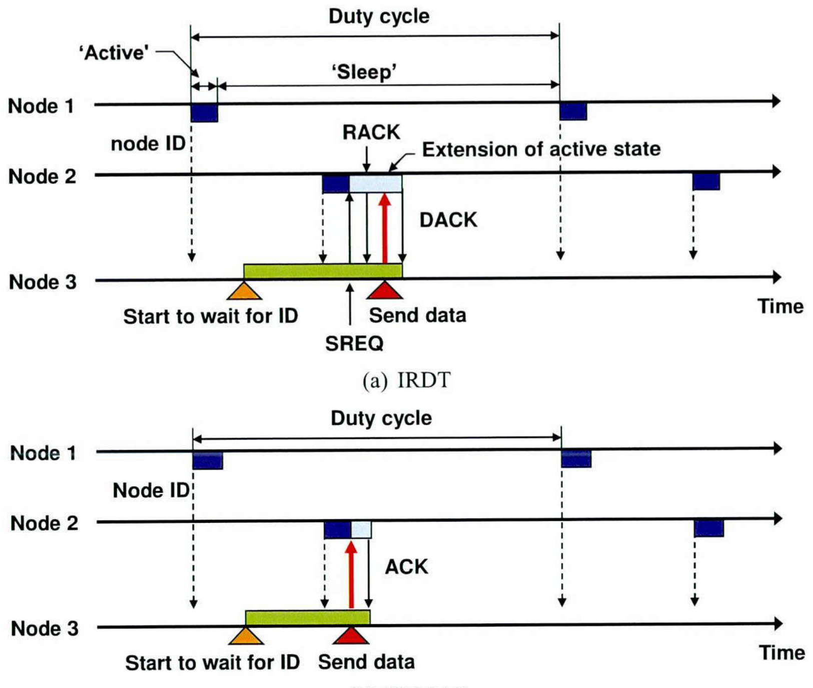
3.2.1 Sender-Initiated MAC Protocols

B-MAC [6] is the basis of *low power listening* (LPL) protocols in which receiver nodes periodically probe the state of the channel (Figure 3.1(a)). Figure 3.1(a) presents an instance where node 3 (the sender) is ready to send a data packet to node 1 (the receiver). If the channel is idle, the receiver returns to the sleep state after probing. In contrast, if the channel is busy, preparations are made to be ready for data reception. After receiving intended data, node 1 returns an acknowledgement (ACK) message. To activate the channel and initiate communication, the sender sends a continuous preamble over a period of time that is longer than the duty cycle. The sender then sends the data after sending the preamble. A number of shortfalls are found in using this protocol. As the duty cycle increases, each sender node occupies the channel for a longer period of time during preamble transmission. Such occupation of the channel then interferes with communication between neighboring nodes. Moreover, preamble transmission from the sender consumes the power of unrelated receivers, and is known as the overhearing problem.

X-MAC [7] (Figure 3.1(b)) was designed to solve the overhearing problem of B-MAC. To prevent the sender preamble in B-MAC from occupying the channel, X-MAC continuously transmits short preambles to which the ID of a certain receiver is appended. The receiver node then replies with an early ACK when the appended ID corresponds to its own. After receiving this early ACK, the sender transmits the data packet and waits for the ACK of the data. Thus, receivers that detect unrelated short preambles can resume their sleep state soon after the end of data reception and the overhearing problem generated in B-MAC by continuous preamble transmission is solved.

3.2.2 Receiver-Initiated MAC Protocols

As discussed in an early chapter, *Intermittent receiver-driven data transmission* (**IRDT**) is a receiver-initiated MAC protocol that was developed and is actually used for products with meters [14]. In previous chapter, we clarified the performance of IRDT by comparing this performance with that of the sender-initiated MAC protocol, *energy-aware adaptive LPL* [8]. As shown in Figure 3.2(a), receivers that are ready to receive data transmit small messages containing their ID in order to inform the senders. A sender waits for an appropriate receiver's ID, and after acquiring this ID, the sender establishes a link with the receiver by returning a send request (SREQ). After getting a request acknowledgement (RACK) for the SREQ, the sender then transmits the data packet and finishes communication following receipt of a data acknowledgement (DACK). Another receiver-initiated



(a) IRDT

(b) RI-MAC

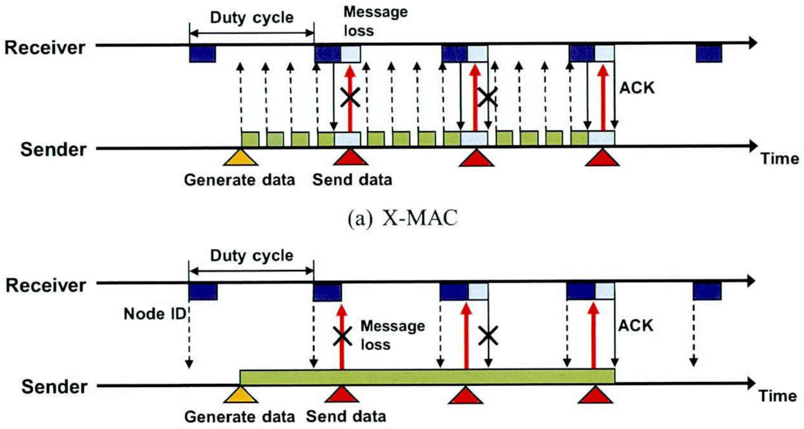
Figure 3.2: Receiver-initiated MAC protocols

Protocol is the receiver-initiated MAC (**RI-MAC**) [57] which is a simple type of RIT. In RI-MAC, the sender transmits the data packet immediately after receiving an appropriate ID (Figure 3.2(b)).

Two types of message collisions cause critical problems in receiver-initiated MAC protocols:

1. Periodical ID transmissions can interfere with other nodes' communication. To avoid these collisions, receiver-initiated MAC protocols exploit channel clear assessment before transmitting an ID, and a node terminates transmission of the ID if the channel condition is busy.

2. When a receiver transmits its ID and multiple senders possess data for the receiver, transmission from different senders of multiple SREQs in IRDT or multiple data packets in RI-MAC may result in collision. To avoid these collisions, both RI-MAC and IRDT use collision detection and exponential backoff.



(a) X-MAC

(b) RI-MAC

Figure 3.3: Retransmission-procedures in two MAC protocols

3.2.3 Difference in Robustness and Resilience Between Sender-Initiated and Receiver-Initiated MAC Protocols

In the previous section, we defined robustness and resilience as those properties that maintain and recover performance when environmental changes occur. Since the main role of the MAC layer is one-to-one data communication, we do not consider a node failure and a link failure. Instead, we consider environmental changes due to sharp increases in traffic load, which incur congestion and message collisions. For robustness and resilience to traffic increases in the MAC layer, changes to the receiver's condition must be detected. To maintain performance, senders should retransmit data packets only if the receiver's normal operation is confirmed, and this requires monitoring.

Although a retransmission mechanism is naturally applicable to both sender-initiated and receiver-initiated MAC protocols as shown in Figure 3.3, detecting changes in the receiver's condition is nontrivial for sender-initiated MAC protocols. To monitor the receiver's condition, a sender must transmit messages to the receiver in sender-initiated MAC protocols; however, when no response is given by the receiver, the sender cannot distinguish between failure of the receiver and failure of message reception. Conversely, in receiver-initiated MAC protocols, the receiver periodically transmits its ID and shows evidence of its existence. If a sender waiting for a particular receiver's ID does not receive this ID for a period of one or more duty cycles, the sender conjectures that the receiver has failed or—since the receiver does not transmit an ID when its buffer is full—is congested. Eventually, senders in sender-initiated MAC protocols must retransmit data packets repeatedly until they achieve success, since they cannot know the receiver's condition. In receiver-initiated MAC protocols, senders will retransmit data if they receive the receiver's ID, and senders will discard their data packets if they do not receive this ID for a period of one or more duty cycles.

This procedure in receiver-initiated MAC protocols is similar to soft-state protocols. In soft-state protocols, periodical refresh messages are used, and a node that receives an intended refresh message maintains its state for as long as such refresh messages arrive. When the node cannot receive a refresh message within a given time period, it returns to its default state. As Lui et al. [45] stressed, soft-state protocols are robust to unanticipated fluctuations. In contrast, because senders cannot get information about a receiver's condition in sender-initiated MAC protocols, senders continue to transmit preambles as if the receivers were operating normally, similar to hard-state protocols.

To improve resilience, MAC protocols must detect congestion and select an appropriate duty cycle. Nevertheless, sender-initiated MAC protocols cannot distinguish interference from traffic congestion, and so we do not discuss their resilience in this chapter. In receiver-initiated MAC protocols, receivers perceive network congestions when bit errors (most likely caused by collisions) are detected in SREQ messages or in data packets received immediately after transmitting ID messages. In such circumstances, receivers increase their duty cycles; otherwise, receivers decrease their duty cycles or leave them unchanged.

3.3 Robustness and Resilience in Routing Protocols

Here, our focus is on robustness and resilience to route changes induced by severe environmental changes. To ensure robustness and resilience to route changes caused by node failure or energy depletion, both connectivity assurance between adjacent nodes and reachability confidence from sensor nodes to the sink node are required. To maintain performance when node failure occurs, aggressive use of detours and alternate routes is shown to be useful. In more severe cases, such

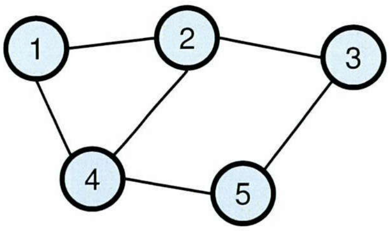


Figure 3.4: Simple network model for explanation of routing protocol

as failure of the destination node, data sent from sensor nodes cannot be correctly collected and the performance of the system eventually degrades. Here, the quick response of routing tables is indispensable for resilience. In this section, we demonstrate the robustness resulting from multipath detour routing over a mesh network and the resilience resulting from soft-state management of routing tables.

3.3.1 Management of Routing Tables for A Simple Distance Vector Routing

We adopt a simple distance vector routing (DVR) to provide a definite discussion of the routing table management. In DVR, all nodes have routing and distance matrix tables such that the distance to any node in the network can be calculated. DVR then performs periodic updates where each node sends its routing table to its neighbors. In our simple DVR, the distance metric is the number of hop counts. Therefore, a node's routing table contains its hop counts to all nodes in the network, and by exchanging this routing table with their neighbors, each node can create their "hop matrix table" (distance table).

To begin, we explain the routing tables. In following description, we use the simple network model shown in Figure 3.4. We refer to the node with a unique ID k as node k, and we define $H(m, n)$ as the hop count from node m to node n. Initially, node n registers in the routing table that $H(n, n)$ is zero. When node n receives any type of message (e.g., a HELLO message or messages in MAC layer) sent from node m, node n registers on its routing table that $H(m, n)$ is one; that is, node n refers to node m as a neighboring node. To calculate the minimum hop counts for nodes with distances greater than one hop away, each node must iteratively exchange its routing table with

Original owner : node 2	
TSN: 15	
Node ID	Hop
1	1
2	0
3	1
4	1
5	2

(a) Node 2's routing table

Original owner : node 1	
TSN: 15	
Node ID	Hop
1	0
2	1
3	2
4	1
5	2

Original owner : node 3	
TSN: 12	
Node ID	Hop
1	2
2	1
3	0
4	2
5	1

Original owner : node 4	
TSN: 18	
Node ID	Hop
1	1
2	1
3	2
4	0
5	1

(b) Routing tables received by node 2 from neighbors

Figure 3.5: Routing tables of node 2 in Figure 3.4

		Destination ID				
		1	2	3	4	5
Receiver ID	1	1	0	3	2	3
	2	0	0	0	0	0
	3	3	0	1	3	1
	4	2	0	3	1	1
	5	0	0	0	0	0

Figure 3.6: Hop matrix table of node 2 in Figure 3.4

its neighboring nodes. This table exchange is performed with constant period, T_i. All routing tables are given a table sequence number (TSN) that is used to determine whether to exchange routing tables, and TSN is incremented when the node's routing table is updated. Figure 3.5(a) shows an example of the routing table of node 2 in which the minimum numbers of hop counts from node 2 to all nodes have been registered. This table is calculated using routing tables of node 2's neighbors, as shown in Figure 3.5(b).

Each node's corresponding hop matrix table (denoted M) is then represented by an $N \times N$ matrix (Figure 3.6), where N is the number of nodes in the network. Here, we define r_{ij} as the element in row i and column j of M such that i corresponds to the receiver node ID and j to the destination node ID. Each r_{ij} is assigned an integer value that indicates the type of relays to destination node i by way of receiver node j as follows. Given sender node n whose destination is node i, if node n receives an ID from node j, node n compares $H(n, i)$ in its routing table with $H(j, i)$ in the routing table received from node j. If $H(n, i) - H(j, i) = 1$, then node n sets r_{ij} to equal to one. If $H(n, i) - H(j, i) = 0$, then r_{ij} is set equal to two, and if $H(n, i) - H(j, i) = -1$, then r_{ij} is set equal to three. Otherwise, node j is not a neighbor of node n and r_{ij} is set to zero. In addition, we define "forward", "sideward", and "backward" nodes. For node n with destination node i, if r_{ij} is one, then node j is a forward node. In a similar manner, if r_{ij} is two, then node j is a sideward node; if r_{ij} is three, then node j is a backward node; and if r_{ij} is zero, then node j is a non-neighboring node. An example of the hop matrix table of node 2 in the five node network shown in Figure 3.4 is given in Figure 3.6. The elements in this hop matrix table are calculated based on the routing tables shown in Figure 3.5.

3.3.2 Detour Routing over a Mesh Network

Many studies have been conducted on routing protocols in wireless sensor networks [66]. The majority of these studies use single-path routing algorithms in which all nodes forward data to a single predetermined node according to a metric such as energy efficiency. However, in the case of a link error or node failure, controlling detours and alternative routes is considered to be effective [67]. To examine the robustness of networks, we assume a multihop wireless mesh sensor network and a hop-by-hop routing algorithm. In our single-path routing, each node forwards data packets to one of the forward nodes registered in its hop matrix table. To explain our routing procedure, we define the "routing function".

A routing function is a logic function that determines the transmission of a data packet. The flowchart of an example routing function is shown in Figure 3.7. The function in this figure assumes a routing based on a minimum hop routing, where detours occur when a "sideward-relay condition" is satisfied. Thus, alternative routes exists in minimum hop routing, and a detour is employed by selecting a sideward node as the next hop. An example sideward-relay condition is that "true" is returned when a node fails to transmit a data packet to all of its forward nodes. Note that we append

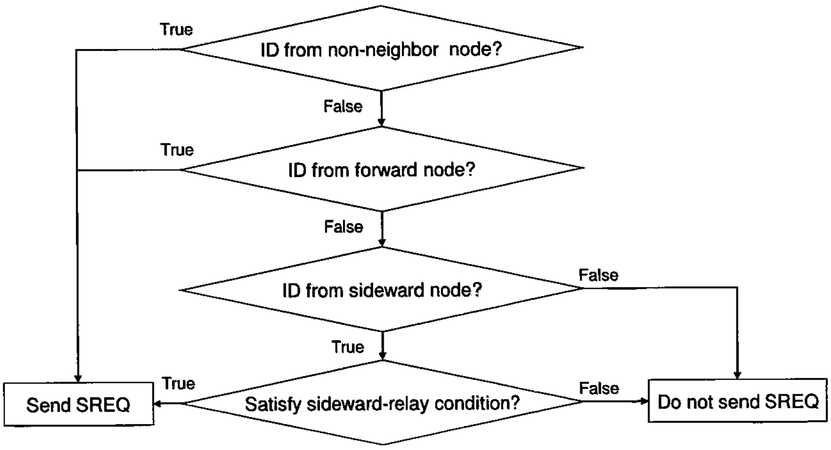


Figure 3.7: Flowchart of routing function

a TTL value in each data packet to prevent the sideward relays from causing a routing loop.

3.3.3 Connectivity and Reachability Management

Next, we present soft-state management of routing tables to improve resilience. Soft-state management, which is used for neighbor relationships and routing tables, is briefly described as follows. If node i does not receive a message from node j during a specified time period, then node i sets $H(i, j)$ to a default value (e.g. a maximum value of the integer variable), removes the routing table received from node j, and recalculates its own hop matrix table.

Under DVR, each node has a routing table in which the hop counts are registered from all nodes in the network. When node i receives a message from node j, node i registers that $H(i, j)$ is in its routing table, and we call this a neighbor relationship. Neighbor relationships in a node's routing table can thus be maintained by probing a message. To manage the relationships, we add a time stamp to each item in a routing table. Each node waits for a message for length of time T_p every T_i, and when the node gets a message during T_p, it updates the time stamp that corresponds to the sender of the message to the current time (in this chapter, T_p is always set to the same value as the duty cycle). This procedure is similar to sender-initiated MAC protocols, which periodically probe the wireless channel. To maintain neighbor relationships in sender-initiated MAC protocols, a node periodically broadcasts data packets containing its routing table to its neighboring nodes. Moreover, in receiver-initiated MAC protocols, a node has to periodically broadcast its routing table, since it probes for an ID to transmit data packets. Therefore, by adding a TSN into an ID message or a short preamble, a receiver can inform the sender whether it requires the routing table identified by the TSN. If the receiver does not need the routing table, it does not transmit an SREQ message, data packet, or data required (DREQ) message. Since strong dependence on past conditions prevents quick responses to sudden changes, when a node does not receive a message from a neighbor within nT_i (where n is constant), the node sets the hop count associated with the former neighbor to infinity, and we call this soft-state connectivity management. After sampling, the node recalculates its routing table by using the tables received from its neighbors.

Receiving routing tables from neighboring nodes is necessary for each node to complete its own routing and hop matrix tables. Here, we also introduce a soft-state management into routing tables. To this end, we add time stamps to the routing tables in addition to the management of neighbor relationships. When a node does not receive a message from a neighbor within T_i, the node deletes the neighbor's routing table. This soft-state management of routing tables thus maintains the reachability of a node to its destination. Note that the management of neighbor relationships and routing tables are done simultaneously.

3.4 Simulation Results

We evaluate robustness and resilience in the MAC and routing layers by using an event-driven simulator written in visual C++, where all results are averaged over 300-time simulations. We employ the disk model of communication between nodes, in which the strength of the radio signals does not deteriorate, and—unless packet collisions occur—a transmitted packet is assumed to be received by nodes within the communication range. In addition, our evaluation is made on safe side; if a collision with other messages occurs while a message is being received, the messages are simply discarded. The parameters in our simulation are set to the values shown in Table 3.1.

