Design Methodology
# of a Wireless Sensor Network Architecture
for Urgent Information Delivery

January 2008

Tetsuya KAWAI

博甲 12756

Design Methodology of a Wireless Sensor Network Architecture for Urgent Information Delivery

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2008

Tetsuya KAWAI

List of Publications

Journal Papers

1. Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata, "Proposal of an Assured Corridor Mechanism for Urgent Information Transmission in Wireless Sensor Networks," IEICE Transaction on Communications E90-B 10, pp.2817–2826, Oct. 2007.

2. Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata, "Design and Evaluation of a Wireless Sensor Network Architecture for Urget Information Transmission," International Journal of Sensor Networks, submitted.

3. Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata, "Design Methodology of a Sensor Network Architecture Supporting Urgent Information and its Evaluation," IEICE Transaction on Communications, submitted.

Conference Papers

1. Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata, "A fast and reliable transmission mechanism of urgent information in sensor networks," Proceedings of the 3rd International Conference of Networked Sensing Systems (INSS 2006), Chicago, Illinois, USA, pp. 14–20, Jun. 2006.

2. Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata, "ACM: A Transmission Mechanism for Urgent Sensor Information," Proceedings of the 26th IEEE International Performance, Computing, and Communications Conference (IPCCC 2007), New Orleans, Louisiana, USA, pp.562–569, Apr. 2007.

3. Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata, "Design, Proposal, and Experiments of a Wireless Sensor Network Architecture for Urgent Information Transmission," Proceedings of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Sysmtems (MASS 2007), Pisa, Italy, Oct. 2007.

4. Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata, "Design Methodology of a Wireless Sensor Network Architecture for Urgent Information

Transmission," Proceedings of the 3rd Annual Wireless Internet Conference (WiCon 2007), Austin, Texas, USA, Oct 2007.

5. Tetsuya Kawai, Naoki Wakamiya, and Masayuki Murata, "Designing a Sensor Network Architecture for Transmission of Urgent Information," Proceedings of Australasian Telecommunication Networks and Applications Conference 2007 (ATNAC 2007), Christchurch, New Zealand, pp.169–174, Dec 2007.

6. Tetsuya Kawai, Naoki Wakamiya, Masayuki Murata, Kentaro Yanagihara, Masanori Nozaki, and Shigeru Fukunaga, "A Sensor Network Protocol for Automatic Meter Reading in an Apartment Building," 2008 IFIP Conferense on Wireless Sensor and Actor Networks (WSAN 2008), Ottawa, Ontario, Canada, Jul 2008, submitted.

Preface

Recent advances in micro-electro-mechanical systems (MEMS) have enabled a tiny device to have a wireless communication feature onto a small package together with the sensing capability, which is called a sensor node. A concept of networked sensor systems to monitor environment has emerged following these advances of hardware packaging technology. A wide variety of industrial, health, agricultural, and environmental applications of WSNs have been considered, for example, target tracking, health monitoring, and disaster detection. In a typical scenario, a number of sensor nodes are densely deployed into the region to monitor in an uncoordinated manner, thrown from an airplane in an extreme case. Then,

1. Each sensor node obtains one or more physical quantities, such as temperature, acceleration, concentration of chemical substances, and so on, by equipped sensors.

2. These sensor data are collected at a special node, called a base station (BS) or a sink, through a wireless mesh network established among sensor nodes.

3. Finally, necessary actions would be taken by an interested person, for example, to call a fire station if the data indicates a sign of a fire.

For each of these three essential phases, *i.e.*, sensing, communication, and utilization of sensor data, we have many problems to solve in order to put this technology to practical use. For example,

1. How can we guarantee that entire region to monitor is covered by deployed sensor nodes?

2. How can we transmit sensor data from such a large number of nodes to a BS? Some of sensor nodes may be few kilometers away from the BS.

3. How can we determine what is happening from raw sensor data?

The first one, the sensing coverage problem has been well studied so far and the last one is related to sensor fusion or data mining techniques. In this thesis, we focus our attention on the second, *i.e.*, communication in a wireless sensor network (WSN).

Although WSNs share the same or similar concepts in many aspects with mobile ad hoc networks (MANETs), there are some differences: large number of nodes, limited power resources, limited computational capacities, and so on. These unique features of a WSN pose many challenging issues in terms of network control. For example, a centralized control is not feasible in a WSN, since it is common that hundreds or thousands nodes are deployed in a WSN. Therefore, a distributed and self-organizing control is preferred for scalability. In addition, sensor nodes are prone to fail and halt for depletion of battery power and being damaged by harsh environment. Therefore, a WSN must be fault tolerant. We can not directly apply existing network control techniques such as for MANETs to a WSN, because such protocols are likely to be too complicated and put too much burden on sensor nodes with limited computational capacities. It is not an easy task to recharge or replace batteries of such a large number of sensor nodes, we must have an energy-efficient network control protocol to prolong the lifetime of a WSN.

Among a variety of potential applications of WSNs, we concentrate upon WSNs for safe and secure living environment. In a WSN as a social infrastructure, urgent information, for example, a fire alarm or natural disaster warning, should be transmitted fast and reliably, whereas the communication among sensor nodes depends on unstable and unreliable wireless links. Therefore a big issue arises here; *How can we provide fast and reliable transmission of urgent sensor information over such an unreliable communication infrastructure?* This is the question that we try to answer in this thesis.

This thesis is organized as follows. First we propose design methodology of an architecture for fast and reliable transmission of urgent information in WSNs. In this methodology, instead of establishing single complicated monolithic mechanism, several simple and fully-distributed control mechanisms which function in different spatial and temporal levels are incorporated on each node. These mechanisms work autonomously and independently responding to the surrounding situation.

Next we show an example of a traffic control protocol designed following the methodology. In this protocol, sensor information is classified into three traffic classes: critical, important, and normal. To adapt to dynamically changing situations and scale of an emergency, five simple mechanisms working in different spatial and temporal levels are incorporated above the network layer: assured corridor mechanism, scheduled hop-by-hop retransmission, priority queueing, rate control by local congestion detection, and rate control by backpressure. The contribution of each mechanism in emergency events is discussed. We evaluated the performance of the protocol by extensive simulation and practical experiments and our claim was supported by the results of these experiments.