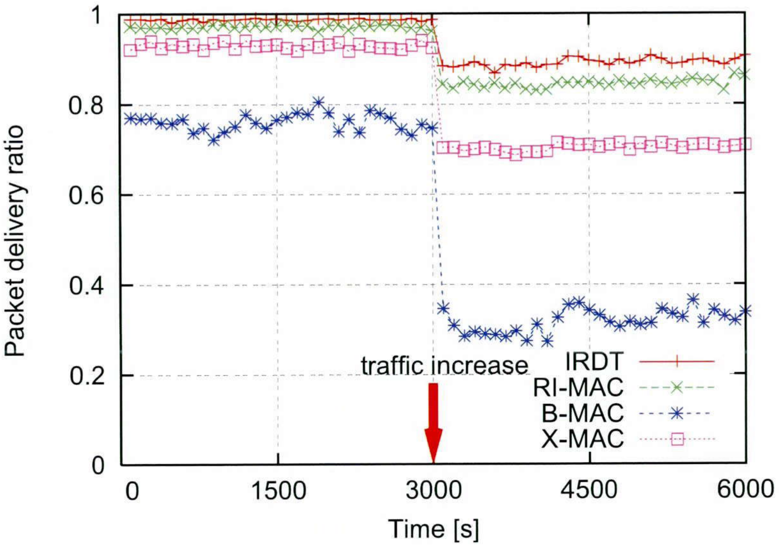
Table 3.1: Parameter settings for robustness evaluation

Parameter	Value
Transmission speed	100 kbps
Communication range	100 m
Duty cycle	1.0 s
Current consumption (TX)	20 mA
Current consumption (RX)	25 mA
Current consumption (SLEEP)	0 mA
Message size (ID, SREQ, DREQ)	24 Byte
Message size (RACK, DACK, ACK)	22 Byte
Packet size (DATA)	128 Byte

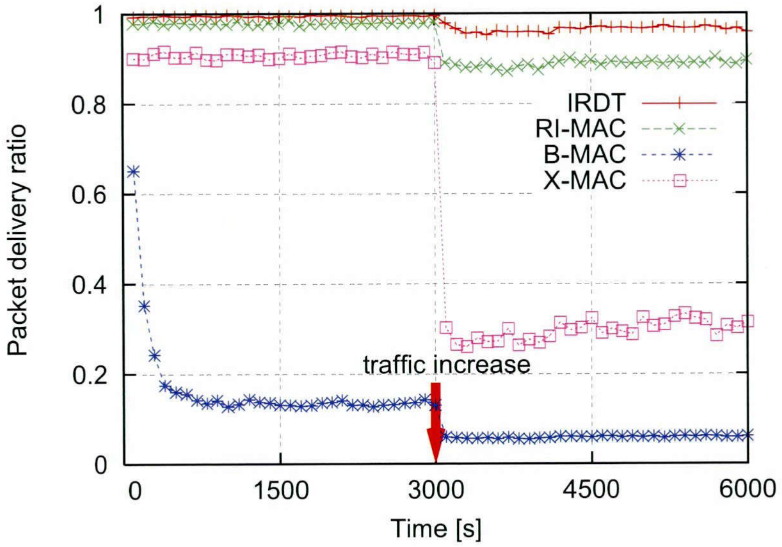
3.4.1 Robustness in MAC Protocols

Since the main function of the MAC layer is one-to-one data communication, we take into no consideration of a node and link failures. Our evaluations are on robustness and resilience against message loss caused by interferences and message collisions. We examine the packet delivery ratio in the case where 30 sensor nodes generate data packets according to Poisson process (with $\lambda = 0.003$) and data is sent to a single sink node. At the same time, we examine the effects on the total energy consumption of improving robustness on the packet delivery ratio. We assume a star network topology in which sensor nodes are deployed with equal angles in a circular pattern. The sink node is then at the center of this circle. The radius of the circle is equal to the communication range, and therefore severe interference can occur and many hidden nodes exist in the network. To evaluate the robustness and resilience of the network, at 3000 s in the simulation, extra 30 sensor nodes are added. Here, the scheduled timer for discarding data (T_d) is set to 2.0 s or 10 s. In the sender-initiated MAC protocols, when a sensor node cannot complete communication with the sink node within T_d, the node drops its data packet. However, in the receiver-initiated MAC protocols, a sensor node retains its data packet as long as an ID from the sink node can be obtained every T_d.

Figures 3.8 and 3.9 show the packet delivery ratio and energy consumption for each 100 s of the two receiver-initiated MAC protocols (IRDT and RI-MAC) and two sender-initiated MAC protocols (B-MAC and X-MAC). Except for B-MAC, the packet delivery ratios and energy consumptions

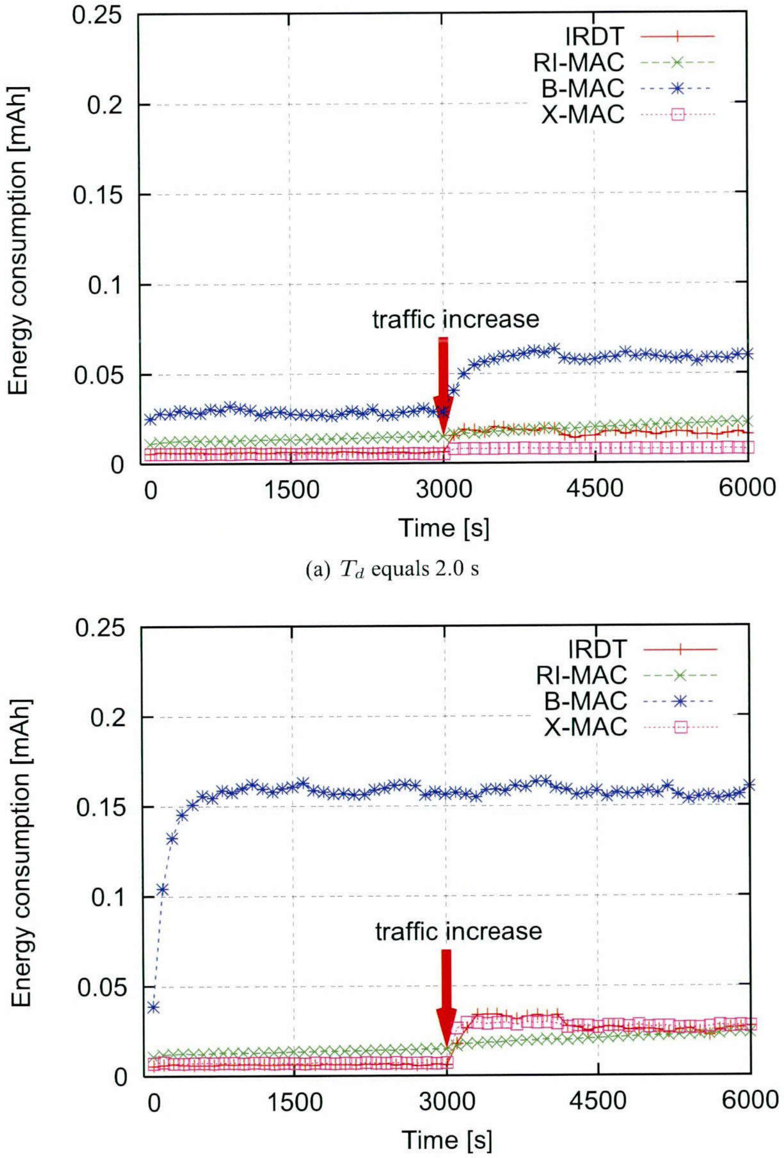


(a) T_d equals 2.0 s



(b) T_d equals 10 s

Figure 3.8: Robustness of a packet delivery ratio in MAC protocols



(a) T_d equals 2.0 s

(b) T_d equals 10 s

Figure 3.9: Robustness of energy consumption in MAC protocols

Table 3.2: R_b of MAC protocols

Protocol	Packet delivery ratio		Energy consumption	
	T_d=2.0 s	T_d=10 s	T_d=2.0 s	T_d=10 s
IRDT	0.104	0.018	1.381	1.856
RI-MAC	0.123	0.094	0.106	0.122
B-MAC	0.536	0.539	0.415	0.007
X-MAC	0.239	0.507	0.660	2.547

of the MAC protocols are not considerably different to each other before 3000 s. In contrast, after the addition of extra nodes, the packet delivery ratio of B-MAC and X-MAC decrease greatly due to message collisions, but the receiver-initiated MAC protocols show good robustness. This is essentially due to the receiver's link-establishment procedure in the receiver-initiated protocols. In sender-initiated asynchronous MAC protocols, since data transmission is initiated at an arbitrary timing, message collisions are essentially inevitable, especially when there are many senders. Meanwhile, in receiver-initiated MAC protocols, since data transmission is conducted after a receiver's ID transmission, message collisions can be avoided in some way. As above-mentioned, RI-MAC and IRDT utilize exponential backoff algorithm to establish a link between a sender and a receiver. In terms of energy consumption, we cannot easily compare the robustness among the four MAC protocols since their packet delivery ratios are different. By definition, B-MAC with T_d of 10 s has the most robust energy consumption but it consumes much more energy. RI-MAC is more robust on average due to our use of the binary exponential backoff mechanism of carrier sense multiple access with collision avoidance for data transmission. After several backoff trials, a sender drops its data packet in RI-MAC, which reduces congestion. The R_b values of the MAC protocols, the relative change of 100-second average performance before and after variations defined in Section 3.1, are listed in Table 3.2. This shows receiver-initiated MAC protocols have about twice robustness of sender-initiated ones.

Table 3.3: R_s (98% recovery) of the receiver-initiated MAC protocol (IRDT)

	Duty-cycle change interval (s)			
	50	100	500	1000
Packet delivery ratio	200	300	700	1200
Energy consumption	300	500	1200	2100

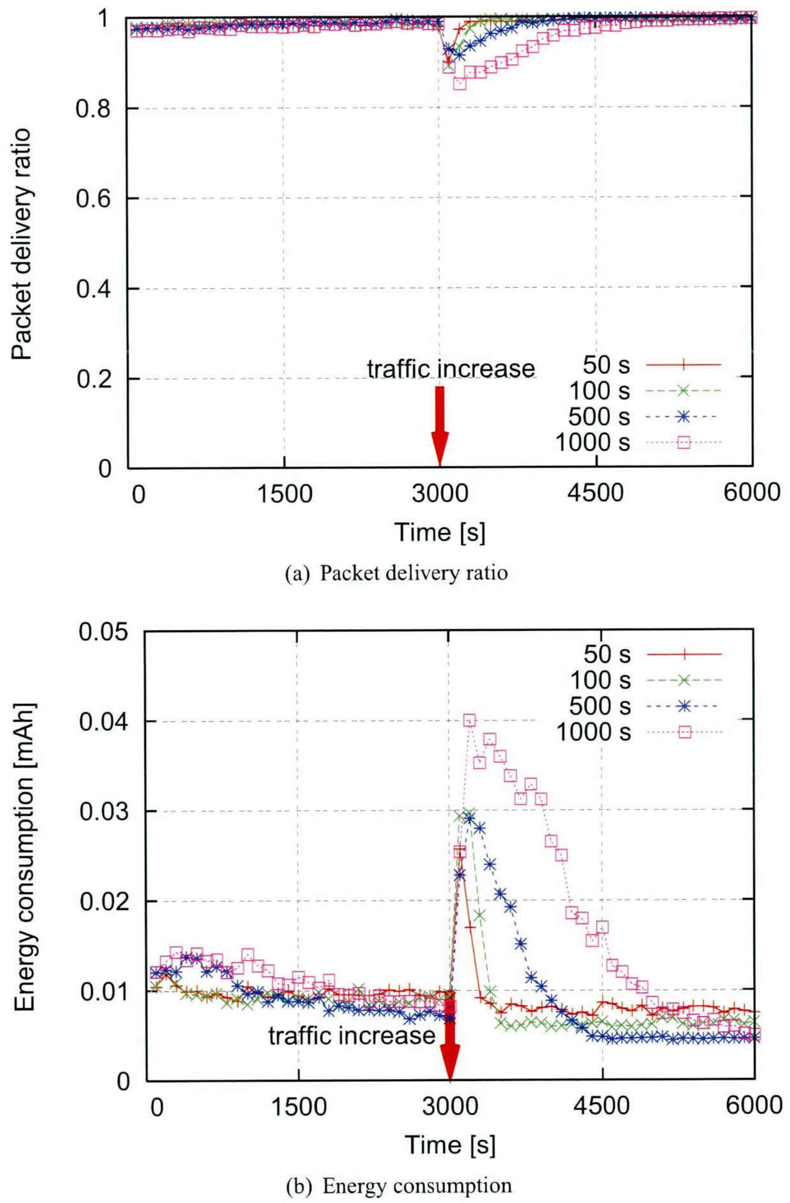
3.4.2 Resilience in MAC Protocols

As discussed in Section 3.2.3, we examine resilience of the receiver-initiated MAC protocols. Particularly, IRDT is evaluated using a similar simulation to that for the robustness measurements. However, here, $\lambda = 0.005$ in Poisson process and T_d is fixed to 2.0 s. To improve resilience, when the sink node (receiver node) detect congestion, its duty cycle is changed every 50 s, 100 s, 500 s, or 1000 s. Specifically, during this interval, if the rate exceeds 0.05 at the sink node that a collision is detected immediately after transmitting an ID, the sink node decreases its duty cycle by 0.2 s. Conversely, if the rate at the sink node is below 0.02, the sink node increases its duty cycle by 0.2 s. In all other cases, the sink node does not change its duty cycle.

Figure 3.10 shows the packet delivery ratio and energy consumption of IRDT for each 100 s and the associated R_s values are listed in Table 3.3 (where R described in Section 3.1 is 98 [%]). From the simulation results, a short interval for changing the duty cycle increases the resilience of the network performance. Note that after node additions, not only the packet delivery ratio, but also the energy consumption shows better performance due to the selection of an appropriate duty cycle.

3.4.3 Robustness in Routing Protocols

Unlike the MAC layer, link and node failures increasingly pose severe problems rather than individual link congestion in the routing layer. In this section, we evaluate the robustness and resilience to node failures and we investigate the R_b and R_s (R is 90 [%]) values of the packet delivery ratio and the energy consumption. Two types of node failures are considered for evaluation: 20 randomly selected sensor nodes fail or one of the sink nodes (denoted by a failed sink) breaks down. Both of these events occur at 1000 s in the simulation.



(a) Packet delivery ratio

(b) Energy consumption

Figure 3.10: Resilience in the MAC protocol

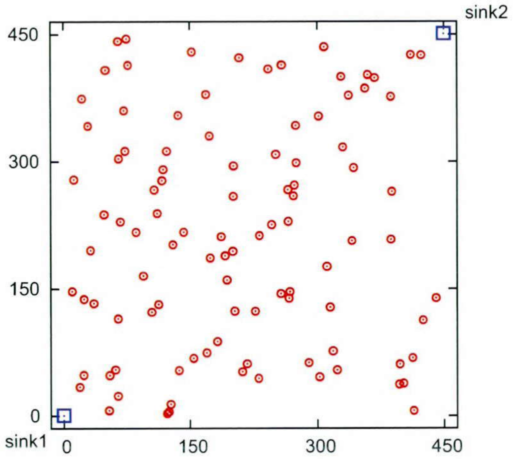


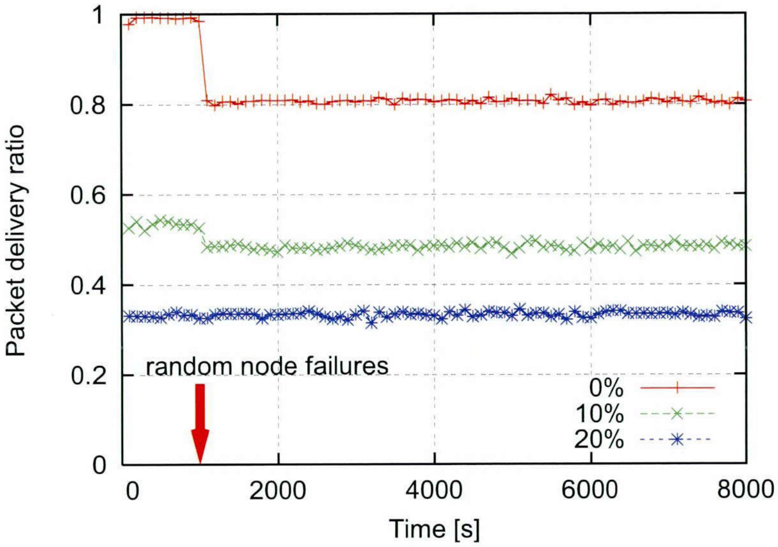
Figure 3.11: An example of network model in which 100 sensor nodes and 2 sink nodes are deployed over a 450 m × 450 m square field

Table 3.4: R_b of the routing protocol

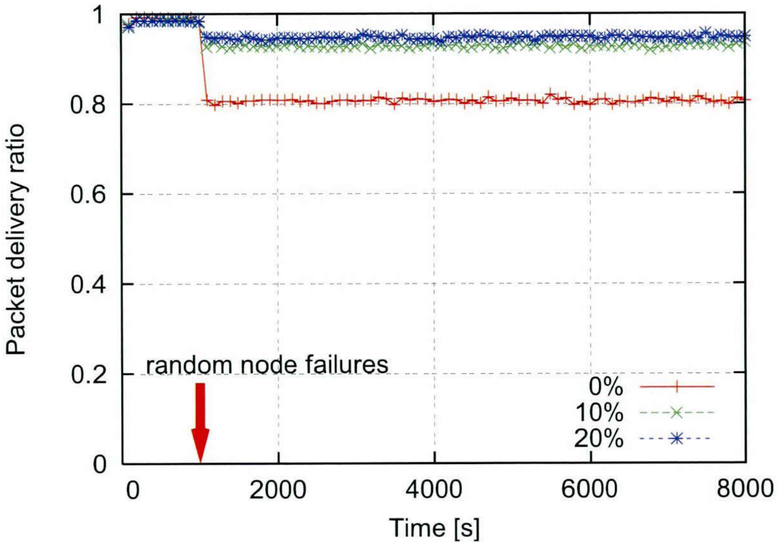
TTL	Packet delivery ratio			Energy consumption		
	0 %	10 %	20 %	0 %	10 %	20 %
Hop count	0.178	0.080	0.001	0.192	0.067	0.020
3(Hop count)		0.055	0.035		0.102	0.036

As shown in Figure 3.11, we use a network model in a square (450 m × 450 m) area to conduct our evaluation. One hundred sensor nodes, represented by circles, are randomly deployed within this area and two sink nodes, represented by squares, are positioned in the bottom left and top right corner of the network. Each sensor node generates data packets according to Poisson process with $\lambda = 0.003$ and these packets are sent to the nearest sink node by multihop relay. In our evaluation for robustness and resilience of the routing layer, we use IRDT as a MAC protocol. The simulation commences after an initializing phase in which each node exchanges its routing table with its neighboring nodes and the simulation ends after 8000 s.

In order to investigate robustness itself, all nodes do not exchange routing tables after initializing phase, but utilize alternative and detour paths. The sideward-relay condition used for detour routing is that the sender returns an SREQ message with a fixed probability (0 %, 10 %, 20 %). The

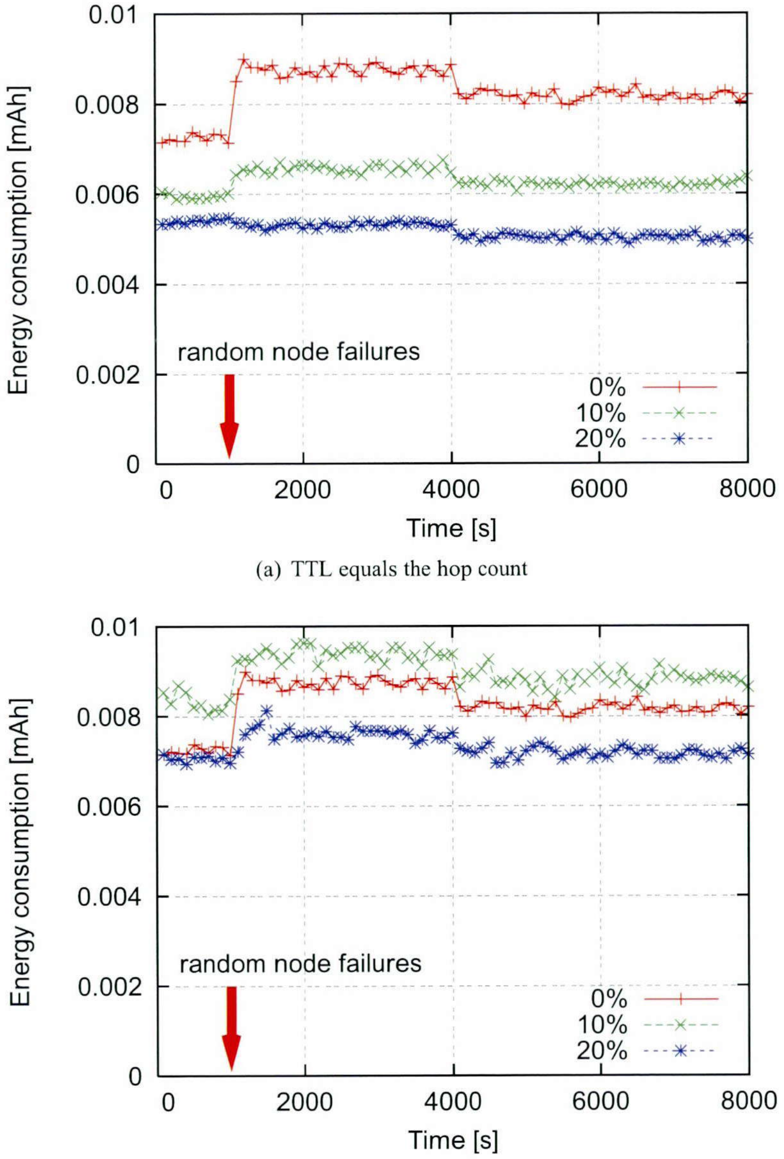


(a) TTL equals the hop count



(b) TTL equals threefold the hop count

Figure 3.12: Robustness of packet delivery ratio in the routing protocol



(a) TTL equals the hop count

(b) TTL equals threefold the hop count

Figure 3.13: Robustness of energy consumption in the routing protocol

robustness of the packet delivery ratio is shown in Figure 3.12, in which multiple nodes fail at 1000 s. After the failure, the packet delivery ratio falls when nodes do not use sideward relays. However, note that after random node failures, more than 80 % of the data packets are still delivered correctly because each node with a failed forward node can use alternative forward nodes. With sideward relays, the packet delivery ratio after the random failures does not considerably decrease, because each node can use a detour by controlling sideward relays. Therefore, the influence of multiple node failure is small in such cases.

In our detour routing, TTL plays a crucial role. Figure 3.12(a) demonstrates that the use of sideward relays degrades the packet delivery ratio. Degradation occurs because once a node transmits a data packet to a sideward node, the data packet cannot reach either sink node since TTL is set to be the same value as the hop count from the nearest sink node. However, if we set TTL equal to threefold of the hop count from the nearest sink node, over 90 % of the data packets reach the sink nodes (Figure 3.12(b)). For the energy consumption, the use of sideward relays intuitively expected to increase the total energy consumption, since the total hop count is increased. However, Figure 3.13(a) shows the opposite result. The main reason for this contradiction is that the use of sideward relays reduces the time for idle listening of a sender node waiting for an ID from receivers. This idle listening is a dominant factor of energy consumption because the idle-listening time (100 milliseconds to seconds) is much longer than the time for message transmissions (milliseconds). When TTL becomes zero at a relay node (not the sink node), the data is discarded without idle listening. Therefore, in case TTL equals to the hop count, sideward relays shorten the time for idle listening. Conversely, in case TTL equals to threefold the hop count, energy consumption increases due to repeated sideward relays. However, 20% sideward relays consume less energy than 10% sideward relays as shown in Figure 3.13(b) because the idle-listening time of a sender gets shorter as the number of multiple receiver candidates increases.

R_b values in the routing layer is listed in Table 3.4 and a more positive use of sideward relays increases the performance robustness. Thus, our detour-routing algorithm has more than tripled robustness than the simple minimum-hop routing algorithm which use alternative paths.

Table 3.5: R_s (90% recovery) of the routing protocol

	Table exchange interval (T_i [s])			
	30	50	100	1000
Packet delivery ratio	1000	1600	3100	over 8000
Energy consumption	200	500	1000	over 8000

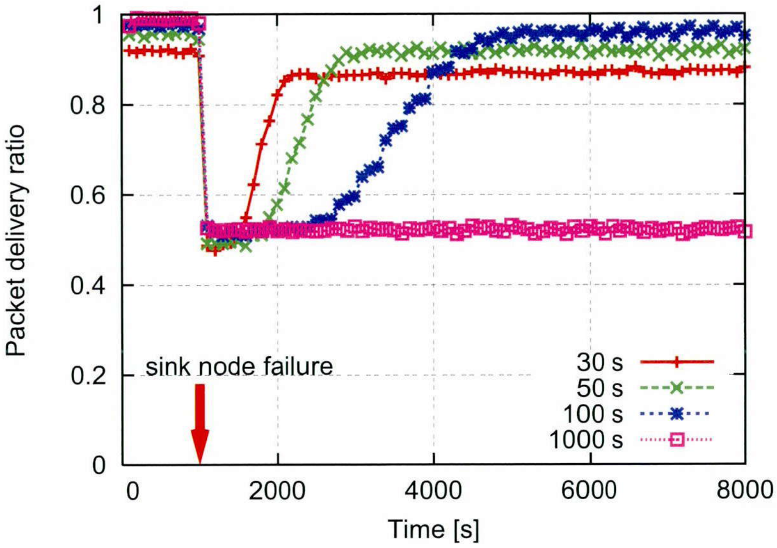
3.4.4 Resilience in Routing Protocols

Finally we investigate resilience to the sink-node (destination-node) failure. In general, since the refresh interval is smaller in a soft-state system, the system has greater flexibility to deal with environmental changes. Namely, with a smaller value of T_i (as described in Section 3.3.3), the network is increasingly resilient to environmental changes. Moreover, shorter T_i potentially leads to a larger overhead energy consumption. Thus, we change T_i (where n is fixed to 3) and evaluate the resilience. Note that all nodes only select a forward node for evaluation on resilience.

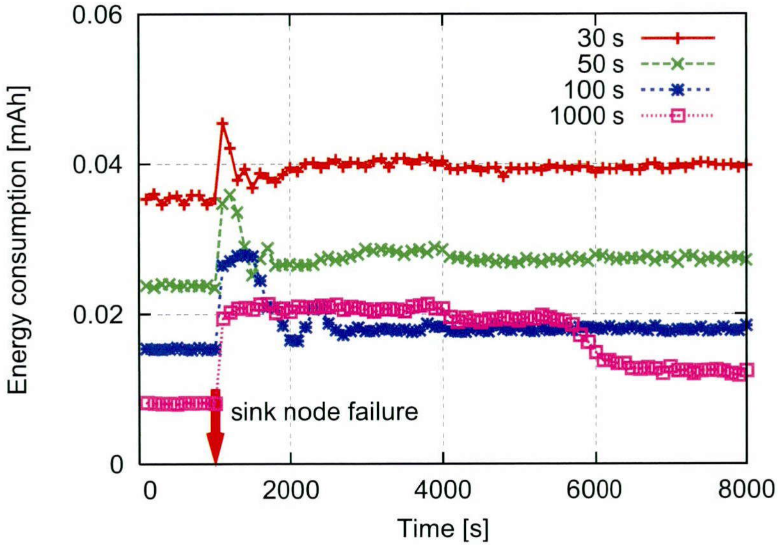
The accuracy of each node's routing table is highly significant in the case of sink-node failure. If a node selects the failed sink as a destination, a transmitted data packet wanders around the sink and cannot reach a sink node. As shown in Figure 3.14, the packet collection ratio decreases to less than 50% right after the sink failure, because about half of the sensor nodes send data destined for the failed sink. The packet delivery ratio rapidly recovers with shorter T_i, but it does not recover completely because the traffic load of the unfailed sink node gets approximately double. Note that the energy consumption is also recovers after the failure due to the accurate route information. R_s values when T_i is 30 s, 50 s, 100 s, or 1000 s are listed in Table 3.5. Although the recovery speed is considerably shorter when T_i is 30 s compared when with the other results, its packet delivery ratio before sink-node failure is lowest due to the overhead of table exchanges.

3.5 Summary

In this chapter, we quantitatively define robustness and resilience in wireless sensor networks and evaluate them. We also discuss what brings in robustness and resilience and how improve them in the MAC layer and the routing layer. Through the computer simulation experiments, we verified



(a) Packet delivery ratio



(b) Energy consumption

Figure 3.14: Resilience in the routing protocol

that receiver-initiated MAC protocols are compatible with the soft-state mechanism and they are more robust than sender-initiated MAC protocols and we show that adaptive settings of duty cycles achieve good resilience in the MAC layer. As for the routing layer, we present leveraging alternative and detour paths bears robustness against random node failures. Monitoring network conditions and highly-frequent exchanges of the monitored information yield great resilience. Especially, the robustness and resilience in the routing layer may be able to expect the energy-saving effect. Our study supports to design robust and resilient wireless sensor networks.

Chapter 4

A Controlled Self-Organization based Routing Protocol for Large-Scale Wireless Sensor Networks

4.1 Scalable Routing Protocols

Studies of self-organized routing protocols indicate that principal benefits include high scalability and good adaptivity to environmental dynamics [68]. There are also many studies on multi-sink sensor networks [33,69–76], but unfortunately, most are related to multi-sink network optimization where a centralized server is assumed. In such optimizations, the objective function is designed to maximize the time until the first node depletes its energy, obtaining the optimal flow and transmission power [69], the optimal destination sink node [70], or the optimal sink-node positions [71,72]. These optimizations can obtain optimal solutions, but computational costs rapidly increase as the number of nodes rises. Additionally, recalculations are needed whenever the network topology changes due to addition or failure of nodes, and with changing wireless channel conditions.

Self-organized routing is, in essence, local selection of the next-hop node. Such routing protocols differ from next-hop selection metrics to deliver data to the destination. All metrics are mainly based on the number of hop counts or the geographical distance to a destination node. The former is called potential- or gradient-based routing, and the latter is called virtual coordinate-based routing.

Various studies related to potential-based routing have been conducted [20–32]. Such efforts can be classified into the following two types:

1. Physics-knowledge based schemes [20–26]

2. Hop-count based schemes [27–32]

Physics-knowledge based schemes do not directly exploit hop counts from sink nodes when calculating potentials; the focus has been on analogies between sensor networks and models from physics such as electrical circuits [20], electrostatic fields [21], and gravitational attraction [22]. Other studies [23–26] directly apply potential theory to sensor networks. In these studies, except for References [20, 22], potentials are assigned at sensor nodes by solving a Poisson's or Laplace's equation. A potential field is constructed by using the solution from the equation, and all relay nodes forward data along the gradient of the potential field. Nodes require certain information to solve the equation in [21, 23] and to construct a gravitational field in [22]. Obtaining and exploiting location information assumes the availability of GPS receivers or some other means, however, significantly increasing the cost of producing such nodes [77]. By extension, another scalability problem is that the economic cost for deploying sensor nodes also rises, which is a potentially critical problem when constructing a large-scale network. A related difficulty is that it might not be feasible to use GPS receivers indoors, in underground rooms, within heavily forested areas, or at other locations with limited or obstructed satellite coverage.

In hop-count-based schemes, nodes calculate their own potential essentially from their hop count to sink nodes [27–32]. In other words, these routing protocols are a combination of minimum-hop routing and some metrics such as residual energy. In References [27, 28, 32], nodes also use their own remaining energy and that of neighbor nodes for load balancing. The authors of [30, 31] proposed an effective data aggregation mechanism supported by potential-based routing where local queue-length information is used to calculate potentials. Kumar et al. exploit potential-based routing for prolonging connectivity of the network in [29]. Although the proposed routing schemes exhibit good performance, location information is required in the schemes of Reference [32]. Also, the parameters used to calculate potentials were insufficiently examined and evaluated in References [27, 29], so the difference between those proposals and simple minimum-hop routing with remaining energy information is not clear. Most importantly, the above-mentioned studies offer no mechanism for guaranteeing intended network operation.

In virtual coordinate-based routing protocols, each node calculates its relative position to a small number of anchor nodes that know geographical location information through local interaction, and existing geographic routing techniques are applied. Scalable routing protocols without geographical location information are discussed in [78–82]. VRR proposed in [78] and VCP proposed in [79] make a virtual ring and a virtual cord in the whole network by assigning a location-independent identifier to all nodes. Meanwhile, protocols proposed in [80–82] assign all nodes a virtual coordinate on the pseudo-Euclidean space formed based on location. The advantage of these routing protocols is point-to-point communication between any two nodes, which is preferable for applications that expect point-to-point communication. However, some wrinkles are pointed out such as the void area problem known in geographic routings, which increases the computational complexity of a node to circumnavigate the void area. Moreover, it is necessary to know the virtual coordinates of the destination node in advance, requiring additional mechanisms. While these are important and interesting studies, they do not consider route optimization. In the following section, we introduce a potential-based routing for realizing CPBR.

4.2 Potential-Based Routing

In this section, we present how to construct a potential field and how to routing using the gradient of the field. CPBR utilizes a physics-knowledge-based scheme inspired by thermal diffusion. CPBR does not require location information in common with the methods proposed in [20, 24–26], which construct a potential field in a distributed manner. We focus on an analogy between conduction from a heat source and potential conveyance from a sink node. In CPBR, sensor nodes change their own potential according to the potential of the sink nodes. Using the diffusion equation that describes heat conduction, CPBR allows diffusion of sink-node potentials set by the control node throughout the entire sensor network.

4.2.1 Potential Field Construction with the Diffusion Equation

The diffusion equation is shown by the partial differential equation (4.1), which provides the magnitude ϕ of the diffusing quantity at time t and position $\mathbf{x}$.

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = D \bigtriangleup \phi(\mathbf{x},t), \tag{4.1}$$

where D is the diffusion rate and takes a positive value. By discretizing this equation and regarding ϕ as a potential, it becomes possible to construct a potential field based on self-organization where the behavior is governed by only local information.

Discrete Diffusion Equation

Node n calculates its own potential at time step $t+1$ (denoted by $\phi(n,t+1)$), based on the discrete diffusion equation (4.2). In equation (4.2), $Z(n)$ denotes a set of nodes neighboring node n. As can be noted from the equation, location $\mathbf{x}$ is cleared and the potential of node n is obtained from the latest potentials of $Z(n)$ and its own last potential. At this point, to calculate potentials, nodes must periodically inform neighbor nodes of their own potentials.

$$\phi(n,t+1) = \phi(n,t) + D(n) \sum_{k \in Z(n)} \{\phi(k,t) - \phi(n,t)\}. \tag{4.2}$$

In the discrete equation (4.2) (derived from the continuous equation (4.1)), $D(n)$ can be considered as a parameter that changes the magnitude of influence by neighbor node potentials. It is important to note that potentials may oscillate when $D(n)$ is large. To solve this problem, we consider the case where node n has only a single neighbor node m. Equation (4.2) can thus be replaced by $\phi(n,t+1) = D(n)\phi(m,t) + (1 - D(n))\phi(n,t)$, which represents an internal/external division of the points on the number line. In the following, we consider the case of $\phi(n,t) < \phi(m,t)$.

In the case where $0 < D(n) < 1$:

After node n receives the potential of node m, the following inequality is satisfied: $\phi(n,t) < \phi(n,t+1) < \phi(m,t)$. Repeating this procedure, the potentials of node n and node m approach and converge between $\phi(n,t)$ and $\phi(m,t)$. In this case, node n's potential remains smaller than node m's potential.

In the case where $1 \leq D(n) < 2$:

After node n receives the potential of node m, the following inequality is satisfied: $\phi(m,t) < \phi(n,t+1) < 2\phi(m,t) - \phi(n,t)$. Repeating this procedure, the potentials of node n and node m approach and converge, but the relationship between the magnitude of node n's potential and node m's potential is indefinite.

In the case where $2 \leq D(n)$:

After node n receives the potential of node m, the following inequality is satisfied: $2\phi(m,t) - \phi(n,t) \leq \phi(n,t+1)$. Repeating this procedure, the potentials of node n and node m remain unchanged or diverge. Moreover, the magnitude relationship between node n's potential and node m's potential is indefinite.

For the diffusion of potentials, it is preferable that $D(n)$ satisfies the following expression: $0 < D(n) < 1$. In the general case (i.e., when there exist multiple neighbor nodes), we set $D(n)$ to $\frac{\alpha}{|Z(n)|}$, where $|Z(n)|$ is the number of elements in $Z(n)$ and α is a constant. As a result, it can be considered that each node has been influenced by the potential of essentially only one node. We then set α to a value between 0 and 1 to keep the potential from oscillating.

Boundary Conditions

As an initial condition, all sensor-node potentials are set to zero. To construct a potential field from equation (4.2), we utilize a Dirichlet boundary condition to specify the sink-node potentials:

$$\phi(d,t) = \psi \quad (d \in N_s), \tag{4.3}$$

where N_s is a set of sink nodes and $\phi(d,t)$ is the potential of sink node d at time step t. ψ (≤ 0) is the constant value of sink-node potential. By the nature of the diffusion equation, this boundary condition is insufficient because the potentials of all nodes will arrive at much the same value as the potential of the sink node. We thus define another boundary condition that must be satisfied by nodes at the edge of the network:

$$\phi(e,t) = 0 \quad (e \in N_{edge}), \tag{4.4}$$

where N_{edge} is the set of nodes at the edge of the network, and node e is an element of N_{edge} that satisfies any of the following conditions (4.5) or (4.6):

$$H(e) > H(k) \quad (k \in Z(e)), \tag{4.5}$$

$$(H(e) \geq H(k)) \wedge (D_{id}(e) == D_{id}(k)) \quad (k \in Z(e)). \tag{4.6}$$

Here, $H(e)$ is the minimum hop count of node e from a nearby sink node, and $D_{id}(e)$ is the ID of the sink node.

Nodes append H and D_{id} to their potential, which is transmitted periodically to let their neighbor know their potential. Sink nodes set their H to zero and D_{id} to their own ID. When node n receives an potential from node m, node n updates $H(n)$ and $D_{id}(n)$. If $H(m) + 1$ is smaller than $H(n)$, node n sets $H(n)$ to the value of $(H(m) + 1)$ and $D_{id}(n)$ to the value of $D_{id}(m)$. When $H(m) + 1$ equals $H(n)$, node n changes $D_{id}(n)$ to the value of $D_{id}(m)$ with a probability of 0.5.

The condition (4.5) cannot define the potentials of network-edge nodes when two or more nodes with the same hop count exist. For that case, we use condition (4.6). Using D_{id} prevents nodes in the middle portion between two sink nodes from mistakenly deciding that they are at the edge of the network, because D_{id} does not coincide among neighboring nodes there. Eventually, a cardinal potential field is obtained, as shown in Figure 4.1.

Local Optimization

In this section, we present the construction of a potential field where nodes can locally select the best next hop. To do this, we add a term ρ on the right side of the discrete diffusion equation (4.2).

$$\phi(n,t+1) = \phi(n,t) + D(n) \sum_{k \in Z(n)} \{\phi(k,t) - \phi(n,t)\} + \rho(n,t), \tag{4.7}$$

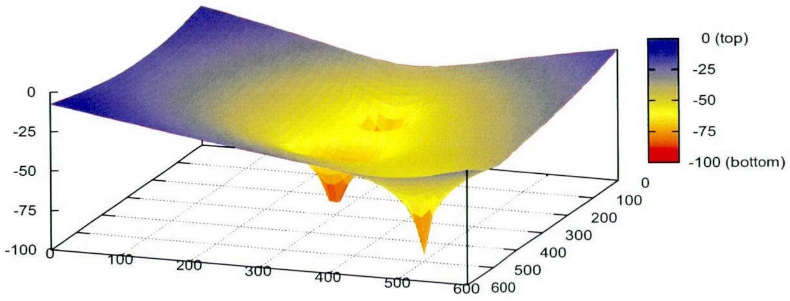


Figure 4.1: Potential field derived from the diffusion equation with 3 heat sources (3 sink nodes)

where $\rho(n, t)$ is a variable indicating the incremental influence of node n on the potential field at time step t (a larger ρ is associated with lower probability that node n is selected as a next hop, and vice versa). Here, we show load balancing based on remaining energy with $\rho(n, t)$.

Node n increases $\rho(n, t)$ when the remaining energy of node n is smaller than the average of that of neighbor nodes whose hop count equals $H(n)$. We assume that remaining energy is informed along with a periodical transmission of potentials.

The algorithm for deciding $\rho(n, t)$ is as follows, and is executed each time a potential is received:

1. Node n extracts the average remaining energy of neighbor nodes that have the same hop count as node n at time step t (denoted by $E_{avg}(n, t)$), and compares $E_{avg}(n, t)$ with own remaining energy at time step t (denoted by $E_{rem}(n, t)$).

   - If $E_{rem}(n, t) \geq E_{avg}(n, t)$, $\rho(n, t)$ is set to zero.
   - If $E_{rem}(n, t) < E_{avg}(n, t)$, it proceeds to step (2).

2. $E_{dif}(n, t)$ is the difference of energy between node n and its neighbors at time step t, and is assigned to $E_{avg}(n, t) - E_{rem}(n, t)$.

   - If $E_{dif}(n, t) < E_{dif}(n, t-1)$, $\rho(n, t)$ is unchanged.
   - If $E_{dif}(n, t) \geq E_{dif}(n, t-1)$, $\delta(n, t)$ is added to $\rho(n, t)$, where $\delta(n, t)$ is the difference between $\phi(n, t)$ and the average potential of the neighbor nodes whose hop count is the

Algorithm 1 *calculateRHO*(n, t); calculate local increase factor of node n's potential at time t

```
Initial setting: $\rho(n, 0) \Leftarrow 0$
$\phi_{avg}(n, t) \Leftarrow 0$
$E_{avg}(n, t) \Leftarrow 0$
$\delta(n, t) \Leftarrow 0$
$Z_s(n) \Leftarrow$ a set of neighbor nodes with the same hop count of node $n$
for all $k$ such that $k \in Z_s(n)$ do
  $\phi_{avg}(n, t) \Leftarrow \phi_{avg}(n, t) + \frac{\phi(k,t)}{Z_s(n)}$
  $E(k) \Leftarrow$ remaining energy of node $k$
  $E_{avg}(n, t) \Leftarrow E_{avg}(n, t) + \frac{E(k)}{Z_s(n)}$
end for
$E_{dif}(n, t) \Leftarrow E_{avg}(n, t) - E_{rem}(n, t)$
if $\phi_{avg}(n, t) > \phi(n, t)$ then
  $\delta(n, t) \Leftarrow \phi_{avg}(n, t) - \phi(n, t)$
end if
if $E_{avg}(n, t) > E_{rem}(n, t)$ then
  if $E_{dif}(n, t - 1) < E_{dif}$ then
    $\rho(n, t) \Leftarrow \rho(n, t - 1) + \delta(n, t)$
  end if
end if
$\rho(n, t) \Leftarrow \frac{1}{|Z(n)|}\rho(n, t)$
```

same as node n at time step t. However, in case that average potential is not larger than $\phi(n, t)$, $\delta(n, t)$ is set to zero.