As another example of a protocol designed by the methodology, we also present a WSN protocol for automatic meter reading in a large-scale building. In such a network, an urgent event alarm needs to be transmitted as well as meter readings. We propose a sensor network protocol for this purpose, in which low duty cycle of sensor nodes is achieved while the latency of transmission is guaranteed to be less than a certain bound. In this protocol, sensor nodes undergo a wake-up and sleep cycle whose length is equal to the delay-bound, and each node determines its own timing to be active in a cycle by selecting a time slot. This process is governed by a function, called a time slot assignment function, of its next-hop's time slot so that the nodes further from the BS can have earlier slots. We explore time slot assignment functions to find one which gives low and homogeneous contention over a grid network. The simulation results show that our protocol performs well close to the optimal case.

From now on, more and more WSNs are going to be deployed, and the role of the WSN technology in our society becomes more important. We believe that this thesis will contribute to realization of our safe, secure, and comfortable future life.

Acknowledgements

First of all, I would like to express my sincere gratitude to Prof. Masayuki Murata, Graduate School of Information Science and Technology, Osaka University, under whose direction the studies described in this thesis have been carried out. This work would not have been possible without Dr. Naoki Wakamiya, Associate Professor of Graduate School of Information Science and Technology, Osaka University. His critical comments and unerring guidance are always informative and helpful. I also thank to Prof. Koso Murakami, Prof. Makoto Imase, and Prof. Teruo Higashino, Graduate School of Information Science and Technology, Osaka University, and Prof. Hirotaka Nakano, Cybermedia Center, Osaka University, for reviewing this thesis.

I would also like to thank Mr. Shigeru Fukunaga, Mr. Masanori Nozaki, and Mr. Kentaro Yanagihara, Oki Electric Industry Co., Ltd., for their useful suggestions, fruitful discussion, and providing software and equipments for the experiments conducted in this research work.

Thank you to all the members of Advanced Network Architecture Laboratory in Graduate School of Information Science and Technology, Osaka University, for their detailed and valuable instructions.

I would like to acknowledge Prof. Taro Eguchi, Director of the Museum of Osaka University, for his continued support. His words have always inspired me to aim for higher goals. I also like to express my gratitude to Dr. Nobuo Nakamura, Emeritus Professor of Graduate School of Science, Osaka University. If he had not let me use his HP710 workstation, I would not have been even interested in the field of computer networks.

I wish to thank Mr. Nobuhiko Tada, Mr. Masakado Anbo, and Mr. Akihito Hayashi of Matsushita Electric Industrial Co., Ltd. as well as Mr. Ratanachai Sombat of rarkar.com. My motivation for studies on information networks has been fostered in daily conversation with them.

I can not finish without thanking my parents and sister, who have given me a loving environment where to develop. Finally, I am deeply grateful to my beloved wife, Emi. She has always supported and encouraged me to do my best. It is to them that I dedicate this thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Wireless Sensor Network (WSN) technology is expected to play an essential role for our society in the near future. A WSN consists of a number of sensor nodes and a base station (BS). A node is equipped with a processing unit, a radio transceiver, and sensors. Nodes are deployed in a region to monitor, called sensor field, and environmental information detected by sensors is collected to a BS through wireless communication among sensor nodes (Fig. 1.1). Since nodes are usually battery-driven and it is not an easy task to replace or recharge batteries, an energy-efficient network control is essential to prolong the lifetime of a WSN. Moreover, since sensor nodes are prone to fail and halt by exhaustion of energy resources or being damaged by harsh environment, fault tolerance is one of the primary concerns for a WSN. In addition, the large number of deployed sensor nodes, *e.g.*, hundreds or sometimes thousands, and extremely dynamic nature of wireless communication make a centralized control infeasible, thus a distributed and self-organizing control is preferred from the viewpoint of scalability. Simplicity is also important for a WSN architecture since a node has limited computational and memory resources for constraints of its production cost. These requirements for a WSN pose many challenging issues which do not exist in traditional wired networks. There has been a lot of research work on WSNs in the wide range of research fields, like device, sensors, wireless signal processing, medium access, data transmission, data processing, and applications [1]. Furthermore, maintaining sensing coverage and connectivity [2], data aggregation [3], and synchronization among sensor nodes [4] are also important issues to consider. Among them, those directly related to a networking aspect include energy-aware MAC and routing protocols.

For the MAC layer, a number of energy-aware protocols of contention-based [5, 6, 7, 8] and TDMA-based [9, 10] have been proposed. There are two types of contention-based MAC protocols for saving energy consumption, ones which schedule channel access [5, 6] and the others which employ channel sampling [7]. The major advantage of scheduling is that a sender knows a receiver's wakeup

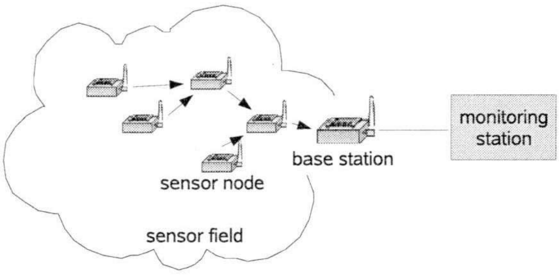


Figure 1.1: A wireless sensor network.

time and thus transmits a packet efficiently. Low-power listening approach with channel sampling, such that in [7], is proposed for reducing the power consumed by idle listening in a contention period. These two techniques are integrated in [8] to achieve a low duty cycle of 0.1 %.