3. Finally, $\rho(n, t)$ is set to $\frac{\rho(n,t)}{|Z(n)|}$.

Procedure (3) is to suppress dependence of the number of neighbor nodes on potential, thus reducing dependency on the network density in our routing. At last, we present the pseudo code of this local load-balancing mechanism in Algorithm 1.

Local Minima Avoidance and Loop-Free Mechanism

Since a diffusion equation solution converges to a harmonic function, neither the local maxima nor the local minima are taken inside a certain domain. However, local optimization or topology changes may cause local minima and routing loops. Once data gets stuck in local minima, it permanently cannot reach any sink node. Routing loops may also occur due to the next-hop decision

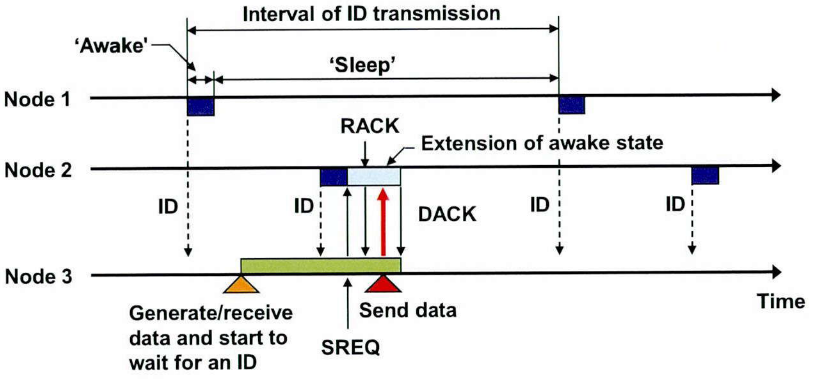


Figure 4.2: Data transmission procedure in MAC layer

along the gradient when the potential of nodes at the network edge increases to a maximum value. To prevent these defects, we change α (described in Section 4.2.1) to one and introduce hop-based routing. Since a potential is calculated with equation (4.7), if the potential of a node is the smallest among it and its neighbors, the node may be still a local minima at the next time step when α is smaller than one. Because each node can detect whether it is a local minima, it sets α to one when that occurs. Routing loops can be avoided by using hop-based routing when a potential is the same as a neighbor's, because each node knows the hop counts of its neighbors for the boundary condition.

MAC Layer Protocol and Potential Dissemination

Another major challenge in wireless sensor network research is energy efficiency. Energy efficiency in wireless sensor networks requires consideration of a duty-cycling MAC in which wireless nodes sleep and periodically wake up. Instead of taking multiple layers into consideration independently, considering them in combination is critical for system performance improvements. Thus, for the MAC layer protocol, we use an *intermittent receiver-driven data transmission (IRDT)* protocol, which aims to save energy and obtain high reliability as discussed in Chapter 2. Note that our routing protocol is not limited to IRDT—it is also applicable to other underlying protocols. In IRDT, each receiver sends its own identifier (ID) periodically to inform other nodes that it is ready to receive a data packet (Figure 4.2). A sender node waits for a receiver's ID, and when it acquires an ID from an appropriate receiver, it establishes a link by returning a send request (SREQ) message. After obtaining a request acknowledgement (RACK) for the SREQ, the sender transmits a data packet and terminates the communication upon receipt of a data acknowledgement (DACK). Sender node can thus communicate with one or more receivers flexibly, which can improve communication reliability and save considerable energy. IRDT is furthermore scalable, because it is an asynchronous MAC protocol that does not require synchronization.

Transmitting the potential with the periodical ID transmission, which is a simple modification that produces little overhead, allows IRDT nodes to inform neighbor nodes of their potential. Note that because IRDT uses a duty-cycling mechanism where each node periodically cycles between awake and sleep states, transmitted potentials are not necessarily received by nodes within the range of the communication. Therefore, each node wakes up and waits to receive potentials for a period of T_p at intervals of T_i. We refer to this period as the "sampling period", and to this interval as the "sampling interval". In IRDT, the sampling period should be longer than the interval of the periodic ID transmission to ensure that nodes know neighbor potentials.

Neighbor node potentials are managed in a soft-state manner. In other words, if a node receives a potential from a neighbor node during a sampling period, the node stores the potential; otherwise, the node deletes its information about the neighbor node. The procedure for calculating a potential is shown below.

During a sampling period T_p:

1. If a node receives a potential, it returns its own potential. After returning its potential, it calculates its own potential according to equation (4.2) or equation (4.7).
2. A node that receives, and that was intended to receive, the potential returned in step (1) also calculates its own potential.

While waiting for an ID for data transmission:

1. If a node receives an ID, it returns an SREQ containing its own potential. After returning the SREQ, it calculates its own potential.

Algorithm 2 Calculate potential of node n at time $t+1$

```
**Calculation is done after a sampling period or after receiving a potential**
**Initial setting:** $\phi(n,0) \Leftarrow 0$
$average \Leftarrow 0$
$D(n) \Leftarrow |Z(n)|$
**for all** $k$ such that $k \in Z(n)$ **do**
    $average \Leftarrow average + \frac{\phi(k,t)}{D(n)}$
**end for**
$\phi(n,t+1) \Leftarrow (1-\alpha) \cdot \phi(n,t) + \alpha \cdot average$
**if** local load balancing is used **then**
    $calculateRHO(n,t)$
    $\phi(n,t+1) \Leftarrow \phi(n,t+1) + \rho(n,t)$
**end if**
**if** $n \in N_{edge}$ **then**
    $\phi(n,t+1) \Leftarrow 0$
**end if**
```

2. A node that receives, and that was intended to receive, the potential returned in step (1) also calculates its own potential.

Immediately after a sampling period T_p:

1. Potentials of nodes whose potentials have not been updated for a period of T_i are deleted. After this process, the node calculates its own potential.

Finally, we show overall algorithm of potential calculation in pseudo code in Algorithm 2.

4.2.2 Routing in Potential Field

Consideration of both routing protocols and MAC protocols (duty-cycling MAC protocols in particular) is important for energy efficiency. Existing potential-based routing schemes use *only one receiver*, the one with minimal potential. Thus, because much time is expended and most of the energy is consumed while sender nodes wait for receivers to awaken in duty-cycling MAC protocols, simply combining these protocols offers no advantages. For increasing energy efficiency and reliability, we allow our potential-based routing to have multiple next-hop candidates.

In IRDT, a node that has data to send waits for an ID from an appropriate node. When the node receives an appropriate ID, it forwards the data to the sender. In our potential-based routing,

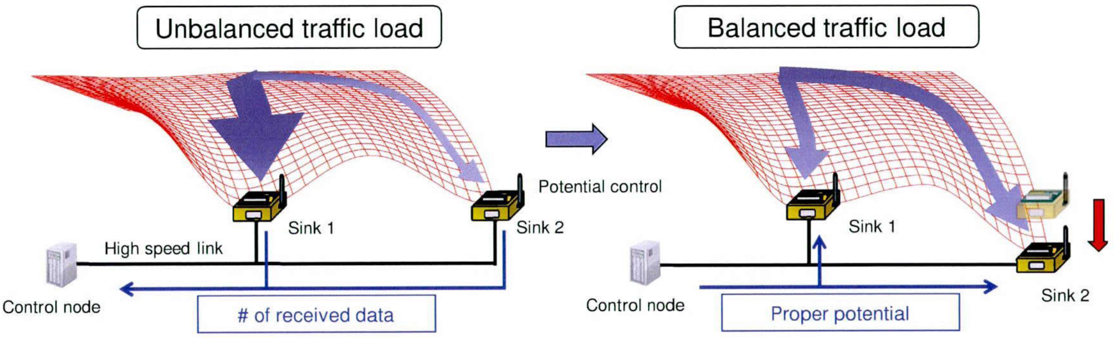


Figure 4.3: Potential control for balancing traffic flow traveling toward two sink nodes

a potential is transmitted along with an ID. A sender waits for an ID, and when it receives one, it decides whether to forward the data. To improve energy efficiency and reliability, when node n receives an ID from node m whose potential is not greater than its own potential, node n always returns an SREQ to node m. This also could be a great advantage for load balancing, which is shown in our other work [59].

4.3 Controlled Potential-Based Routing

In CPBR, multiple sink nodes report network information to a control node, and the control node decides sink-node potentials for constructing a desired potential field. Sink nodes report at regular time intervals T_m for the purpose of control. We call this information the "metric value" (denoted by m). We next show that it is possible to control the rough direction of traffic flow by controlling sink-node potentials. Figure 4.3 shows an obvious example of CPBR, where traffic flow moving toward two sink nodes is balanced. CPBR can control not only near-equalization of traffic, but also control the ratio of the number of data packets received by each sink node. We aim here at balancing traffic and energy consumption among sink nodes.

- **Traffic balancing of sink nodes**

  For traffic balancing, sink nodes control their own potential to maintain a uniform number of received data packets. The metric value here is the number of data packets received by sink node d at time t (denoted by $A(d,t)$).

- **Energy-density balancing of sink nodes**

  We define energy density $P_{ed}(d,t)$ as the sum of the remaining energy of neighbor nodes around sink node d at time t. Nodes in the network, particularly those neighboring sink nodes, frequently relay data and thus consume more energy. Thus, the metric value based on energy density can maximize the duration over which the energy density of all sink nodes runs down. To do so, sink nodes control their potential to equalize the average remaining energy of nodes neighboring the sink nodes. The metric value, $P(d,t)$, is defined as following:

$$P(d,t) = \frac{P_{ed}(d,t)}{|Z(d)|}.$$

The potential of sink node d at time t, $\phi(d,t)$, is given by the potential control function $\Phi(d,t)$ instead of ψ. $\Phi(d,t)$ is decided according to the following algorithm:

1. The control node set sink nodes' potentials to the initial value Φ_{init}:

$$\Phi(d,0) = \Phi_{init} \;\; (\Phi_{init} < 0). \tag{4.8}$$

2. The control node calculates $\overline{m(t)}$, the average of the metric value. For example, $\overline{m(t)}$ for $A(d,t)$ (denoted by $\overline{m_A(t)}$ for convenience) is defined by equation (4.9) and $\overline{m(t)}$ for $P(d,t)$ (denoted by $\overline{m_P(t)}$) is defined by equation (4.10):

$$\overline{m_A(t)} = \frac{\sum_{d\in N_s} A(d,t)}{\sum_{d\in N_s} 1}, \tag{4.9}$$

$$\overline{m_P(t)} = \frac{\sum_{d\in N_s} P_{ed}(d,t)}{\sum_{d\in N_s} |Z(d)|}. \tag{4.10}$$

3. The potential of sink node d is given according to expression (4.11):

$$\Phi(d,t+1) = \Phi(d,t) * \Big(1 - \sigma\frac{m(d,t) - \overline{m(t)}}{\overline{m(t)}}\Big), \tag{4.11}$$

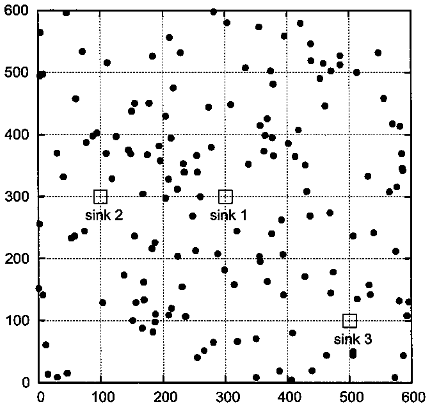


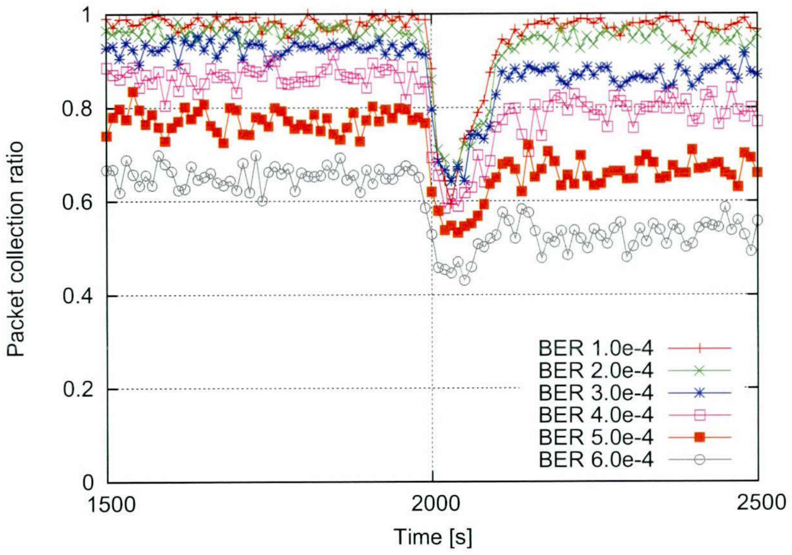
Figure 4.4: An example of network model in which 150 sensor nodes and 3 sink nodes are deployed over a 600 m × 600 m square field

where σ is a constant $(-1 < \sigma < 1)$. The change in potential can be larger when it is away from the mean value. Conversely, the change can be smaller when it is closer to the mean value. To avoid aberrant potential values, the potential is taken to be within a range decided beforehand, $[\Phi_{min}, \Phi_{max}]$.

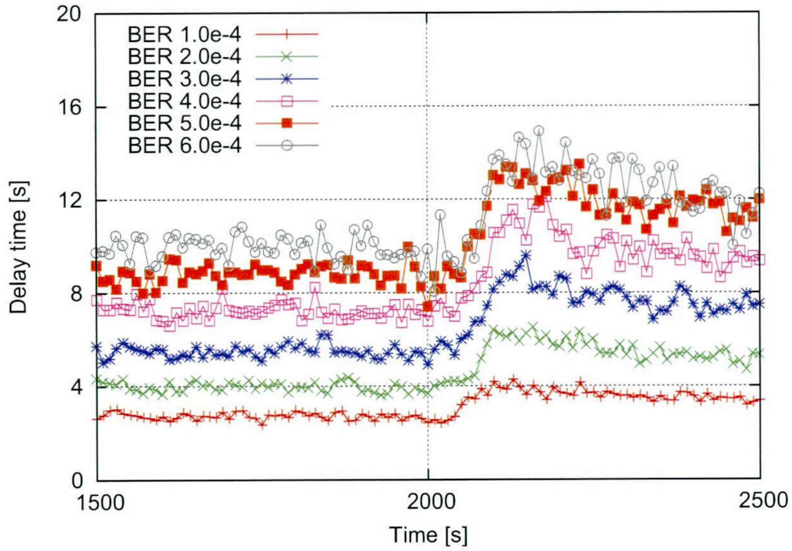
4.4 Simulation Results

We evaluate the impact of CPBR through computer simulation by using an event-driven packet-level simulator written in Visual C++ and all results are the average of 100 trials. The network model is a square (length of each side: 600 m) in which 150 sensor nodes are randomly deployed and 3 sink nodes (sink 1, sink 2, and sink 3) are set at points (300, 300), (100, 300), and (500,100), respectively. The communication range of each node is 100 m. We employ the disk model of communication between nodes, where the strength of the radio signals does not deteriorate, and a transmitted message is assumed to be received by nodes within the communication range unless message collisions occur. In addition, our evaluation is made on safe side; if a message collision occurs while a message is being received, the messages are simply discarded.

We assume that data packets are generated by each sensor node according to a Poisson process with intensity λ, and are sent to the sink nodes by multi-hop relay. The simulation commences



(a) Packet collection ratio



(b) Delay time

Figure 4.5: Robustness against bit error and resilience to sink-node failure

Table 4.1: Parameter settings for CPBR evaluation

Parameter	Value
Transmission speed	100 kbps
Communication range	100 m
λ	0.003 packet/s/node
Current consumption (TX)	20 mA
Current consumption (RX)	25 mA
Current consumption (SLEEP)	0 mA
T_i	100 s
T_m	500 s
Φ_{init}	-30
Φ_{min}	-100
Φ_{max}	0
α	0.9
σ	± 0.2
Message size (ID, SREQ)	24 byte
Message size (RACK, DACK)	22 byte
Packet size (Data)	128 byte

after an initialization phase in which each node sufficiently exchanges its potential with neighbor nodes. The interval of ID transmission is 1.0 s, and T_p is also set to 1.0 s. Table 4.1 shows other parameters. Note that σ decides the rate of potential increase and decrease, with a positive value of σ increasing Φ when m is greater than $\overline{m}$, and vice versa. Here, we set σ to 0.2 for $\overline{m_D(t)}$ and to -0.2 for $\overline{m_P(t)}$. For T_m, the interval of the potential control, a larger value than T_i is used: T_m is set to 500 s in order to wait for the convergence of self-organized potential calculation.

4.4.1 Robustness of Self-Organized Routing

We consider robustness of the self-organized process and robustness of the control process separately. First, we show the robustness of self-organized routing. Figure 4.5 shows the transient performance of the packet collection ratio and the average sensor-to-sink delay time. In this simulation, sink 1 fails at 2000 s and bit error rate (BER) is set to $1.0\times10^{-4} \sim 6.0\times10^{-4}$. For simplicity, we assume that bit error occurs with a probability corresponding to the product of packet or message size (bit) and BER (%/bit). When a node detects a bit error in a received message (ID, SREQ, RACK, DACK, or Data), it discards the message. Because the data packet size is 128 bytes, data packets are discarded approximately 10–60% of the time.

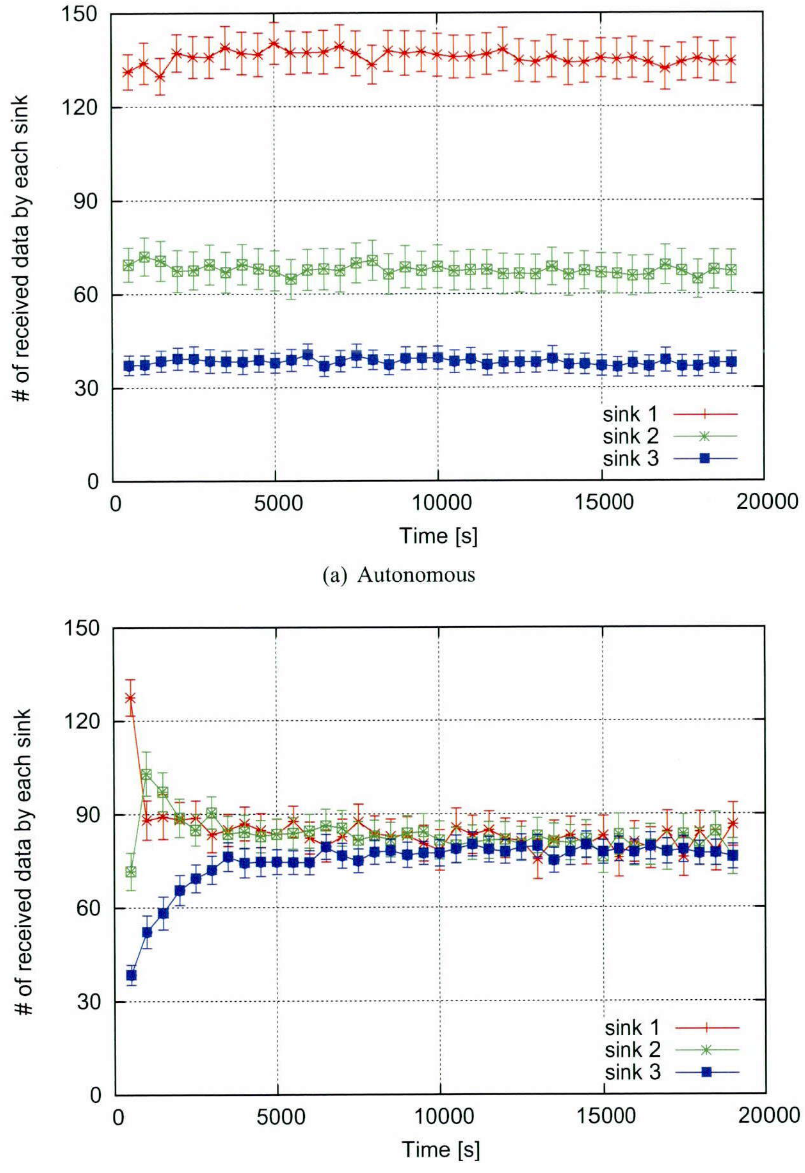
In Figure 4.5(a), immediately after a sink failure, the packet collection ratio decreases and then recovers, which indicates good resilience of the self-organized routing. When nodes select the failed sink as a destination, data packets wander around the sink node and cannot reach any other sink node, which causes the decrease of the packet collection ratio. After the sink node failure, neighbor nodes of the failed sink node clear the sink potential after sampling interval (T_i), and neighbor nodes of those nodes update the potential in sequence until eventually the potential field is reconstructed. Clearly, as T_i becomes shorter, the recovery time for the packet collection ratio, too, becomes shorter. As a result, recovery in the figure is within $2T_i$. Because loads of the existing two sink nodes are heavier than before the failure of sink 1, the packet collection ratio fails to completely recover. Even under high bit error rates, a comparatively high collection ratio can be attained thanks to the use of multiple receivers in our routing. Particularly, even when the probability of data bit error is 50%, the packet collection ratio reaches approximately 80%.