Quite a number of routing protocols for WSNs have also been proposed [11, 12]. According to [11], these protocols are categorized into three classes: flat routing, hierarchical routing, and location-based routing. In flat routing, every node participates in data transmission as a source, a router, and a destination of transmission. Directed diffusion [13] is a typical example of flat routing. In hierarchical routing, neighboring nodes comprise a cluster and elect one among them as a cluster head. The cluster head collects sensor data from its cluster members and forwards the aggregated or fused sensor data to a BS. By devoting a cluster head in energy consuming tasks, cluster members can suppress their energy consumption. To balance energy consumption among nodes, which further leads to the prolonging the lifetime of a WSN, most of hierarchical routing protocols rotate the role of cluster head among sensor nodes [14]. In location-based routing, such as [15], sensor data are forwarded in the direction of a destination node. Thus it usually needs an assumption that every node knows its own and neighbors' location, by embedding a GPS to each node for example.

1.2 Outline of Thesis

The WSN technology will be used for a wide variety of applications, such as agricultural, health, environmental, and industrial purposes. Among them, we focus on a WSN used as a social infrastructure to make our life safe, secure, and comfortable, for example, building automation, disaster detection, and public surveillance systems. This sort of WSNs are supposed to carry some sort of urgent information, such as a fire alarm or intrusion warning, following detection of an abnormal event. Urgent information has to be transmitted through a WSN with higher reliability and lower latency than usual information. In this thesis, we first present design methodology of a WSN architecture for fast and reliable transmission of urgent sensor information, which is the main theme of this thesis. Then two WSN protocols which are designed following the methodology are proposed. One of the two protocols is designated for a general application, where application-dependent data gathering mechanism is used for usual data gathering. The protocol consists of five mechanisms, one of which is our newly developed path-level mechanism for delivery of urgent information. We will show results of performance evaluation of the new mechanism itself as well as those of the entire protocol through extensive simulation and practical experiments. The other protocol is designed following the methodology for a specific application, namely automatic meter reading, and its performance will be verified by simulation experiments. In the rest of this section, we summarize objectives of this thesis with some related works.

1.2.1 Design methodology of a network architecture for fast and reliable urgent sensor information delivery

In Chapter 2, we first introduce an overview of an application system we assume throughout this thesis, and then we propose design methodology of a WSN architecture used for urgent sensor information transmission. An emergency event varies in its scale, from a small one like a gas leakage to a large one like an earthquake attack. In a small scale event, we only have to control nodes along the path from a source of urgent information to a BS for prioritizing urgent information transmission and it is not necessary for other nodes to get involved. However, in the event of a large emergency, a lot of nodes detect it and send urgent information at the same time. A WSN should mitigate a serious congestion caused by this simultaneous emission of a lot of urgent data with a network-wide control mechanism. Therefore, adaptability to the scale of an event is an essential factor in designing a WSN architecture.

In this thesis, we take an approach to combine several simple mechanisms working in different spatial and temporal levels instead of developing a complicated monolithic mechanism. A WSN is able to adapt to the scale of an event by having each of the mechanisms work autonomously and independently according to the surrounding situation.

1.2.2 A sensor network protocol for urgent information

Following the above design policy, we construct UMIUSI (aUtonomous Mechanisms Integrated for Urgent Sensor Information), which will be presented in Chapter 3. In UMIUSI, five simple mechanisms working in different spatial and temporal levels are incorporated for fast and reliable transmission of urgent sensor information. UMIUSI is introduced above the network layer and supposed to be incorporated with an existing data gathering protocol.

There have been several proposals of QoS (Quality of Service) control [16, 17, 18] for WSNs. In ReInForM [19], each node stochastically relays received packets according to the different forwarding probability, and the reliability of transmission is improved through a multipath and retransmission mechanism.

However, congestion mitigation is not considered, which is crucial in a large scale event. ESRT [20] regulates the emission rate at source nodes with feedback from a BS to accomplish the desired reliability. A congestion mitigation mechanism is incorporated in ESRT, however, every decision making is done at a BS, which means that this is apparently a centralized protocol. In [21], overheard packets are used for error correction in single-hop and multihop routing to improve the reliability. This kind of error correction technique can be employed as a neighbor-level or path-level mechanism in our design methodology. The authors of [22] utilize data correlation among sensor nodes and achieve reliable and energy-efficient delivery by allowing each node to predict the data transmission reliability based on the preceding transmissions from other nodes. Although such correlation does not necessarily exist in WSNs we assume in this thesis, this kind of techniques can be incorporated together with our protocol. In [23], the authors propose a routing protocol to find the best path in terms of delay. Felemban *et al.* [24] expands this idea to provide QoS differentiation both in reliability and in latency using a multipath technique. These protocols are based on a distributed approach, but every node needs to be aware of locations of its own and neighbors.

For congestion mitigation in a WSN, CODA [25] combines hop-by-hop backpressure and end-to-end feedback techniques. In [26], a prioritized MAC protocol is introduced in addition to a hop-by-hop traffic control. IFRC [27] is developed to realise adaptive fair and efficient rate allocation by sharing information on the level of congestion among nodes. Siphon [28] introduces multi-radio "virtual sinks" which siphon off data events from the region with high traffic load. It enables congestion mitigation without reducing application fidelity caused by rate control or packet drop mechanisms. However, implementing multi-radio capability leads to increase of the production cost of a sensor node.

Despite many research work on QoS control for WSNs as shown above, few considers the aspect of adaptability to dynamically changing situation or scale of an event. In other words, although they have demonstrated their effectiveness in a specific situation, the contribution diminishes once the situation changes. On the other hand, we focus on the adaptability of network control to the variety of the scale of an event and dynamically changing situation. In comparison to the research work mentioned above, our approach is also unique in that several simple mechanisms are incorporated above the network layer. The expected contribution of each mechanism comprising UMIUSI to fast and reliable transmission of urgent information in a small and large scale event is discussed. Performance evaluation of UMIUSI through simulation and practical experiments is conducted to verify the expected contribution of each mechanism.

1.2.3 A sensor network protocol for automatic meter reading

In Chapter 4, we propose another network protocol following our methodology. Since UMIUSI assumes a general purpose WSN and works above the network layer, its performance in terms of delay and energy-efficiency largely depends on the underlying protocols. In contrast, the second protocol considers an energy-aware routing scheme as well as sleep scheduling for a specific application, that is, automatic meter reading, to guarantee a low duty cycle and an upper bound of end-to-end delay of urgent information.