Figure 4.5(b) shows the average sensor-to-sink delay time before and after a sink failure. The delay time increases soon after the failure, and then decreases gradually with the reconstruction of the potential field. This increase is due to detours and loops in the proximity of the failed sink. As the decreased delay time continues after recovery of the packet collection ratio, detour routes are modified gradually.

The above simulation demonstrates the effectiveness of local decisions based on local interactions in bringing about robustness and resilience in self-organized routing protocols. However, it cannot tackle optimality of the whole network. Henceforth, we show the advantage of control from outside the system.

4.4.2 Traffic Balancing Management in CPBR

Potential control based on the amount of received data can balance the traffic load of the sink nodes. We now examine the effectiveness and adaptivity of CPBR to heterogeneous sensor node densities. Figure 4.6 shows the impact of potential control when sensor node densities differ. We



(a) Autonomous

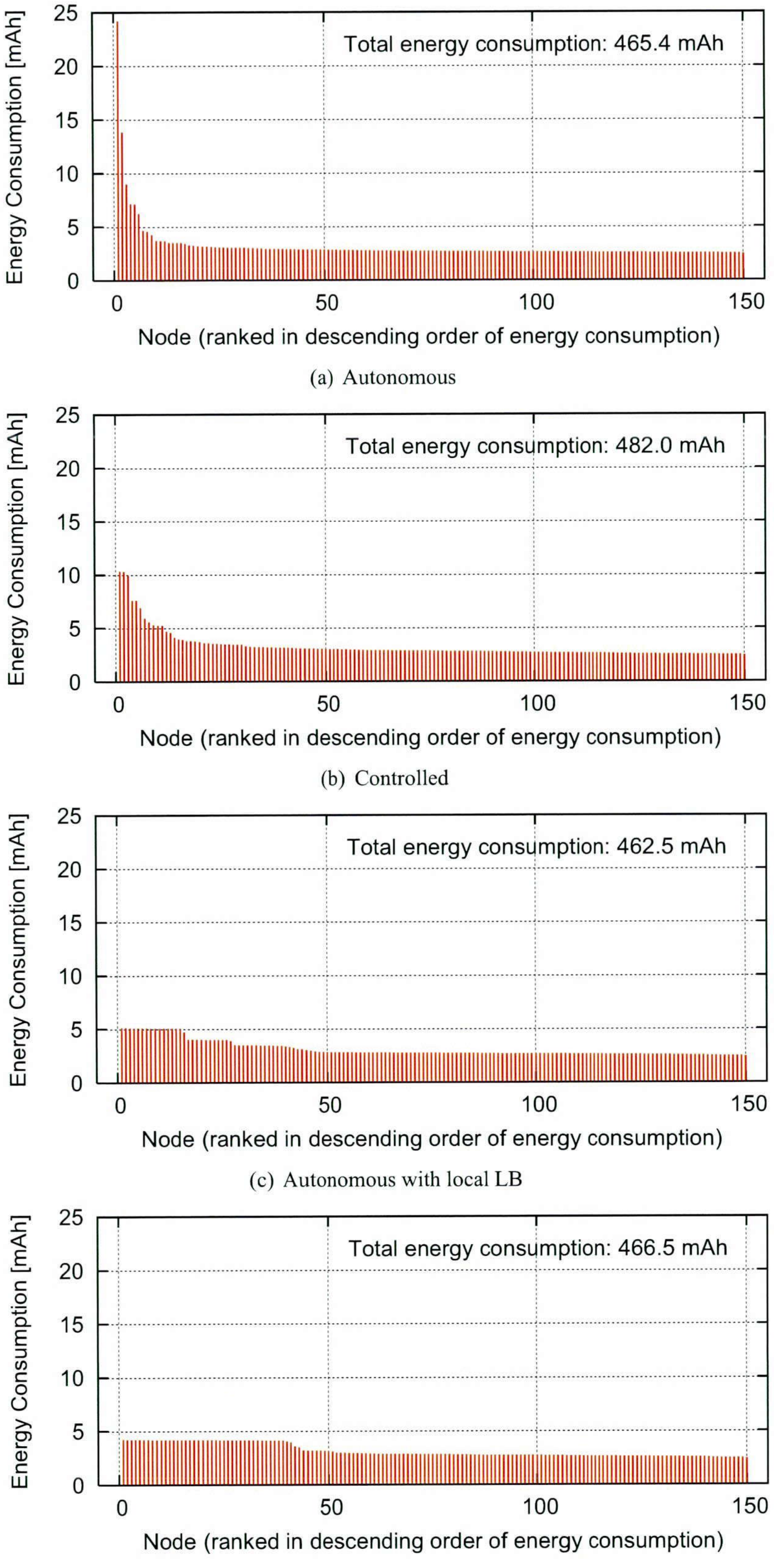
(b) Controlled

Figure 4.6: Potential control based on the number of received data packets (150 sensors and 3 sinks)

use the field shown in Figure 4.4 for the placement of the sink nodes. On the deployment of sensor nodes, the right half of the field has double the node density of the left half. In this result, the y-axis indicates the average number of data packets received by each sink node during T_m (500 s), with a 95% confidence interval. When the control node does not manage potentials, the number of data packets received by each sink node differs greatly, and remains mostly unchanged over time (Figure 4.6(a)). Such concentrations of traffic load are induced by the density difference of sensor nodes and a lopsided sink-node distribution that self-organized routing protocols cannot cope with. With sink node control, the number of received data packets (over 3 sink nodes) converges to a nearly identical value, equal to $\lambda T_m \frac{N_n}{N_s}$ $(= 75)$, where N_n is the number of sensor nodes. Convergence time is about 10000 s, which indicates that 20 controls causes the number of received data packets to converge. When considering the operating time of an actual sensor network system, which can be in units of years, we note that convergence within a realistic time is possible. CPBR can also attain good convergence of the number of received data packets in this case, indicating that our proposed potential control can adaptively accommodate heterogeneous densities of sensor nodes.

4.4.3 Energy-Density Balancing Management in CPBR

Balancing the energy density is clearly a practical application of CPBR for prolonging network lifetime, expected to be accomplished by potential control based on $P(d,t)$. Figure 4.7 is the results of a simulation using potential control based on $P(d,t)$. The network model is same as that of Figure 4.4. The y-axis of the figure indicates the energy consumption of each node in a 6-hour simulation, and the x-axis represents each node sorted in descending order. In our potential-based routing, relay load is concentrated on the node with the minimum potential among the neighbors of the sink node. Hence, once a potential field has been constructed, the relay load remains concentrated on a specific node (as is apparent in Figure 4.7(a)). In this figure, because the number of received data packets at sink 1 is largest, the energy consumption of the heaviest loaded node is larger. Figure 4.7(b) indicates that potential control can reduce the energy consumption of that node, because the number of neighbor nodes at each sink nodes is not different very much and



(a) Autonomous

(b) Controlled

(c) Autonomous with local LB

(d) Controlled with local LB

Figure 4.7: Potential control based on neighbor energy density (energy consumption distribution)

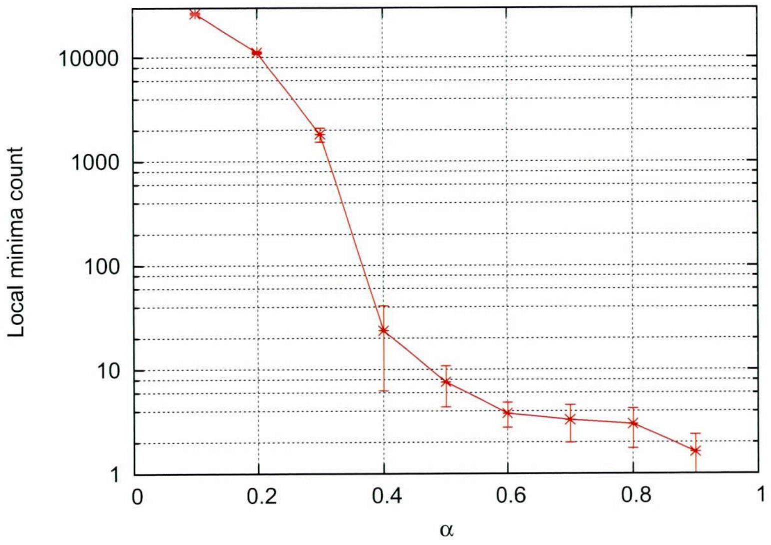


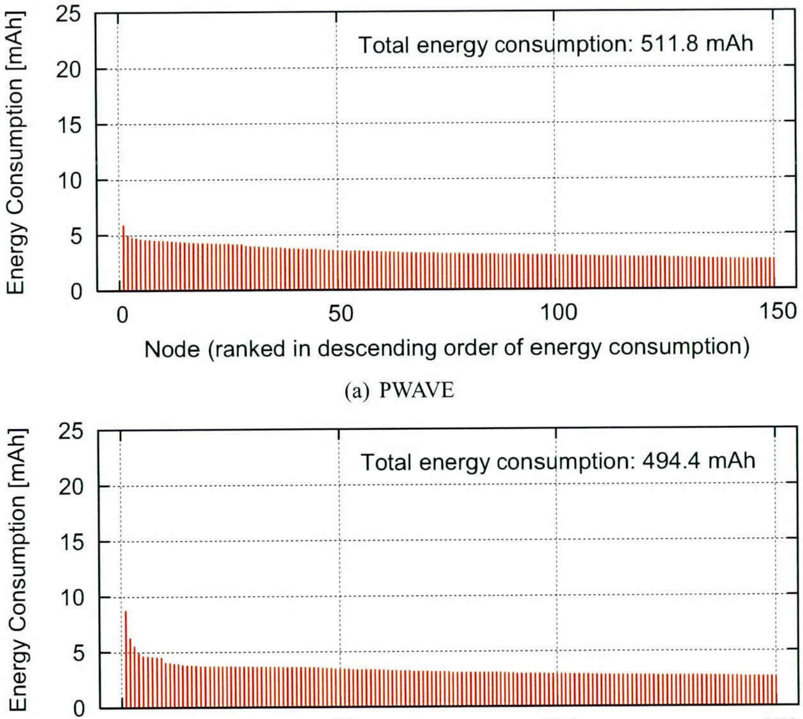
Figure 4.8: The number of local minima vs. α

therefore the number of data packets received by each sink node is also nearly equal. If the number of neighbor nodes differs considerably, the energy consumption bias may grow even more than the result shown in Figure 4.7(a). In any situation where there is a major difference among the number of neighbor nodes, the local load-balancing (local LB) mechanism described in Section 4.2.1 can substantially reduce the energy consumption of the node with the heaviest load, as shown in Figures 4.7(c) and 4.7(d). Figure 4.7(d) shows the results of CPBR with a local load-balancing mechanism. While the total energy consumption rises due to increased detours, a 82.6% reduction in the energy consumption of the heaviest loaded node was attained as compared with the results shown in Figure 4.7(a).

When the local load-balancing mechanism is used, local minima may be quite serious problem as we pointed in Section 4.2.1. We count the number of occurrence of local minima and show an appropriate decision of α can prevent the emergence of local minima as shown in Figure 4.8.

We examine the energy consumption of PWAVE and EBRP described in [20] and [28] respectively for comparing with our CPRB in Figure 4.9. PWAVE framework generates globally balanced traffic allocation and maximizes the network lifetime approximately. This is done by using iterative calculation of potentials just like equation (4.7) and stochastic determination of a next hop node. In EBRP, each node establishes a mixed virtual potential field in terms of depth (U_d), energy density (U_{ed}), and residual energy (U_e). The mixed potential field is linear sum of them, that is, $(1 - \alpha_{EBRP} - \beta_{EBRP})U_d + \alpha_{EBRP}U_{ed} + \beta_{EBRP}U_e$. Thus, EBRP carries packets toward sink nodes through the dense energy area to avoid nodes with relatively low remaining energy. We select a combination of the parameter pair $(\alpha_{EBRP}, \beta_{EBRP})$ to (0, 0.4), which produces the similar degree of the packet delivery ratio of CPBR and PWAVE. The comparative results among Figure 4.7(d), Figure 4.9(a), and Figure 4.9(b) show that our CPBR can reduce the most energy consumption of the node with the heaviest relay loads. This is because the local load-balancing mechanisms in PWAVE and EBRP cannot achieve more efficient load balancing than that in CPBR. Allowing CPBR to have multiple next-hop candidates can disperse relay loads effectively as presented in [59].

Figure 4.11 shows the network lifetime based on the number of alive nodes, and the network lifetime based on the reachability of sink nodes. There are various definitions for sensor network



(a) PWAVE

(b) EBRP

Figure 4.9: Comparison of energy consumption with PWAVE and EBRP

lifetime, depending on the application [83], but in this chapter, we use following two simple definitions:

1. The time until the first node depletes its energy (alive node).
2. The time until 20% of nodes lose reachability to sink nodes (80% reachability).
3. The time until All nodes have reachability to sink nodes (100% reachability).

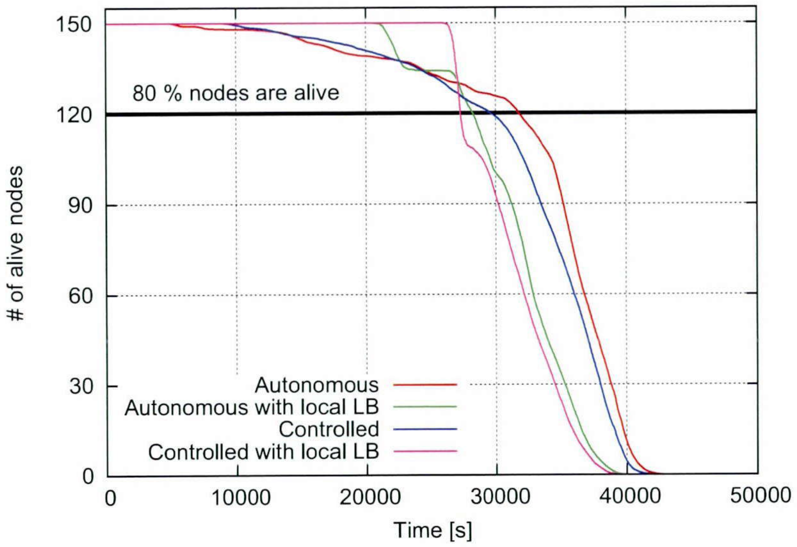
To evaluate network lifetime, we set the battery of sensor nodes to a comparatively small value (5.0 mAh) and simulation time was set to a time longer than the battery lifetime. Comparing "controlled with local LB" with "autonomous" in Figure 4.10, the time until the first node depletes its energy is more than fourfold, as noted above. In terms of the time guaranteeing 80% reachability, that of CPBR with load balancing ("controlled with local LB") is 14.7% shorter than the default potential-based routing (autonomous), because CPBR increases total energy consumption. However, the both time of CPBR with load balancing to ensure 100% reachability and to keep 100% nodes alive is 5.49 times longer than that in the default as shown in Figure 4.10(b).

Comparison results are shown in Figure 4.11. CPBR can achieve the best lifetime in terms of alive node and 100% reachability thanks to the global and local load balancing. Meanwhile, 80% reachability of EBRP is longer than other two results. The reason of this is that CPBR and PWAVE aim for global load balancing and the total number of hop count of them is larger than EBRP. However, 80% reachability of CPBR (autonomous) is longer than that of EBRP because CPBR has multiple next-hop candidates and reduces idle time in the MAC layer.

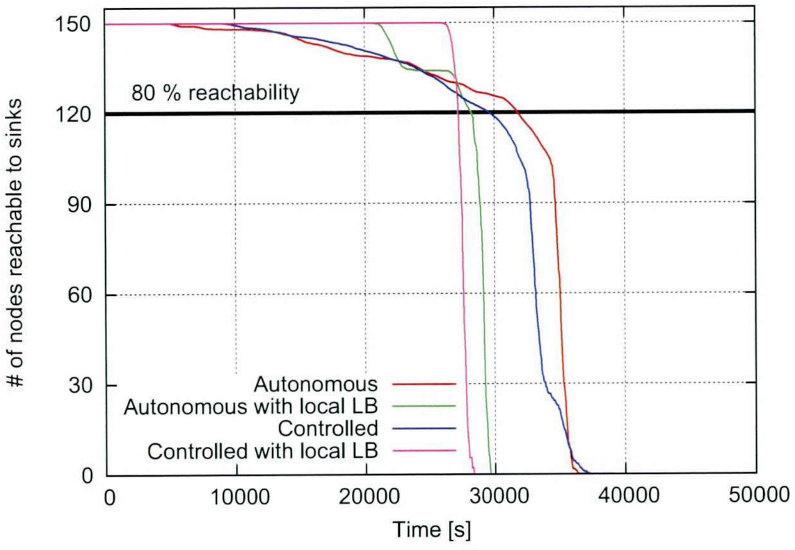
In the remainder of this section, we demonstrate the scalability and robustness of CPBR.