The trade-off between latency and energy-efficiency has been discussed in some literatures. In [29], the authors discuss the distribution of latency against the active-to-sleep ratio using a Markov model based approach. The delay-bound in a WSN where each node follows a completely uncoordinated sleep schedule is shown in [30]. Cohen and Kapchits [31] tackle the problem of maximization of the lifetime of a WSN under an end-to-end delay constraint. They assume that energy consumption is not governed by sending packets but by the frequency of transition between sleep and active modes. Then, it is shown that energy minimization is achieved by setting the wake-up frequency of a node in accordance with the location of the node in the path. In [32], the authors determine the best path from a node to a BS considering the trade-off between the expected energy consumption and the probability that the latency exceeds a certain threshold under Markovian assumptions on the sleeping schedules and the channel conditions. However, they do not consider reliability of data transmission.

In [33], the authors take into account the sleep scheduling to achieve energy-efficient and fast transmission of sensor information. A node holding data to send first finds a neighbor which is closer to a BS than itself and whose wake-up time is the closest to its own. Then, it adjusts its sleep schedule to have a wake-up interval overlap with that of the designated neighbor. This mechanism reduces the end-to-end delay, but neither contention in the MAC layer nor dynamic topology changes is not considered. Lu *et al.* [34] propose several heuristic algorithms for tree and ring topologies to minimize the communication latency given that each node has a low duty cycle requirement. In [35], the authors consider an efficient wakeup scheduling scheme under bidirectional end-to-end delay constraints and present a multi-parent forwarding technique where multipath routing and wakeup scheduling are integrated. Both low quality of wireless communication and latency caused by sleeping nodes are considered in the dynamic switch-based forwarding [36], which optimizes the expected delivery ratio, expected communication delay, or expected energy consumption. However, contention in the MAC layer is out of consideration, that is, the ideal condition is assumed. In [37], assuming that nodes are synchronized by an existing synchronization protocol, the influence of synchronization error, *i.e.*, phase offset and clock skew, to the trade-off between energy consumption and reliability of transmission under a multi-hop scenario is well studied, but there is no consideration of latency.

Our protocol in Chapter 4 is designed especially for automatic meter reading (AMR) in an apartment building to achieve a low duty cycle with guaranteed delay bound. Key ideas here for our sensor network protocol for AMR (SNPAMR) are (i) sleep interval of a node is less or equal to the delay bound specified by an application, and (ii) each node is assigned a time slot to wake up and receive messages. Unlike most of TDMA-type MAC protocols, where slot assignment is to solve channel contention, in our SNPAMR, an assigned time slot is used not for sending but for receiving messages. A node holding data to send waits for a time slot of its next-hop node. In addition, a node further from a BS is assigned an earlier time slot, so that data reach a BS within the given delay bound. Slot assignment is accomplished in a fully distributed and self-organizing manner, based on a function of the hop distance from a BS. The function is chosen to equalize the contention degree among nodes.

In Chapter 4, we first describe the details of our protocol and we define the contention degree, which represents the intensity of contention at each node, and expected contention degree in a grid network is presented. Next we define a function for time slot assignment based on comparison among possible functions.

Chapter 2

Design Methodology of a Network Architecture for Fast and Reliable Urgent Sensor Information Delivery

2.1 Assumed System

In this thesis, we consider a design methodology of a WSN architecture to transmit urgent information as fast and reliable as possible. We assume a WSN deployed in a building or house to monitor and control living and working environment. A WSN consists of one BS and a number of immobile sensor nodes. The BS corresponds to a gateway server or home server with power supply and sends sensor information to a monitoring station if necessary. Sensor nodes are operated on a battery and equipped with a variety of sensors.

Each node reports obtained sensor information to the BS at regular intervals, which is defined by an application's requirement. Once a node detects an emergency or an unusual condition, it begins to emit packets containing the urgent information. The method for detecting an event is predetermined by an application, and it is out of scope of this thesis. In-network aggregation or sensor fusion techniques can also be applied. In such a case, urgent information would be aggregated or fused with other sensor information, but it must be transmitted faster and more reliably as a whole. On receiving the urgent information, the BS reports the emergency to the monitoring station through a regional network. Then, an acknowledgement is sent back to the BS. The ACK is forwarded to the source node of the urgent information and the node stops sending urgent information.

Although the protocols proposed in this thesis do not depend on any specific MAC or routing protocols, we assume a contention-based MAC protocol and a multihop routing protocol. A time division multiple access (TDMA) protocol is also applicable, but we consider that it has many practical problems to be solved such as scheduling overhead and severe time synchronization requirements. As for the network layer, a multihop scheme with limitation on the radio transmission energy is usually preferred to avoid contention among wireless communication and prolong the lifetime of batteries.

2.2 Detailed Description of the Design Methodology

As mentioned before, the scale of an emergency ranges from a small one to a catastrophic one. Moreover, it is unpredictable and dynamically changing as time passes. When a small event happens, it is not necessary to involve all nodes to respond to the emergency. Instead, only nodes along the path from the node which detects the event to a BS participate in transmission of urgent information and adopt a hop-by-hop retransmission mechanism for example. As the scale of the emergency grows over time, additional mechanisms come into effect and more nodes become involved in the urgent information transmission. On the other hand, in the event of a large emergency, a lot of nodes detect the emergency at the same time and send urgent information simultaneously and independently from others. A WSN in this case should immediately react as a whole to control serious congestion and ensure fast and reliable transmission of urgent information.

In summary, our design objectives of a WSN architecture for transmission of urgent sensor information are:

High reliability and low latency The reliability and latency of transmission of urgent information are the most important issues. Urgent information should be differentiated from other information and receive preferential controls according to their importance. We consider that energy efficiency can be sacrificed to some extent for transmission of urgent information.

Self-organizing and localized behavior The type and scale of an emergency and the number of simultaneous emergency events are unpredictable and dynamically change as time passes. A centralized architecture is infeasible in an emergency due to variations of traffic pattern and the level of congestion. Therefore, we need an architecture which is fully-distributed, self-organizing, and adaptive to dynamically changing conditions. As a consequence of localized reactions of each sensor node to the surroundings and local interactions among nodes, a globally-organized behavior of a WSN against a detected emergency emerges as a whole.

Simplicity Since a node has limited computational capacity and a small amount of memory, mechanisms to support fast and reliable transmission of urgent information must be simple enough. Simplicity also contributes to low energy consumption and less programming errors.

To satisfy the above requirements, we propose design methodology to combine several simple control mechanisms which function in different temporal and spatial levels instead of developing a complicated monolithic mechanism. In Fig. 2.1,

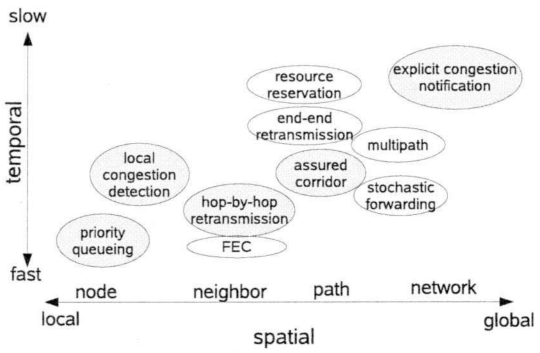


Figure 2.1: Examples of control mechanisms.

typical control mechanisms are arranged in accordance with their temporal and spatial effect. In general, larger the spatial area where a mechanism influences is, longer the time required to achieve effective control is. In the methodology, at least one mechanism is chosen for each of spatial levels. For example, in our UMIUSI protocol discussed later, those five mechanisms indicated by gray circles are embedded in a sensor node. Although another combination with other mechanisms is also possible, we must carefully consider the relation among mechanisms. For example, we can also introduce in-network aggregation [3] for reducing the number of packets and thus suppressing congestion occurrence. However, adopting both hop-by-hop and end-to-end retransmission mechanisms is likely to be inefficient or redundant.

When an emergency occurs, one or more mechanisms among them start working autonomously and independently as a reaction to the surrounding situation of a node. The collective behavior of these mechanisms offers appropriate preferential transmission of emergency packets. At the beginning of an event, quick-acting node-level and neighbor-level mechanisms contribute to reliable and fast transmission until slower path-level mechanisms come into effect. As the event develops and situation becomes more serious, additional mechanisms eventually become effective and network-level control is conducted. In this way, mechanisms with different time-span and working area complement each other and enable a WSN to adapt to the various emergency situations. Note that a node does not detect or conjecture the current situation of an emergency. It does not activate the mechanisms selectively, but all the incorporated mechanisms work in reacting to the occurrence of an emergency and its dynamically changing situation autonomously. A WSN as a whole adapts to the scale of an event by the mechanisms having different contribution in different situations.

We should note here that it is not possible to transmit all emergency packets with high reliability and low latency because the capacity of a WSN is limited. Therefore, it is necessary to classify sensor information into several classes in accordance with the required QoS in terms of delay and reliability. Classification and prioritization can be determined beforehand. Context-aware prioritization is also helpful to adapt to dynamically changing emergency conditions. Each packet has a field in its header to indicate its corresponding class and packets in different classes are treated in a different way in a WSN.

Chapter 3

A Sensor Network Protocol for Urgent Information

3.1 Overview

In this chapter, we assume an application, such as a building automation or plant automation, in which various types of information are transmitted through a WSN. Dozens of immobile nodes with various types of sensors and detectors are deployed in a room. This sort of WSNs have to transmit urgent sensor information with higher reliability and lower latency than other non-urgent information. Since the capacity of a wireless network is limited, a WSN must be capable of differentiating and prioritizing packets depending on their urgency and importance of embedded sensor information.

Since a large number of battery-driven nodes are deployed in a WSN, energy efficiency, fault tolerance, and scalability should be taken into account in designing a WSN architecture. These factors need to be well considered also in the aforementioned WSN. However, in the event of emergency, urgent information must be transmitted as fast and reliable as possible, thus reliability and low latency are primary concerns. Therefore, we need a WSN architecture which satisfies requirements in both of normal and emergency conditions.

There have been a lot of excellent works on data gathering schemes which can be applied in normal situations, for example, [13]. We take an approach here to incorporate mechanisms for urgent information transmission together with any data gathering scheme well-designed for application-oriented communication. It means that a WSN operates on a data gathering scheme in the normal situation. Once an emergency occurs, an appropriate series of actions take place to deliver urgent information to the BS.

3.2 Detailed Description of aUtonomous Mechanisms Integrated for Urgent Sensor Information (UMIUSI)

We construct UMIUSI (aUtonomous Mechanisms Integrated for Urgent Sensor Information) for a WSN for fast and reliable transmission of urgent sensor information following the design policy stated in Chapter 2. First, we consider three classes of sensor information as one normal class and two emergency classes and prioritize emergency class information over normal class information. The two types of urgent information are distinguished in more important and less important. The former includes critical information, *e.g.*, a fire alarm, which should be transmitted with very high reliability and low latency. An example of the latter is temperature information, which is important to monitor the condition of a fire.

- *Normal Class.* Any non-urgent information belongs to this class. Normal class information is gathered to the BS at regular intervals of t_{norm}. Without an emergency, the latency and reliability of normal class information should satisfy the application's requirements, depending on the adopted data gathering scheme. An application can tolerate delay and loss of normal class information under emergency conditions. Packets of this class are called normal packets.

- *Important Class.* This class is for urgent information, but an application can tolerate loss and delay of important class information to some extent. Packets belonging to this class, called important packets, can be delayed or dropped depending on the level of congestion in an emergency. The interval of emission of important packets $t_{\mathrm{imp}} < t_{\mathrm{norm}}$ is determined by an application, but could be regulated to be larger than t_{norm} to mitigate congestion.