4.4.4 Scalability of CPBR

Figure 4.12 shows the number of received data packets in a network where 5,000 sensor nodes and 100 sink nodes are randomly deployed. The network field forms a square with side length 3500 m. In this case, potential control works properly without significant change in convergence time. However, some issues remain; one being that the deviation of Figure 4.12(b) is larger than that of Figure 4.6(b). This is an inevitable result of a self-organization mechanism when network

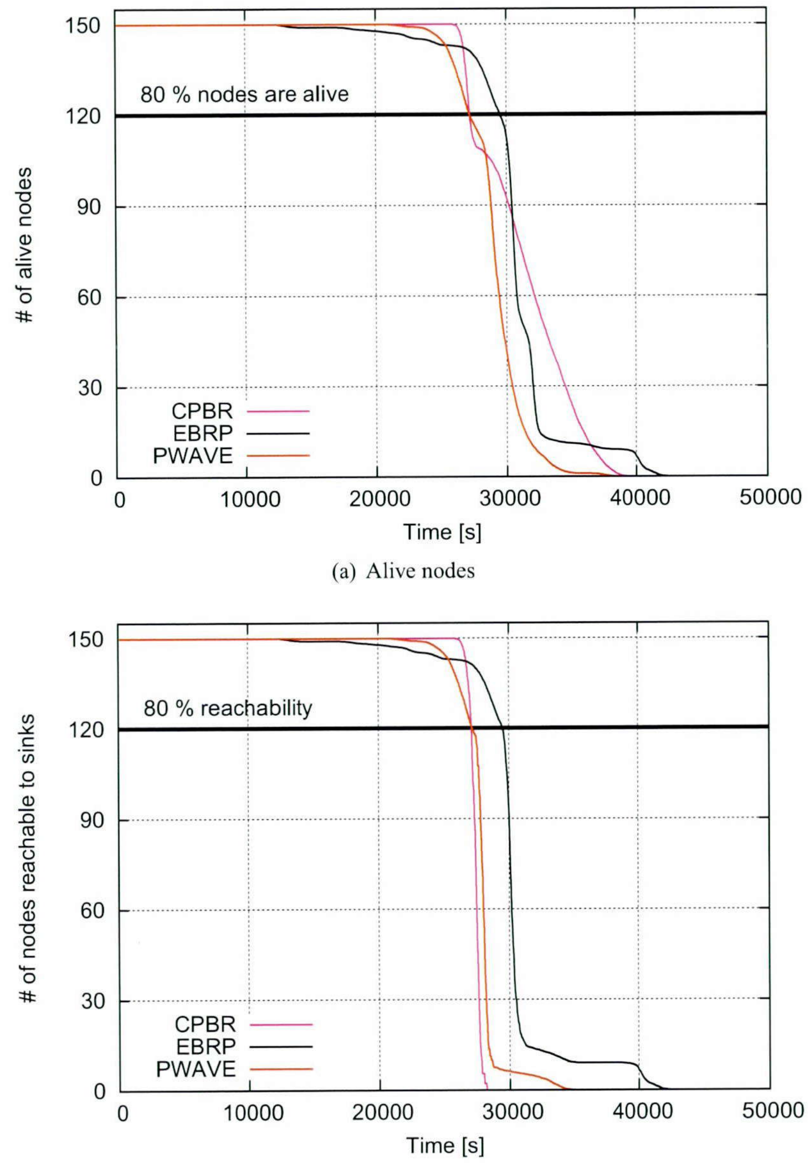


(a) Alive nodes



(b) Reachability to sink nodes

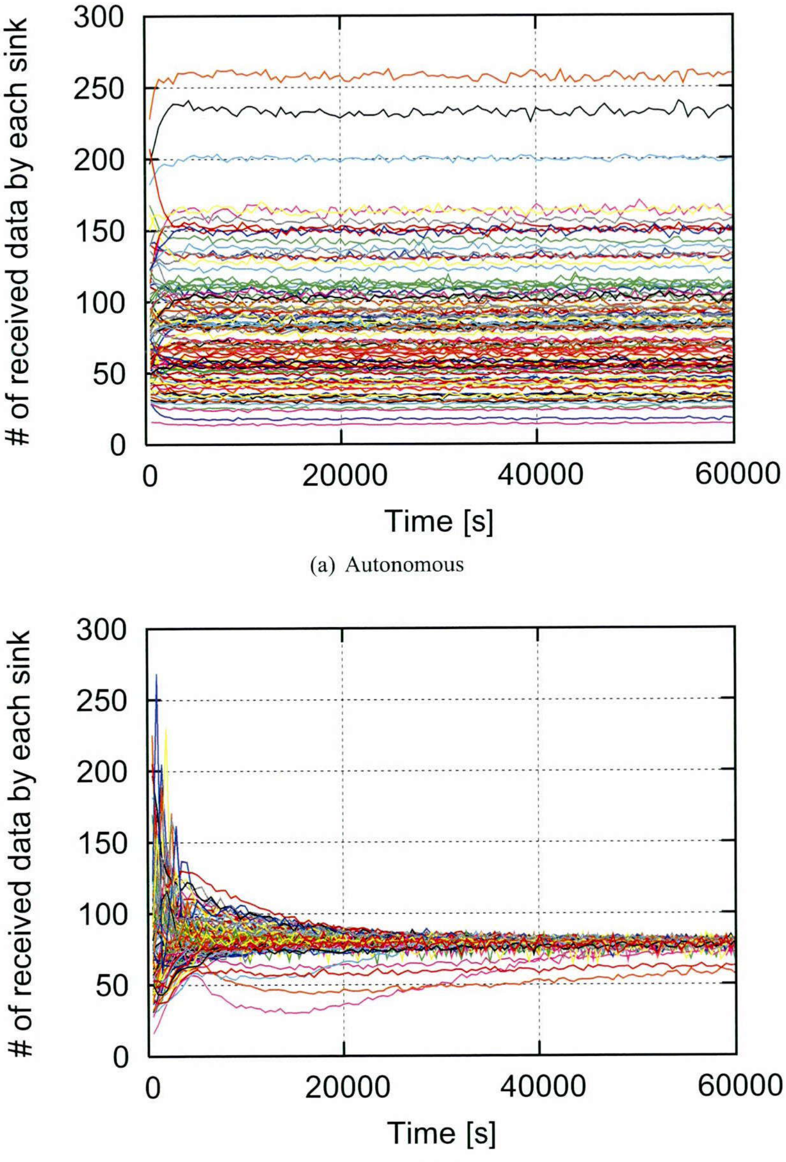
Figure 4.10: Potential control based on neighbor energy density (network lifetime)



(a) Alive nodes

(b) Reachability to sink nodes

Figure 4.11: Comparison of network lifetime with PWAVE and EBRP



(a) Autonomous

(b) Controlled

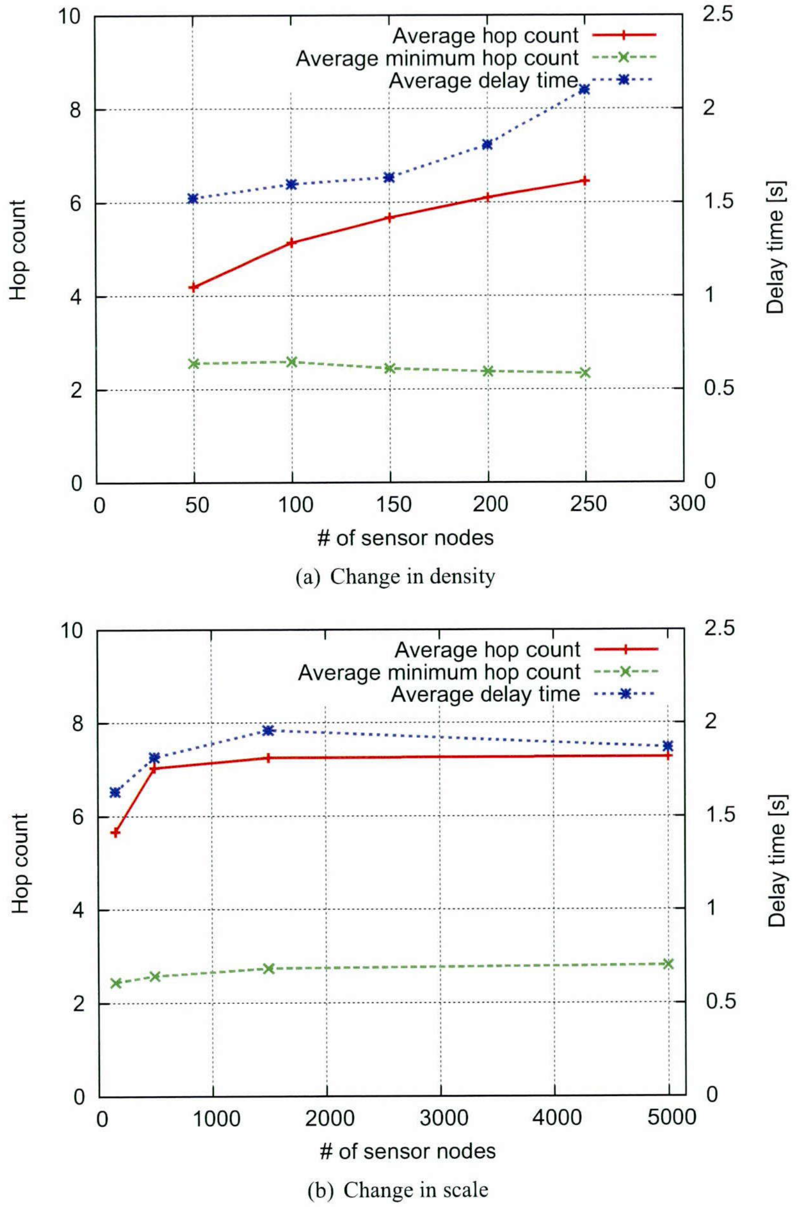
Figure 4.12: Potential control based on the number of received data packets (5000 sensors and 100 sinks)

scale increases. However, the main reason for this problem is the simple algorithm based on equation (4.11). Reinforcement through learning algorithms or evolutionary algorithms has potential to improve the convergence time. In this chapter, we aim at showing the effectiveness of control from the outside, and therefore do not target control efficiency.

In Figure 4.13, we investigate scalability of CPBR by evaluating the average hop count and the average delay time when network density and network scales change. For the evaluation in density change, network field and positions of sink nodes are the same, as shown in Figure 4.4, and the number of sensor nodes is increased. The area of the field is increased with a constant node density to evaluate performance when network scale changes. Here, the results of 150 nodes in Figures 4.13(a) and 4.13(b) are typical. As described in Section 4.2, we do not limit receivers to a single node in our potential-based routing. Therefore, the increase in the number of detour hops arises with the increase in network density. From Figure 4.13(a), in cases where node density increases 5 times, the increase of the average hop count is at most 2 hops, and the average delay time only increase by about 30%. Note that this is due not only to increasing detour paths, but also to congestion caused by the increase in traffic. As for the increase in network scale, if the number of nodes is larger than 500, there is little change in the average hop count or average delay time, as shown in Figure 4.13(b). The average hop count and the average delay time are smaller in the case of 150 nodes because the ratio of nodes existing at the network edge is larger. These nodes transmit data in a direction that certainly approaches sink nodes, due to the boundary condition. These results indicate that CPBR is scalable with regards to network density, average hop count, and average delay time. A remaining scalability problem is convergence time, but this chapter omits that discussion because the convergence time is much shorter as compared with the operation time of applications in sensor networks.

4.4.5 Robustness of CPBR

We also demonstrate the robustness of the control process of CPBR against sink node failures and additions. As shown in Figure 4.14, we randomly deployed 300 sensor nodes over a square network with side length 850 m, and placed 9 sink nodes at locations (142, 708), (142, 425), (142, 142), (425,



(a) Change in density

(b) Change in scale

Figure 4.13: Scalability of CPBR

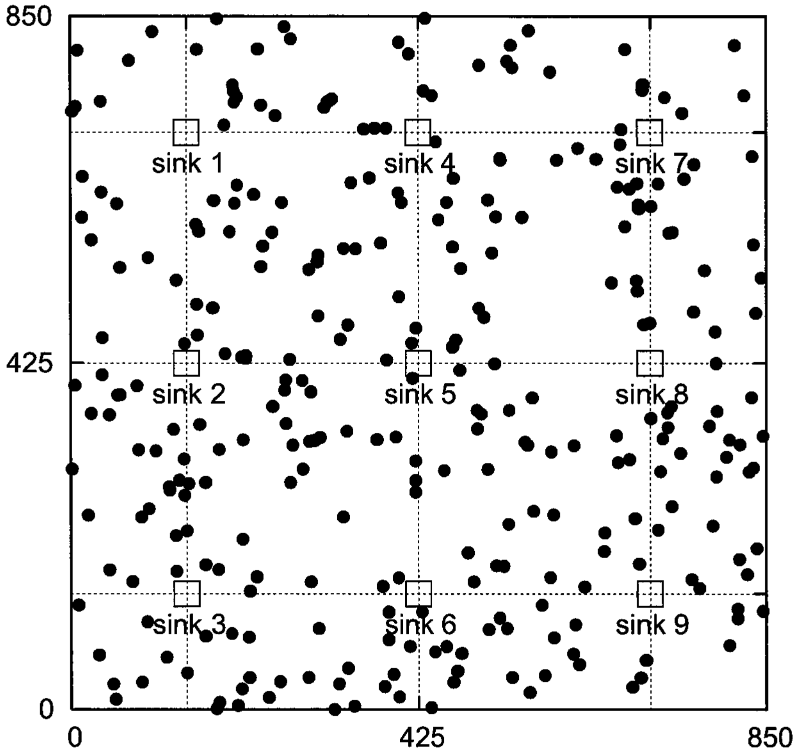


Figure 4.14: An example of network model in which 300 sensor nodes and 9 sink nodes are deployed over a 850 m × 850 m square field

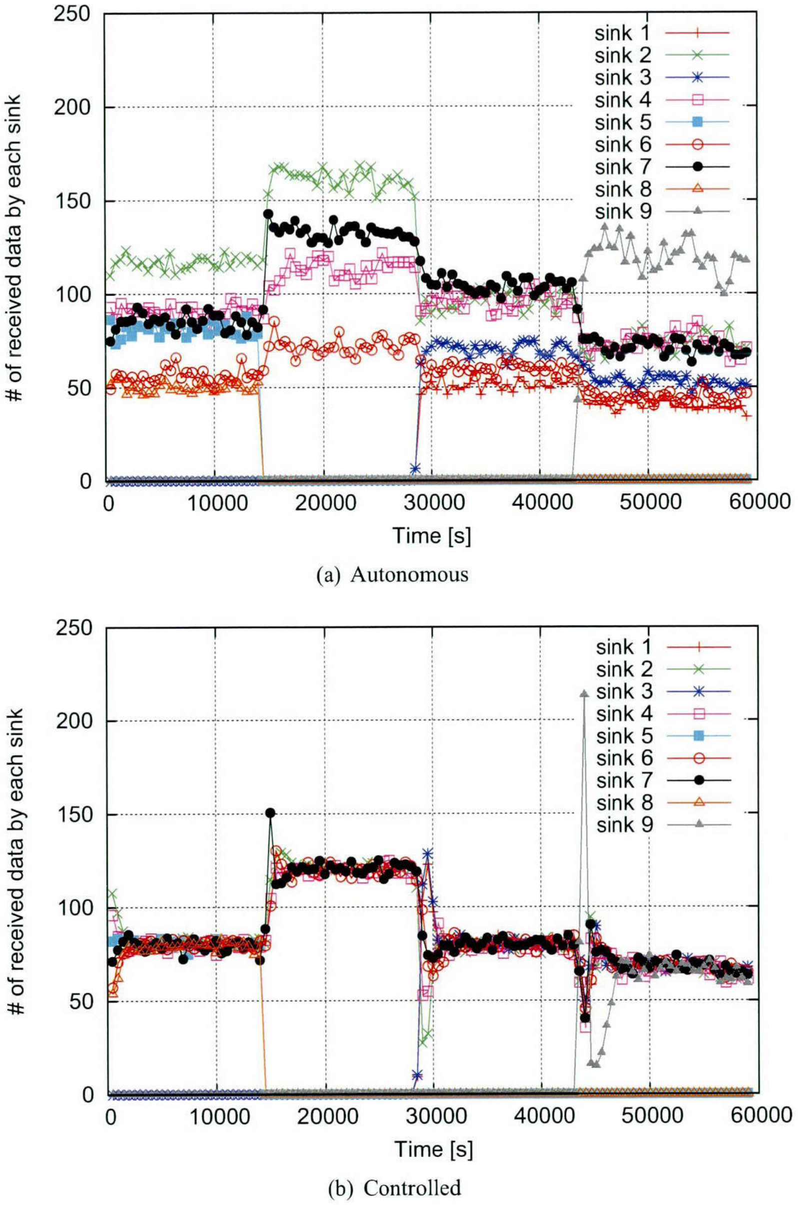
708), (425, 425), (425, 142), (708, 708), (708, 425), and (708, 142) (denoted by sink 1 to sink 9, respectively). The number of sink nodes was varied as follows:

- Six sink nodes (2, 4, 6, 7, 8, and 9) are active at 0 seconds.
- After four hours, two sink nodes (6 and 9) break down.
- Eight hours after that failure, two sink nodes (1 and 3) are added.
- At twelve hours into the simulation, a sink node (5) is added.

Figure 4.15 shows the results of the simulation. Comparing Figure 4.15(a) and Figure 4.15(b), we find that CPBR can control potentials adequately after failures and additions of sink nodes. In Figure 4.15(b), the number of data packets received by exiting sink nodes is equalized, and we expect that potential control based on the energy density works appropriately in the same way. CPBR is thus robust against sink node failures and additions.

4.5 Summary

In a controlled self-organization scheme intended to ensure desired network behavior, one or more controllers control a portion of self-organizing nodes through centralized control, distributed control, or some other control scheme. In this chapter, we proposed controlled potential-based routing (CPBR), which is based on a controlled self-organization scheme. In this scheme, sensor nodes calculate their own potential in a self-organized manner, while a control node manages sink-node potentials by centralized control so as to construct a desired potential field. The demonstrated CPBR operates over an IRDT protocol, but is not limited to IRDT; it is also applicable to sensor networks where other MAC protocols are adopted. Through computer simulation, we showed that load balancing of the sink nodes could be attained in diverse situations, with potential control based on the amount of data received at each sink node. We also showed that CPBR with potential control based on the energy density could extend the time until the first node depletes its energy by 449%. We also verified the robustness of the proposed method.



(a) Autonomous

(b) Controlled

Figure 4.15: Robustness of CPBR

Chapter 5

A Design Approach for Managed Self-Organization Control Focused on Control Timescale for Future Wireless Sensor Networks

5.1 Protocol Overview in Each Layer

In this section, we give an overview of controlled potential-based routing again, and especially we discuss the control timescale in the MAC layer, routing layer, and external control.

5.1.1 Sleep Control in the MAC Layer

One-hop communication is performed in the MAC layer, which takes several milliseconds in the most sensor network scenarios. Therefore, it is difficult to deal with perturbations that cause the topology changes with cycle of a few milliseconds or less. Moreover, in many MAC protocols in the sensor network, the sleep control is assumed, where power-saving operation is expected. For example, B-MAC [6], which is a widely known MAC protocol with the sleep control, allows nodes to sleep every tens of milliseconds to several seconds. Since each node can communicate with its neighbor nodes only when it is awake, the cycle of this sleep control means the minimum unit time of one-hop data transmissions. We use the intermittent receiver-driven data transmission protocol as a MAC protocol. As described in the previous chapter, this protocol is one of the receiver-driven or receiver-initiated MAC protocols where nodes periodically sleep and transmit a beacon to inform their neighbors that they are ready to receive data.

5.1.2 Route Management in the Routing Layer

CPBR is a kind of potential-based routing protocols, and it utilizes the proactive self-organized route management. In a potential-based routing, all nodes have a scalar value “potential”. This potential of a node is lower as the hop count from the nearby sink node is smaller. Therefore, a node only forwards data to the neighbor with lower potential than its own for delivering data toward a sink node.

In CPBR, a potential of node n at time t, denoted by $\phi(n,t)$ is given by equation (5.1) (see more details in Chapter 4). $Z(n)$ is a set of neighbors of node n and $|Z(n)|$ is the size of it. For the calculation of potentials, each node has to manage its neighbors’ potential. In order to do that, each node informs its potential to its neighbors periodically. When a node receives a neighbor’s potential, it registers the potential of the neighbor, and when it cannot receive any potential from a neighbor during a certain period, it clears memory of the neighbor’s potential received previously.

$$\phi(n,t+1)=\phi(n,t)+\frac{1}{|Z(n)|}\sum_{k\in Z(n)}\{\phi(k,t)-\phi(n,t)\}. \tag{5.1}$$

5.1.3 External Control for Self-Organization

In CPBR, a control node, which is able to communicate with all sink nodes, is responsible for observing and controlling potentials of all sink nodes. The control node controls potential of sink node d at time t, denoted by $\Phi(d,t)$, via equation (5.2). m is a metric for the control given by the network manager. Then, $m(d,t)$ is collected from sink node d periodically and $\overline{m(t)}$ is the average of the metric at time t. Potentials of all sink nodes are controlled according to equation (5.2) simultaneously. The constant value of θ means the intensity of the control.

$$\Phi(d, t+1) = \Phi(d, t) * (1 - \theta \frac{m(d,t) - \overline{m(t)}}{\overline{m(t)}}). \tag{5.2}$$

5.2 Perturbation Model

We assume four types of perturbations that cause topological changes in the network.

Varying wireless channel condition:

The burst packet errors occur in various timescale as mentioned in Reference [84]. Then, we assume that burst packet errors happen due to varying wireless channel condition according to the Gilbert-Elliot model [85]. In this model, wireless channel is described with two-state Markov chain, that is, each link has two conditions "good" and "bad" respectively and alternates the conditions stochastically. In this chapter, when a condition of a link is "good", no bit error occurs in the link and when "bad", bit error and packet loss always happen. The probabilistic transition of the channel condition occurs at fixed cycles T_c.

Node mobility:

The individual sensor node (except for sink nodes) is based on the random waypoint model [86]. A node determines a destination and moves there with constant speed. After arriving at the destination, it pauses for a definite period and moves toward a new destination again. This destination and speed is randomly chosen.

Node addition/failure:

We assume a random addition and failure of a number of sensor nodes. This node addition occurs at the same time in the simulation, and the same is true for node failures.

5.3 Design Approaches for Control Timescale

In this section, we present design approaches for a controlled self-organization based network particularly focused on control timescales in the MAC, routing layers, and the external control.