- *Critical Class.* This class is for the most urgent and important information which requires highly reliable and fast transmission to the BS. Critical packets are emitted by a node detecting an emergency at a fixed regular interval of $t_{\mathrm{cri}} < t_{\mathrm{norm}}$, which is determined by an application. The total amount of critical class traffic should not exceed the network capacity to guarantee a high delivery ratio and low delay of the required level. Therefore, the number of sensor nodes for critical information should be limited at the deployment, or some of them should be categorized into the important class. It is not a trivial task to determine the number of nodes of the critical class. Because of the application-oriented deployment of sensor nodes, we cannot optimize the number and location of nodes in an actual situation. We plan to consider a design strategy to control the number of nodes of the critical class.

For the sake of simplicity, we adopted equal values of t_{norm}, t_{imp}, or t_{cri} among all sensor nodes in the experiments. However, diverse setting of intervals can be used depending on the application's requirements.

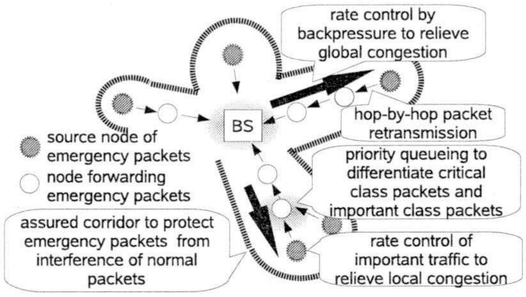

rate control by backpressure to relieve global congestion
BS
hop-by-hop packet retransmission
source node of emergency packets
node forwarding emergency packets
priority queueing to differentiate critical class packets and important class packets
assured corridor to protect emergency packets from interference of normal packets
rate control of important traffic to relieve local congestion

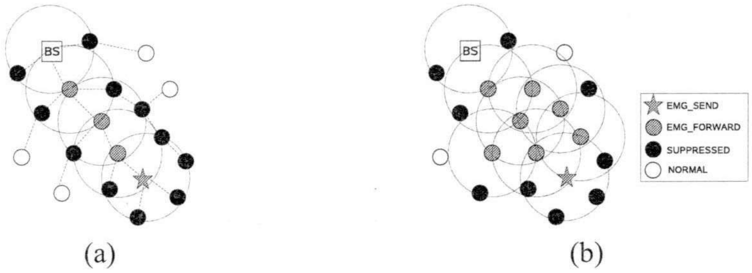
Figure 3.1: The mechanisms leveraged in UMIUSI.

As stated in Chapter 2, mechanisms leveraged in UMIUSI must be simple and work independently of other schemes and protocols. In addition, a WSN should adapt to dynamically changing emergency situations by adopting control mechanisms working in different time and topological levels, from fast and local to slow and global. From these points of view, we incorporate following five mechanisms into UMIUSI: assured corridor mechanism (ACM), retransmission, priority queueing, rate control by local congestion detection, and rate control by backpressure. The spatial and temporal level of each mechanism is shown in Fig. 2.1. Figure 3.1 briefly summarizes how and where they work.

In a normal situation, sensor information is collected from sensor nodes to the BS at a regular interval of t_{norm} by a data gathering scheme. Once an emergency event occurs, a node detecting the emergency emits emergency packets, which are labeled as the critical class or important class, at a regular interval of t_{imp} or t_{cri}, respectively. They are forwarded to the BS. A hop-by-hop retransmission is conducted to recover from a loss of an emergency packet. On the way to the BS, emergency packets establish an "assured corridor" along the path from the source to the BS. By keeping nodes awake in this path and making surrounding nodes refrain from emitting normal packets, an assured corridor protects the emergency packets from collision with normal packets and avoid delay caused by sleeping nodes.

When there are two or more nodes detecting an emergency, there occur collisions among emergency packets. An intermediate node adopts a priority queueing mechanism to offer a better forwarding service to critical packets for higher delivery ratio and lower delay than important packets. In addition, to avoid collisions among radio signals, a source node of important packets regulates its emission rate in accordance with the level of congestion in its vicinity. Furthermore, to mitigate congestion around the BS and at intermediate nodes, the existence of congestion is notified to source nodes of important packets by means of backpressure. On receiving a backpressure message, they reduce the emission rate of important packets.

The detailed description of each mechanism is presented in the following.



(a) (b)

Figure 3.2: An "assured corridor" in (a) a tree-based network and (b) a broadcasting-based network.

3.2.1 Assured corridor mechanism (ACM)

The main purpose of this newly developed mechanism is to avoid loss of emergency packets caused by collisions with normal packets. In addition, ACM contributes to avoiding delay caused by sleeping nodes.

Examples of an assured corridor are illustrated in Fig. 3.2 for a tree-based sensor network and a broadcasting-based sensor network. In the figure, a star corresponds to a node which detects an emergency and becomes a source node of emergency packets. Grey circles correspond to nodes on a path from the source node to the BS and they keep awake during the emergency. Nodes in ranges of radio signals of those grey nodes are denoted as filled circles, which suppress emission of normal packets.

In ACM, a node follows the state transitions illustrated in Fig 3.3. A node stays in the *NORMAL* state in its normal operation. When a node detects an emergency event, its state is changed to the *EMG_SEND* state and it begins emission of emergency packets. The traffic class of an emergency packet is identified by an emergency flag in its header. When a neighbor node in the *NORMAL* state receives or hears the emergency packet, its state moves to either of the *EMG_FORWARD* or *SUPPRESSED* states depending on its location. If the node is on the path to the BS, in other words, if the node is a next-hop of the *EMG_SEND* node, it moves to the *EMG_FORWARD* state. It suspends its sleep schedule and forwards the emergency packet. If the node is not involved in forwarding the emergency packet, *i.e.*, not on the path to the BS, it moves to the *SUPPRESSED* state and suppresses the transmission of normal packets in order to avoid collisions with emergency packets in the MAC layer. A node can recognize its role in transmission of urgent information based on network or MAC layer information. For example, when a tree-based routing protocol is used, a node recognizes that it is on the path to the BS by finding that the destination MAC address of a received packet is its own. In a broadcast-based network where nodes closer to the BS than a sender are expected to forward a packet to the BS, a node only compares its hop distance to the BS with that of a sender to determine its relative location.

Similarly, among neighbor nodes receiving or hearing an emergency packet

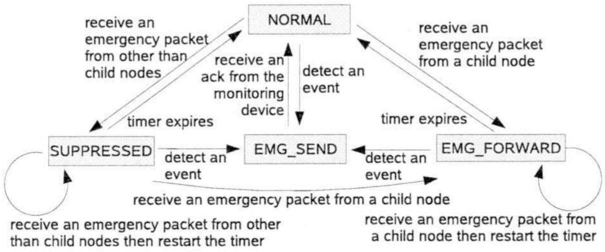

NORMAL
receive an emergency packet from other than child nodes
receive an ack from the monitoring device
detect an event
receive an emergency packet from a child node
timer expires
timer expires
SUPPRESSED
detect an event
EMG_SEND
detect an event
EMG_FORWARD
receive an emergency packet from a child node
receive an emergency packet from other than child nodes then restart the timer
receive an emergency packet from a child node then restart the timer

Figure 3.3: State transitions.

forwarded by the *EMG_FORWARD* node, ones on the path to the BS become *EMG_FORWARD* nodes and the others become *SUPPRESSED* nodes. By repeating this process at every hop to the BS, an assured corridor, which consists of *EMG_FORWARD* nodes forwarding emergency packets along the path and *SUPPRESSED* nodes surrounding the path, is eventually completed when the first emergency packet arrives at the BS.

Once an assured corridor is established, following emergency packets propagate through the corridor which consists of awake nodes forwarding emergency packets and surrounding silent nodes. The rest of the nodes in the WSN are not aware of the emergency and they remain in their normal operation.

These mechanisms imply that the reliability and latency of transmission of emergency packets are improved at sacrifice of the lower delivery ratio and larger transmission delay of non-urgent information and the depletion of a battery of awake nodes. Although low energy consumption is one of the most important requirements in WSNs, we should not sacrifice the reliability and latency of transmission of emergency packets for the energy efficiency. Therefore, we do not pay much attention to energy efficiency in ACM. We believe that such a design policy is acceptable, because it is reasonable to assume that emergency events rarely happen. The lifetime of a WSN depends on energy efficiency not in urgent conditions but in normal operation. If allowed we can introduce a sleep schedule to nodes in a corridor, but it is left as one of our future works.

Detailed description of the four states of a node in ACM is given in the following.

NORMAL As long as there is no emergency event, a WSN operates as usual and nodes are in the *NORMAL* state. They periodically wake up, receive and transmit a data packet, and go back to sleep at regular intervals of t_{norm}.

EMG_SEND When a node detects an emergency event, *e.g.*, a fire, it enters the *EMG_SEND* state. It broadcasts emergency packets with the emergency flag at shorter intervals of t_{cri} or t_{imp} depending on the traffic class of the sensor data. Every emergency packet sent is given a unique sequence number at the source node.

EMG_FORWARD A node which receives an emergency packet for the first time from its preceding nodes moves into the *EMG_FORWARD* state. A preceding node is a node for which the node is responsible in forwarding a packet toward the BS. For example, if the WSN adopts tree topology whose root is the BS, a preceding node is a child node. On receiving the emergency packet, a node first suspends its sleep schedule. Then, it sends the received emergency packet to the designated next-hop node on the path to the BS, after waiting for the activation of the next-hop node if it is in the sleep mode. The next-hop node also keeps awake once it receives the emergency packet. Therefore, following emergency packets sent after the first emergency packet by the source node are immediately relayed by *EMG_FORWARD* nodes toward the BS.

SUPPRESSED A node which receives an emergency packet from a neighboring node which is not its preceding node moves into the *SUPPRESSED* state. A node in this state should suppress transmitting some or all of normal packets.

We assume that a monitoring station or control center receives the urgent information through the BS. Then, an acknowledgment is sent back to the BS and it is forwarded to the source node of the emergency packets. On receiving the acknowledgement, the *EMG_SEND* node returns back to the *NORMAL* state. On the other hand, *EMG_FORWARD* and *SUPPRESSED* are "soft states." Entering these states, a node starts a timer. When the timer expires, it returns to the *NORMAL* state. The timer is restarted every time when a node receives an emergency packet. A typical length of the timer is the interval of data gathering in the *NORMAL* state, *i.e.*, t_{norm}, since emergency packets are sent more frequent than normal packets to inform a control center of up-to-date emergency condition.

Note that an assured corridor is established while the first emergency packet is being forwarded to the BS. Therefore, the transmission delay of the first emergency packet, in other words, the time needed to establish a corridor, depends on the sleep schedule of the data gathering scheme used for normal operation. After a corridor is established, following emergency packets are forwarded immediately by *EMG_FORWARD* nodes, which keep awake, thus the delay is minimal and independent of the sleep schedule.

3.2.2 Retransmission

In order to recover a lost emergency packet and provide differentiated services, we introduce a prioritized scheduling algorithm of hop-by-hop retransmissions, which can be incorporated with most retransmission mechanisms in either MAC layer or network layer, such that in [38].

A node retransmits an emergency packet when it detects a loss. The hop-by-hop acknowledgement can be easily done by, for example, overhearing a packet sent by a next-hop node. If the overheard packet does not contain the information that the node sent, the packet is considered to be lost.

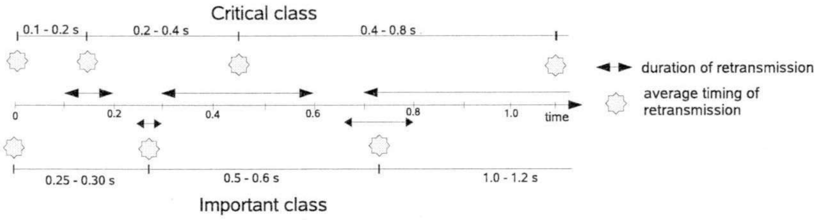


Figure 3.4: An example of retransmission schedule of emergency packets.