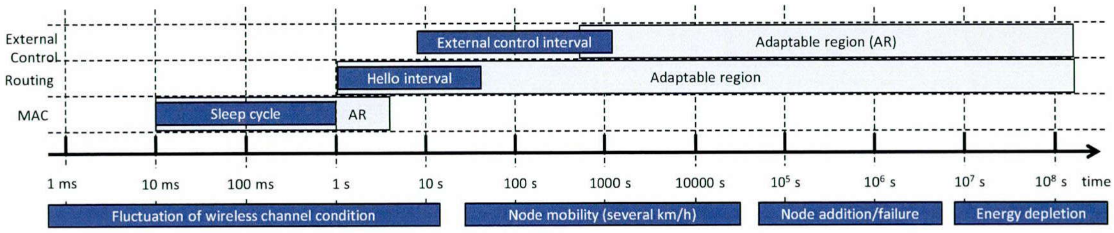


Figure 5.1: Timescale of environmental changes and each layer's control

5.3.1 MAC Layer Design

To changes of wireless channel conditions, which arise with the cycle of 1 ms to 1,000 ms, retransmission in the MAC layer is important. In the MAC layer, a node obtains more opportunities to detect a next hop node when the cycle of sleep control is shorter, in case the node holds a data for a certain period of time (denoted by T_d) until it finishes forwarding the data to the next hop node. However, to the changes with a cycle shorter than this, a MAC layer cannot handle them fundamentally, and we need to choose a robust modulation method against severe changes of radio in the physical layer.

5.3.2 Routing Layer Design

When movement, additions, and failures of nodes occur, latest route information is necessary for data delivery. Therefore, more correct selection of a next-hop node is attained as the updating cycle gets shorter. As well as possibly supposed scenarios on wireless sensor networks, our research supports static and comparatively slow mobility of nodes, taking account for monitoring application of human health, animal behaviour, or etc. Then, approximately tens seconds of periodical messages are used for neighbor detection and message exchanges to maintain route information.

5.3.3 External Control Design

Comparatively long-term perturbations such as movement, additions and failures of sensor nodes may cause global topological changes, which cannot be dealt with by self-organized routing protocols based on only local information. Thus, since these perturbations degrade the performance of such routing protocols, the control and observation mechanism is required for normal operation.

As well as the principle of routing layer design, the shorter control and observation cycle seems to be better. However, this cycle is closely bound together the cycle of self-organized route construction in the routing layer, and therefore, the external control process and self-organized routing process can interfere mutually. In addition, convergence speed of self-organized methods is generally slow and when the external control is conducted before routes do not convergence, a system does not satisfy desired performance.

In order to examine the convergence speed of self-organized potential calculation, first we show analytical solution of the 2-dimensional diffusion equation, $\frac{\partial\phi(x,y,t)}{\partial t} = D\nabla^2\phi(x,y,t)$. Here, we change Cartesian coordinates (x, y) to polar coordinates (r, θ) in order to reduce one of variables ($r_{min} \leq r \leq r_{max}$ and $-\pi \leq \theta \leq \pi$). Since we consider symmetric diffusion of potential from the origin, the solution of the equation is independent of angular coordinate θ. Then, the diffusion equation converted into polar coordinates is as following:

$$\frac{\partial}{\partial t}\phi(r,t) = D(\frac{\partial^2}{\partial r^2}\phi(r,t) + \frac{1}{r}\frac{\partial}{\partial r}\phi(r,t)). \tag{5.3}$$

Various boundary conditions can be found in natural world and we assume two simple Dirichlet boundary conditions: $\phi(r_{min}, t) = \phi_{min}$ and $\phi(r_{max}, t) = \phi_{max}$ ($\phi_{min} < \phi_{max}$). The solution of the equation (5.3) under the conditions is represented by equation 5.4, which is a sum of exponential functions.

$$\phi(r,t) = \sum_{n=0}^{\infty} A_n \mathrm{e}^{-q_n^2 Dt} R(r,n) + C(r). \tag{5.4}$$

In the solution, q_n and A_n are functions of constant number n. Here, q_n ($n = 0, 1, 2, \cdots$) is the real root of the following equation and satisfies the condition $q_k < q_{k+1}$ for any non-negative integer number k:

$$J_0(\phi_{min}q_n)Y_0(\phi_{max}q_n) - Y_0(\phi_{min}q_n)J_0(\phi_{max}q_n) = 0,$$

where $J_0(x)$ and $Y_0(x)$ are the zero-order Bessel function of the first kind and the zero-order Bessel function of the second kind respectively. A_n depends on an initial condition and given an initial condition $\phi(r, 0) = 0$, A_n is calculated according to the following equation:

$$A_n \;=\; -\frac{\pi^2 q_n^2}{2} \frac{Y_0^2(q_n r_{max}) J_0^2(q_n r_{min})}{J_0^2(q_n r_{min}) - J_0^2(q_n r_{max})} \cdot \int_{r_{min}}^{r_{max}} r\Big(a\log(r) + b\Big) \cdot \Big(J_0(q_n r) - \frac{J_0(q_n r_{max})}{Y_0(q_n r_{max})} Y_0(q_n r)\Big) dr.$$

$R(r, n)$ is a function only dependent on n and radial coordinate r as represented in following equation:

$$R(r, n) = J_0(q_n r) - \frac{J_0(q_n r_{min})}{Y_0(q_n r_{min})} Y_0(q_n r).$$

$C(r)$ is represented by a basic logarithm function, $a\log(r) + b$, where a and b are constant number and caluculated as following: $a = \frac{\phi_{max} - \phi_{min}}{\log(r_{max}) - \log(r_{min})}$, $b = \phi_{min} - \frac{\phi_{max} - \phi_{min}}{\log(r_{max}) - \log(r_{min})} \log(r_{min})$.

From equation (5.4), it can be found that the potential $\phi(r, t)$ exponentially converges without relying on the distance from the potential source, but relying on time. In Reference [87], the authors point that the solution of the discrete diffusion equation also exponentially converges. From the above discussion, we could obtain an approximate solution of the diffusion equation. If the solution is represented by a basic exponential function, $f(x) = u\,\mathrm{e}^{-\frac{x}{\tau}} + v$, convergence of the potential can be estimated using time constant τ. It is worth noting that calculation of τ requires the value of ϕ after convergence. Therefore, in order to understand the convergence behavior of the system, computer simulation is one of the means.

For an example of the potential convergence, in Figure 5.2, the simulation results of the potential convergence in two grid networks (99 × 99 and 49 × 49), where the center node is a potential source (potential is 100) and the outer circumferential nodes have potential of zero, are shown. Each node performs potential exchanges with its nearby four or eight nodes, and updates its own potential according to the equation (5.1) every time step. In the figure, the horizontal axis means the time step and the vertical axis is potential of a neighbor node of the center node. The symbols (circle, square, and triangle) mean 90% and 99% convergence from an initial value of zero in each result. From the results, convergence speed becomes more rapid as a network size becomes small and as

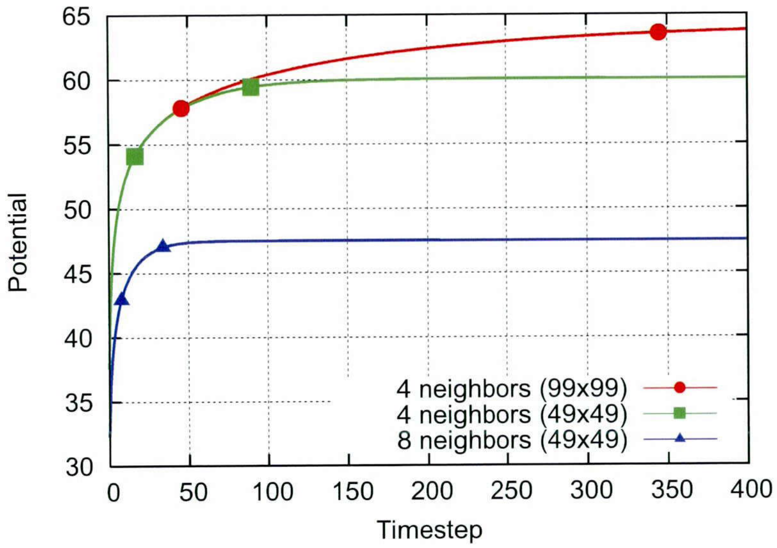


Figure 5.2: Potential convergence in grid networks

communication range increases.

5.4 Simulation Results

In this section, we show the evaluation about the packet delivery ratio under the environmental perturbations which occur periodically. We use an event-driven simulator written by C++ for evaluation. For a network model, we deploy 100 sensor nodes at random places over the square region 500 m on a side, and install a sink node in a corner of the domain. Each sensor node generates one data every 500 s, and it is delivered to the sink node in a multi-hop manner. For a communication model, we utilize the disk model, and communication between two nodes within communication range is successful unless a message collision occurs or wireless channel condition between the nodes is bad. The main parameters in a simulation are shown in Table 5.1.

Table 5.1: Parameter settings for evaluation of a robust network design

Parameters	Value
Transmission speed	100 kbps
Communication range	100 m
Time to live (TTL)	32 hops
T_d	5 s
Channel-condition transition probability (good to bad)	30%
Channel-condition transition probability (bad to good)	70%
Node speed	4–6 km/h
Pause time	250–350 s
Memory span for neighbor potential	250 s
Update interval of potential	50 s

5.4.1 Transitions of Channel Conditions

When the cycle of the sleep control in the MAC layer is set to 0.5 s, 1.0 s, and 2.0 s respectively, the data delivery ratio against the periodic transition of the channel condition is shown in Figure 5.4.1. When the transition of the channel condition arises with the cycle of 10 ms and 100 ms, it turns out that shorter sleep control cycles are required for a high data delivery ratio. Since the MAC layer quickly responds to change in the channel conditions and the opportunity of the retransmission in the MAC layer increases as a sleep control cycle is shorter, even if there is no support in an underlying layer, perturbations with shorter cycle are absorbed. On the other hand, when perturbations occur with the cycle more than 1,000 ms, the delivery performance deteriorates greatly, and above the cycle, the MAC layer cannot handle perturbations. Therefore, it is essential to cope with such perturbations in a higher layer.

5.4.2 Node Mobility

Here, we set the same value to the cycles of potential advertisement and update. In order to eliminate the influence of perturbations other than the cycles, the number of the maximum relay has a sufficiently large value so that there may be no excess. Figure 5.4 shows that the delivery ratio is decreasing as the update cycle of potential becomes large. It is because as longer the update cycle is,

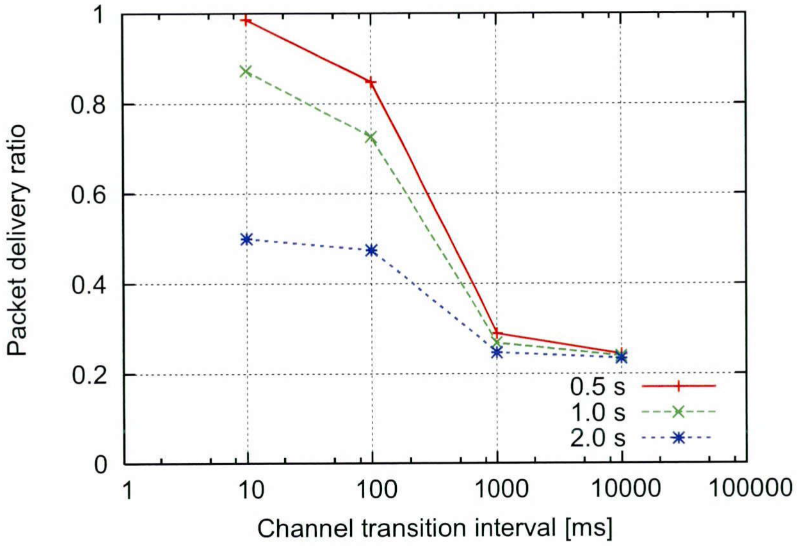


Figure 5.3: Packet delivery ratio against channel condition transition

the higher the possibility that potentials of a node are incorrect, and as a result, the node transmits a data away from the sink node, or discards the data since there have already been no appropriate next-hop neighbor nodes.

Given the update cycle of potential is T_i [s], when a data is generated at a node at a certain time, the elapsed time from the last update is $\frac{1}{2}T_i$ on an average. Since we assume nodes move at 4 – 6 kilometers per hour, the displacements of nodes from the last update is presumed to be $0.55T_i$ – $0.83T_i$ [m]. In our network model, where 100 sensor nodes and square domain with a 500 m side are assumed, the average distance with the nearest node is about 50 m, and therefore, connectivity between the nearest node can change with the cycle of a 10 s order. It is obvious that connectivity with other neighbor nodes can change with much shorter cycle. Therefore, in this network model, it is desirable that the value of the update cycle is at least shorter than 10 s, and when it is set to 10 s, from the simulation result, the delivery ratio more than 95 % is obtained.

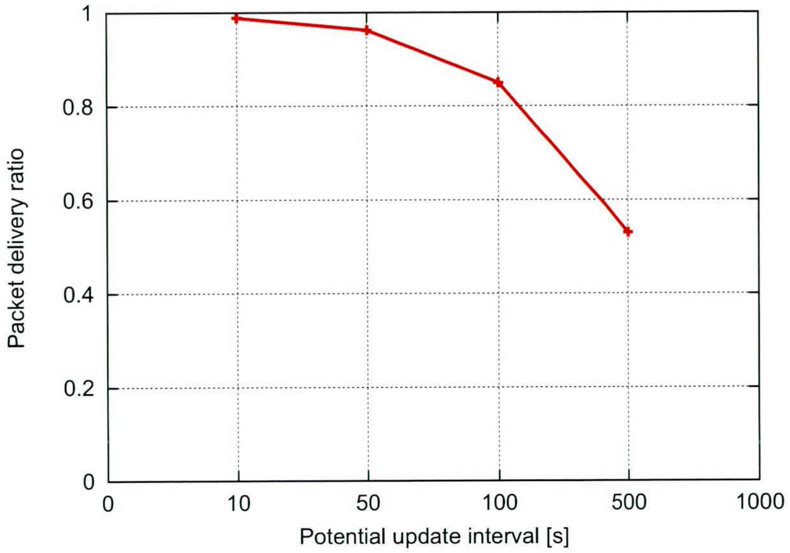


Figure 5.4: Packet delivery ratio against node mobility

5.4.3 Cross-Layer Interaction

Here, we set two sink nodes at the center (sink 1) and a corner (sink 2) of the network. In Figs. 5.5 and 5.6 , we show the potential of two sink nodes (Figs. 5.5(a) and 5.6(a)) and the number of received data by the sink nodes every control interval (Figs. 5.5(b) and 5.6(b)). In those results, 50 sensor nodes are added at a random position at time 20,000 s, and random 50 sensor nodes fail at 40,000 s. The potential update cycle is set to 50 s as shown in Table 5.1, and from the preliminarily experiment, it is found that simulation time of 500 s (100 s) can obtain the 99% (90%) convergence of the potential at each node.

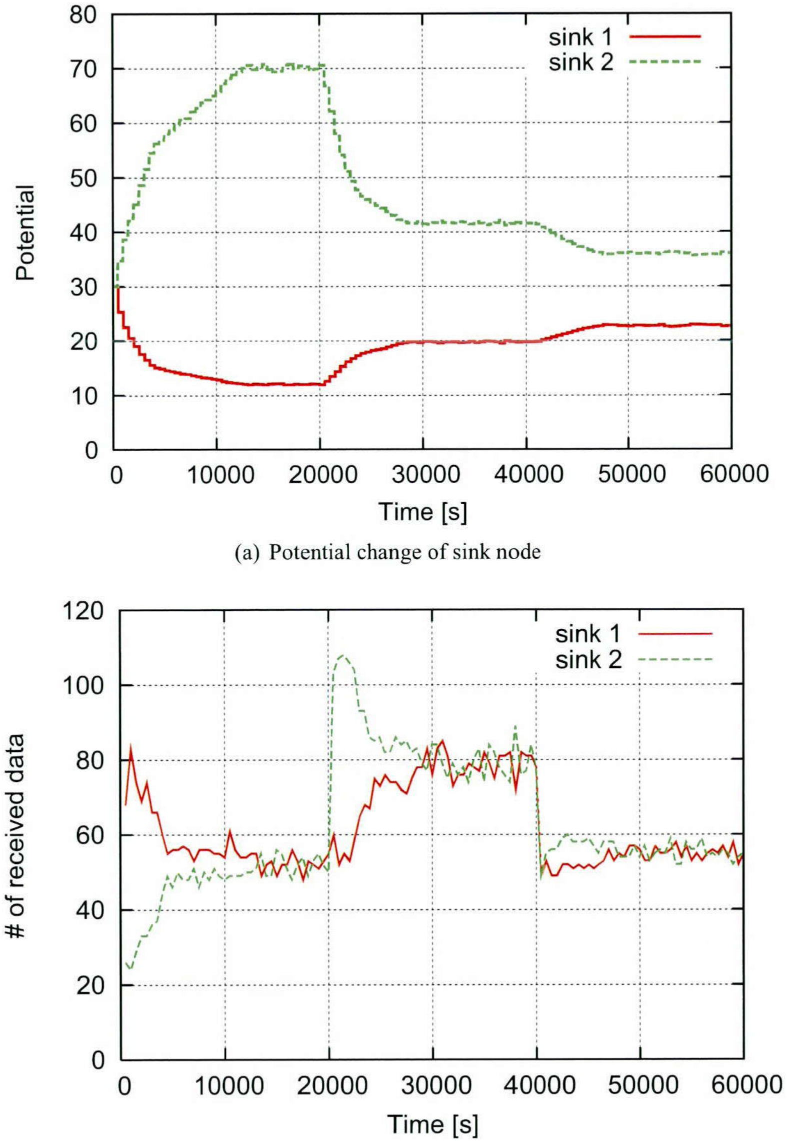
The potential of two sink nodes is controlled by equation (4.11) so that the number of received data mutually becomes equal for every fixed cycle. At the begining of the simulation, more data arrive sink 1. Then, the control node makes potential of sink 1 up in order to reduce the number of received data by sink 1. Equalization of the received data by the sink nodes is attained at 12,000 s as shown in Fig. 5.5(b) and their potential is also converged. Furthermore, equalization of the received data is attained right from the beginning as shown in Fig. 5.6(b).

Meanwhile, some changes takes place to the number of the received data immediately after

20,000 s when addition of 50 nodes occurs. Also in this case, convergence finishes within about 10,000 s (or immediately after the purterbation). Shortly after the failures of sensor nodes at 40,000 s, the number of received data decreases. It is because a convergence commences after nodes erase the potential of failed nodes. The time for erasing depends on the potential memory span shown in Table 5.1. It turns out that after failure, as well as the addition, potential converges.

5.5 Summary

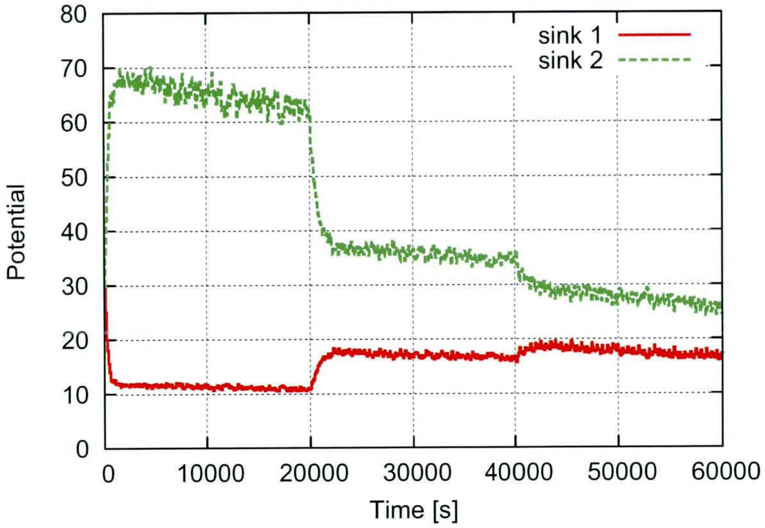
In this chapter, we discussed an approach for network design based on controlled self-organization. This approach is for future large-scale and complex networks. As an example of networks based on controlled self-organization, we focus on a wireless sensor network where a self-organized routing protocol and an external control mechanism are applied. In particular, our concern is on cyclic nature of the environmental perturbations. In order to obtain robustness of a system against environmental perturbations, multiple layers should not handle them separately, but should cope with in a coordinated fashion. We show our approach can deal with various perturbation by appropriate defining the control timescale of each layer.



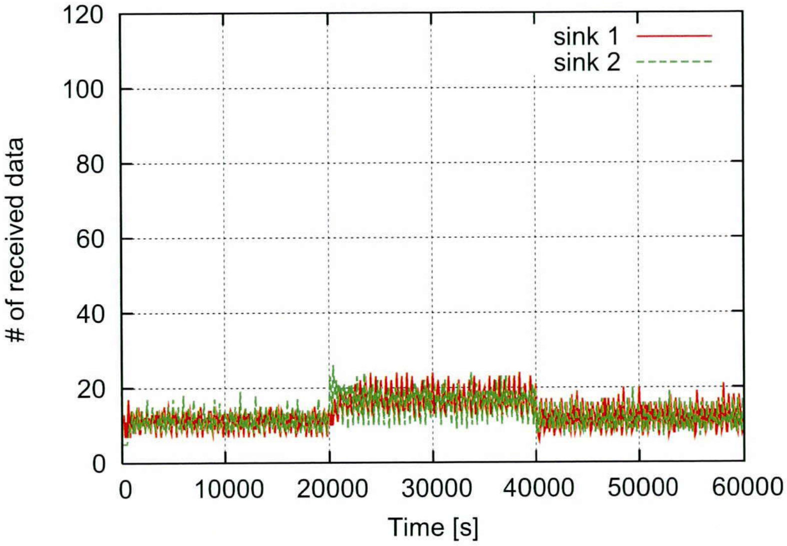
(a) Potential change of sink node

(b) Number of received data by sink node

Figure 5.5: Potential control in case of 2 sinks (control interval 500 s)



(a) Potential change of sink node



(b) Number of received data by sink node

Figure 5.6: Potential control in case of 2 sinks (control interval 100 s)

Chapter 6

Conclusion

Future wireless sensor networks will have massive numbers of elements and should have highly scalable and adaptable properties. Since controlling such a large-scale network is very difficult challenges, self-organization has attracted an increasing attention due to its nature of scalability, adaptability, and robustness. Each element in self-organization makes a decision on the basis of local interactions and local rules, which leads the emergence of global behavior. However, this pure self-organization has some problems because of its bottom-up design, such as difficulty of managing the whole network and slow convergence speed after perturbations. In order for practical realization of a self-organized network, it is desirable that the complicated emerged behavior is manageable, and to this end, controlled self-organization is proposed. In controlled self-organization, an external observer and controller are responsible for guaranteeing that the system behavior remains within the constraint given by the system manager. The main task of the observer is to monitor the system behavior by sampling information of a part of the system elements. The controller evaluates the system behavior reported by the observer and takes control actions to influence the system to achieve a given objective function. This loop of observing and controlling is taken periodically to satisfy the system goal.

First, in Chapter 2, we investigate energy efficiency in wireless sensor networks, which is a significantly important property for battery-limited sensor devices. Our focus is put on the sleep control in the MAC layer, and we evaluate the basic performance characteristics of the receiver-driven asynchronous system, IRDT. Through the computer simulation, we clarify that the performance of receiver-initiated MAC protocols deteriorates due to control-message collisions. Then, we propose some mechanisms for avoiding message collisions in IRDT, for which theoretical derivation of the optimal duty cycle, dynamic control of duty cycle, and a simple data-aggregation mechanism are presented in this chapter. We examined their efficacy in IRDT through a comparison with RI-MAC, and X-MAC, by using computer simulation. We show IRDT can bring about more than a 33% reduction in the average energy consumptions compared with RI-MAC and X-MAC.

Then in Chapter 3, we quantitatively define robustness and resilience in wireless sensor networks. Moreover, we discuss what brings in robustness and resilience and how improve them in the MAC layer and the routing layer. Computer simulation experiments verify that receiver-initiated MAC protocols are compatible with the soft-state mechanism and they are more robust than sender-initiated MAC protocols. Adaptive settings of duty cycles are also found to achieve good resilience in the MAC layer. As for the routing layer, we present leveraging alternative and detour paths bears robustness against random node failures. Monitoring network conditions and highly frequent exchanges of the monitored information yield great resilience.

In a controlled self-organization scheme intended to realize desired network behavior, one or more controllers control a portion of self-organizing nodes through, for example, centralized control. In Chapter 4, we propose controlled potential-based routing, which is based on the controlled self-organization scheme. Sensor nodes calculate their own potential based on interactions with their neighboring nodes, while a control node manages sink-node potentials by centralized control so as to construct a desired potential field. Thus, our proposed routing can obtain good scalability and manageability. We show that load balancing of the sink nodes can be attained in diverse situations and in a large-scale network. Furthermore, potential control based on the energy density can obtain more than four times longer lifetime.

Chapter 5 discusses a design approach for wireless sensor networks based on controlled self-organization. In particular, our concern is on cyclic nature of the environmental perturbation. In order to obtain robustness of a system against environmental perturbations, since different routing layers have quite different control timescale, they should not individually handle various perturbations, but should cope with in a coordinated fashion. We show our approach can deal with various types of perturbations by defining the control timescale of each layer appropriately.

In the future, in a variety of fields, self-organization will be important concept so that the scale of a system becomes much larger. Therefore, managing systems based on self-organization is of increasing significance, and much more investigation on how the controlled self-organization mechanism can control and manage self-organization takes on a growing importance. Our future work contains further research on external control mechanisms in various self-organized systems. Finally, we hope that the discussion in this thesis has implications for future large-scale wireless sensor network research.

Bibliography

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey,” *Computer Networks*, vol. 38, pp. 393–422, Mar. 2002.

[2] M. Weiser, “The Computer for the 21st Century,” *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[3] W. Weber, J. M. Rabaey, and E. Aarts, *Ambient Intelligence.* Springer, 2005.

[4] X. Du, Y. Xiao, and F. Dai, “Increasing Network Lifetime by Balancing Node Energy Consumption in Heterogeneous Sensor Networks,” *Wireless Communications and Mobile Computing*, vol. 8, no. 1, pp. 125–136, Jan. 2008.

[5] S. J. Baek and G. de Veciana, “Spatial Energy Balancing through Proactive Multipath Routing in Wireless Multihop Networks,” *IEEE/ACM Transaction on Networking*, vol. 15, no. 1, pp. 93–104, Feb. 2007.

[6] J. Polastre, J. Hill, and D. Culler, “Versatile Low Power Media Access for Wireless Sensor Networks,” in *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2004, pp. 95–107.

[7] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks,” in *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2006, pp. 307–320.

[8] R. Jurdak, P. Baldi, and C. V. Lopes, “Adaptive Low Power Listening for Wireless Sensor Networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 988–1004, Aug. 2007.

[9] E. A. Lin, J. M. Rabaey, and A. Wolisz, "Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 7, Jun. 2004, pp. 3769–3776.

[10] K. Pister and L. Doherty, "TSMP: Time Synchronized Mesh Protocol," in *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*, Nov. 2008, pp. 391–398.

[11] Y. Wei, H. John, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of the International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, Jun. 2002, pp. 1567–1576.

[12] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511–524, Jun. 2008.

[13] "MICA2," available at https://www.eol.ucar.edu/rtf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf.

[14] T. Hatauchi, Y. Fukuyama, M. Ishii, and T. Shikura, "A Power Efficient Access Method by Polling for Wireless Mesh Networks," *IEEJ Transactions on Electronics, Information and Systems*, vol. 128, no. 12, pp. 1761–1766, Dec. 2008.

[15] F. Kojima, H. Harada, T. Hatauchi, M. Tanabe, K. Sakamoto, A. Kashiwagi, T. Banno, and H. Nishiyama, "Low energy MAC for non-beacon enabled PAN," available at https://mentor.ieee.org/802.15/dcn/09/15-09-0594-01-004e-low-energy-mac-for-non-beacon-enabled-pan.pdf.

[16] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[17] L. Paradis and Q. Han, "A Survey of Fault Management in Wireless Sensor Networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171–190, 2007.

[18] C. Li, H. Zhang, B. Hao, and J. Li, "A Survey on Routing Protocols for Large-Scale Wireless Sensor Networks," *Sensors*, vol. 11, no. 4, pp. 3498–3526, 2011.

[19] C. Prehofer and C. Bettstetter, "Self-Organization in Communication Networks: Principles and Design Paradigms," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 78–85, 2005.

[20] H. Liu, Z.-L. Zhang, J. Srivastava, and V. Firoiu, "PWave: A Multi-Source Multi-Sink Anycast Routing Framework for Wireless Sensor Networks," *LNCS Networking*, pp. 4479:179–190, Nov. 2007.

[21] M. Kalantari and M. Shayman, "Design Optimization of Multi-Sink Sensor Networks by Analogy to Electrostatic Theory," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2006, pp. 431–438.

[22] J. Li, S. Ji, H. Jin, and Q. Ren, "Routing in Multi-Sink Sensor Networks Based on Gravitational Field," in *Proceedings of the International Conference of Embedded Software and Systems (ICESS)*, Jul. 2008, pp. 368–375.

[23] S. Jung, M. Kserawi, D. Lee, and J.-K. K. Rhee, "Distributed Potential Field Based Routing and Autonomous Load Balancing for Wireless Mesh Networks," *IEEE Communications Letters*, vol. 13, no. 6, pp. 429–431, 2009.

[24] A. Basu, A. Lin, and S. Ramanathan, "Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Aug. 2003, pp. 37–48.

[25] H. Ochiai and H. Esaki, "Message Routing on Potential-fields in Forwarding-based DTNs," in *Proceedings of the International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, Jan. 2009, pp. 185–193.

[26] H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L. J. Guibas, "Composable Information Gradients in Wireless Sensor Networks," in *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2008, pp. 121–132.

[27] C. Wu, R. Yuan, and H. Zhou, "A Novel Load Balanced and Lifetime Maximization Routing Protocol in Wireless Sensor Networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, May 2008, pp. 113–117.

[28] F. Ren, J. Zhang, T. He, C. Lin, and S. K. Das, "EBRP: Energy-Balanced Routing Protocol for Data Gathering in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 12, pp. 2108–2125, Dec. 2011.

[29] P. Kumar, J. Kuri, P. Nuggehalli, M. Strasser, M. May, and B. Plattner, "Connectivity-Aware Routing in Sensor Networks," in *Proceedings of the International Conference on Sensor Technologies and Applications (SENSORCOMM)*, Oct. 2007, pp. 387–392.

[30] J. Zhang, Q. Wu, F. Ren, T. He, and C. Lin, "Effective Data Aggregation Supported by Dynamic Routing in Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, May 2010, pp. 1–6.

[31] F. Ren, T. He, S. K. Das, and C. Lin, "Traffic-Aware Dynamic Routing to Alleviate Congestion in Wireless Sensor Networks," *IEEE Transaction on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1585–1599, Sep. 2011.

[32] Y. Liu and W. K. G. Seah, "A Scalable Priority-Based Multi-Path Routing Protocol for Wireless Sensor Networks," *International Journal of Wireless Information Networks*, vol. 12, no. 1, pp. 23–33, Jan. 2005.

[33] C. Intanagonwiwat and D. D. Lucia, "The Sink-based Anycast Routing Protocol for Ad Hoc Wireless Sensor Networks," *Computer Science Department, University of Southern California, Technical Report*, 1999.

[34] C. Müller-Schloer, H. Schmeck, and T. Ungerer, *Organic Computing–A Paradigm Shift for Complex Systems.* Birkhäuser, 2011.

[35] 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 *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*, 2006, pp. 185–191.

[36] D. Kominami, M. Sugano, M. Murata, T. Hatauchi, Y. Fukuyama, and T. Shikura, "Evaluation of Intermittent Receiver-Driven Data Transmission on Wireless Sensor Networks," *Technical Report of IEICE (IN2008-155)*, vol. 108, no. 458, pp. 139–144, Mar. 2009 (in Japanese).

[37] D. Kominami, M. Sugano, M. Murata, T. Hatauchi, and Y. Fukuyama, "Performance Evaluation of Intermittent Receiver-driven Data Transmission on Wireless Sensor Networks," in *Proceedings of the International Symposium on Wireless Communication Systems (ISWCS)*, Sep. 2009, pp. 141–145.

[38] D. Kominami, M. Sugano, M. Murata, T. Hatauchi, and J. Machida, "Performance Improvement by Collision Avoidance Mechanism in Receiver-Driven Multi-Hop Wireless Networks," *Technical Report of IEICE (AN2009-32)*, vol. 109, no. 247, pp. 65–70, Oct. 2009 (in Japanese).

[39] D. Kominami, M. Sugano, M. Murata, T. Hatauchi, and J. Machida, "Energy Saving in Intermittent Receiver-Driven Multi-hop Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, Jun. 2010, pp. 296–303.

[40] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, "Energy-Efficient Receiver-Driven Wireless Mesh Sensor Networks," *Sensors*, vol. 11, no. 1, pp. 111–137, Jan. 2011.

[41] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, "Robustness of Receiver-Driven Multi-Hop Wireless Network with Soft-State Connectivity Management," *The Papers of Technical Meeting on Information Systems, IEE Japan (IS-10-038)*, pp. 81–86, May 2010 (in Japanese).

[42] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, "Robustness of Intermittent Receiver-Driven Wireless Networks Against Fluctuations of Wireless Channel Quality," *Technical Report of IEICE (AN2010-21)*, vol. 110, no. 129, pp. 63–68, Jul. 2010 (in Japanese).

[43] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Robustness of Receiver-Driven Multi-Hop Wireless Network with Soft-State Connectivity Management,” in *Proceedings of the 5th International Conference on Systems and Networks Communications (ICSNC)*, Aug. 2010, pp. 46–51.

[44] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Robust and Resilient Data Collection Protocols for Multihop Wireless Sensor Networks,” *IEICE Transactions on Communications*, vol. E95-B, no. 9, pp. 2740–2750, Sep. 2012.

[45] J. C. Lui, V. Misra, and D. Rubenstein, “On the Robustness of Soft State Protocols,” in *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, Oct. 2004, pp. 1–11.

[46] S. Raman and S. McCanne, “A Model, Analysis, and Protocol Framework for Soft State-based Communication,” in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, Aug. 1999, pp. 15–25.

[47] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Controlled Potential-Based Routing for Large-Scale Wireless Sensor Networks,” *Technical Report of IEICE (AN2010-48)*, vol. 110, no. 377, pp. 25–30, Jan. 2011 (in Japanese).

[48] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Controlled Potential-based Routing for Large-Scale Wireless Sensor Networks,” in *Proceedings of The 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Oct. 2011, pp. 187–195.

[49] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Controlled and Self-Organized Routing for Large-Scale Wireless Sensor Networks,” to appear in *ACM Transactions on Sensor Networks*, Mar. 2012.

[50] D. Kominami and M. Murata, “A Design Approach for Controlled and Self-Organized Networks Focused on Control Timescale,” *IEICE Technical Committee on Information Network Science (NetSci)*, Aug. 2012 (in Japanese).

[51] D. Kominami and M. Murata, “A Design Approach for Managed Self-Organization based Sensor Network Focused on Control Timescale,” in submitted to *the 10th Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, Dec. 2012.

[52] D. Kominami and M. Murata, “A Design Approach for Managed Self-Organization based Sensor Networks Focused on Control Timescale,” submitted to *International Journal of Distributed Wireless Sensor Networks*, Dec. 2012.

[53] J. Zheng and M. J. Lee, “A Comprehensive Performance Study of IEEE 802.15. 4,” *Sensor network operations*, pp. 218–237, 2004.

[54] K. Zeng, K. Ren, W. Lou, and P. J. Moran, “Energy Aware Efficient Geographic Routing in Lossy Wireless Sensor Networks with Environmental Energy Supply,” *Wireless Networks*, vol. 15, no. 1, pp. 39–51, 2009.

[55] Y. M. Lu and V. W. S. Wong, “An Energy-Efficient Multipath Routing Protocol for Wireless Sensor Networks,” *International Journal of Communication Systems*, vol. 20, no. 7, pp. 747–766, 2006.

[56] J. Garcia-Luna-Aceves and A. Tzamaloukas, “Receiver-Initiated Collision Avoidance in Wireless Networks,” *Wireless Networks*, vol. 8, no. 2, pp. 249–263, 2002.

[57] Y. Sun, O. Gurewitz, and D. B. Johnson, “RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks,” in *Proceedings of the ACM conference on Embedded network sensor systems (SenSys)*, Nov. 2008, pp. 1–14.

[58] C. P. Singh, O. P. Vyas, and M. K. Tiwari, “A Survey of Simulation in Sensor Networks,” in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation (SIMCA)*, Dec. 2008, pp. 867–872.

[59] C. Damdinsuren, D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Lifetime Extension Based on Residual Energy for Receiver-Driven Multi-Hop Wireless Network,” *Cluster Computing*, pp. 1–12, May 2012.

[60] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC Essentials for Wireless Sensor Setworks,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[61] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, “PW-MAC: An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks,” in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2011, pp. 1305–1313.

[62] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, “RSVP: A new Resource ReSerVation Protocol,” *IEEE Network*, vol. 7, no. 5, pp. 8–18, Sep. 1993.

[63] M. Handley, H. Schulzrinne, E.Schooler, and J. Rosenberg, “SIP: Session Initiation Protocol,” *RFC 2543, Internet Engineering Task Force*, 1999.

[64] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei, “An Architecture for Wide-Area Multicast Routing,” in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM)*, Sep. 1994, pp. 126–135.

[65] M. Handley, C. Perkins, and E. Whelan, “Session Announcement Protocol,” *RFC 2974, Internet Engineering Task Force*, 2000.

[66] K. Akkaya and M. Younis, “A Survey on Routing Protocols for Wireless Sensor Networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[67] H. Alwan and A. Agarwal, “A Survey on Fault Tolerant Routing Techniques in Wireless Sensor Networks,” in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM)*, Jun. 2009, pp. 366–371.

[68] T. Watteyne, A. Molinaro, M. G. Richichi, and M. Dohler, “From MANET To IETF ROLL Standardization: A Paradigm Shift in WSN Routing Protocols,” *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 688–707, 2011.

[69] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.

[70] Y. T. Hou, Y. Shi, and H. D. Sherali, "Optimal Base Station Selection for Anycast Routing in Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, pp. 813–821, May 2006.

[71] Y. Gu, Y. ji, J. Li, H. Chen, B. Zhao, and F. Liu, "Towards an Optimal Sink Placement in Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Communication (ICC)*, May 2010, pp. 1–5.

[72] H. Kim, Y. Seok, N. Choi, Y. Choi, and T. Kwon, "Optimal Multi-Sink Positioning and Energy-Efficient Routing in Wireless Sensor Networks," in *Proceedings of the International Conference on Information Networking*, Jan. 2005, pp. 264–274.

[73] M. Soyturk and T. Altilar, "A Novel Stateless Energy-Efficient Routing Algorithm for Large-Scale Wireless Sensor Networks with Multiple Sinks," in *Proceedings of the 8th Annual IEEE Wireless and Microwave Technology Conference (WAMICON)*, Dec. 2007, pp. 1–5.

[74] Z. Vincze, R. Vida, and A. Vidacs, "Deploying Multiple Sinks in Multi-Hop Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Pervasive Services (ICPS)*, Jul. 2007, pp. 55–63.

[75] E. I. Oyman and C. Ersoy, "Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks," in *Proceedings of the IEEE International Conference on Communication (ICC)*, Jun. 2004, pp. 3663–3667.

[76] A. Das and D. Dutta, "Data Acquisition in Multiple-Sink Sensor Networks," *ACM SIGMOBILE Computing and Communications Review*, vol. 9, no. 3, pp. 82–85, Jul. 2005.

[77] N. Patwari, J. N. Ash, S. Kyperountas, A. O. H. III, R. Moses, and N. S. Correal, "Locating the Nodes: Cooperative Localization in Wireless Sensor Networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[78] M. Caesar, M. Castro, E. B. Nightnagale, G. O'Shea, and A. Rowstron, "Virtual Ring Routing: Network Routing Inspired by DHTs," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 351–362, Sep. 2006.

[79] A. Awad, R. German, and F. Dressler, "Exploiting Virtual Coordinates for Improved Routing Performance in Sensor Networks," *IEEE Transaction on Mobile Computing*, May 2009.

[80] Y. Zhao, Y. Chen, B. Li, and Q. Zhang, "Hop ID: A Virtual Coordinate-Based Routing for Sparse Mobile Ad Hoc Networks," *IEEE Transaction on Mobile Computing*, vol. 6, no. 9, pp. 1075–1089, Sep. 2007.

[81] A. Caruso, S. Chessa, S. De, and A. Urpi, "GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Mar. 2005, pp. 150–160.

[82] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic Routing without Location Information," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Sep. 2003, pp. 96–108.

[83] I. Dietrich and F. Dressler, "On the Lifetime of Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 1–39, Feb. 2009.

[84] T. Rusak and P. Levis, "Burstiness and Scaling in the Structure of Low-Power Wireless Links," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 1, pp. 60–64, Jun. 2009.

[85] E.N. Gilbert and et al, "Capacity of a Burst-Noise Channel," *Bell. System Technical Journal*, vol. 39, no. 9, pp. 1253–1265, Sep. 1960.

[86] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, pp. 153–181, 1996.

[87] A. Cunha, R. Teixeira, and L. Velho, “Discrete Scale Spaces via Heat Equation,” in *Proceedings of XIV Brazilian Symposium on Computer Graphics and Image Processing*, Oct. 2001, pp. 68–75.