The first retransmission is scheduled after a random backoff. To prioritize retransmission of a critical packet, the backoff timer for a critical packet is set shorter than that for an important packet. For example, in Fig. 3.4, the backoff for a critical packet ranges from 0.1 to 0.2 seconds after the first emission and that for an important packet is from 0.25 to 0.3 seconds. If the first retransmission fails, one or more trials are conducted by applying doubled backoff, *i.e.*, a binary backoff scheme, until retransmission succeeds or the time when a next-hop node in the *NORMAL* state goes to sleep. An emergency packet is discarded at a node when it receives the next emergency packet originating at the same source node. This is because that sensor data in the waiting packet is obsoleted by the new data. It is also possible to merge them and generate a new emergency packet depending on an application's requirements.

3.2.3 Priority queueing

Each node has a priority queue for emergency packets, with which important packets are served only when there is no critical packet queued. This means that fast transmission of critical packets is accomplished at the sacrifice of longer transmission delay of important packets. Transmission of normal packets at a node in either the *EMG_SEND* or *EMG_FORWARD* state is delayed until the node moves to the *NORMAL* state.

3.2.4 Rate control by local congestion detection

To mitigate congestion as fast as possible by local control, we introduce a rate control mechanism which is triggered by detection of local congestion. In order to keep the reporting rate of critical information, the rate control is applied only to important class traffic. When a source node of important packets detects congestion by, for example, not receiving any acknowledgement from any of its next-hop nodes, it considers that local congestion is occurring and increases the emission interval of important packets. Congestion detection can be done by other methods, including observation of queue occupancy and channel sampling [25, 26].

As a rate control algorithm, we employ a TCP-like AIMD (Additive Increase and Multiplicative Decrease) algorithm, such as that in [39], for its simplicity.

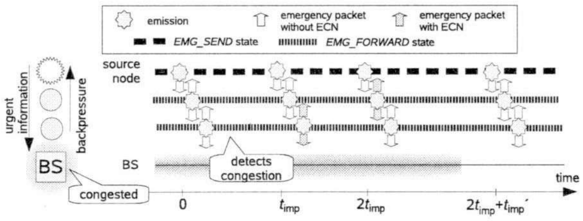


Figure 3.5: The rate control with backpressure.

When an *EMG_SEND* node confirms delivery of an important packet to its next-hop node, its emission rate is increased by decreasing the emission interval t_{imp} as

$$t_{\mathrm{imp}} \leftarrow \frac{t_{\mathrm{imp}}}{1 + \alpha t_{\mathrm{imp}}}, \tag{3.1}$$

where α ($\alpha > 0$) is a constant. The upper bound of the emission rate is determined by the application. When a node detects congestion, its emission rate of important packets is decreased by multiplying the parameter β ($0 < \beta < 1$), which further corresponds to the following adjustment,

$$t_{\mathrm{imp}} \leftarrow t_{\mathrm{imp}}/\beta. \tag{3.2}$$

3.2.5 Rate control by backpressure

In an event of a larger emergency such as an earthquake, even if emission of normal packets is suppressed and source nodes of important packets regulate their emission rate, congestion cannot be fully avoided around a node belonging to multiple paths and around the BS, where many emergency packets concentrate on. One possible way to avoid congestion is to drop some of the important packets at nodes one-hop-closer to the source node. However, it is only a waste of wireless bandwidth to transmit important packets which will be dropped eventually. Thus, we take another approach, which regulates the emission rate of important packets at the source node by giving a feedback from nodes detecting congestion.

To suppress emission of important packets at their source nodes, a backpressure message is sent back to source nodes from a point of congestion by piggybacking on emergency packets (Fig. 3.5). When a node detects congestion, it sets an explicit congestion notification (ECN) bit in the header of important packets which it relays toward the BS. By overhearing the packet, a one-hop-closer node to the source recognizes that congestion occurs in the path to the BS. Then, it also sets an ECN bit of the next important packet. Consequently, a congestion notification propagates to the source node. On receiving the notification, the source node reduces the emission rate of important packets, and the congestion is mitigated. The node which detected the congestion relays important packets without ECN bit once it confirms the mitigation.

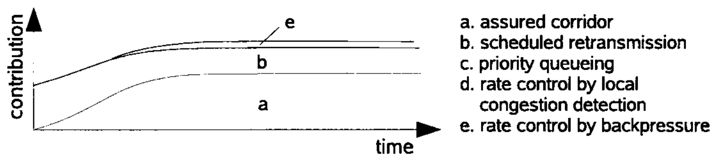


Figure 3.6: The contribution of each mechanism (small scale event).

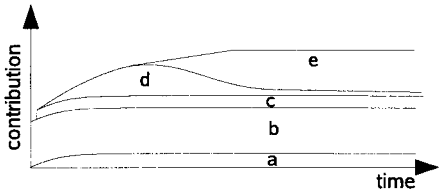


Figure 3.7: The contribution of each mechanism (large scale event).

The propagation of backpressure can be slow. Assuming that the interval of emission of important packets is t_{imp} and congestion is detected at a n-hops distant node from the source node, the estimated propagation delay is $t_{\mathrm{imp}}(n-1)$. However, the backpressure scheme does not involve any additional signaling mechanisms and the only information a node has to retain is whether the last important packet received from a one-hop closer node to the BS has the ECN bit on or not. Thus, the backpressure incurs minimal overhead. In addition, the delay can be shorten with a shorter interval t_{imp}. It means that the more frequent the emission of important packets is, the shorter the delay is. In other words, the more harmful important class traffic is, the faster the control works.

The emission rate of important packets is regulated by the same AIMD algorithm as in the rate control by local congestion detection. When there are multiple paths from the source node to the congested node, the source node would be notified about the congestion several times for one important packet. To avoid excessively reacting to the congestion, only one notification during the interval t_{imp} is taken into account for rate reduction.

3.3 Example Scenarios

Now, let us consider some example scenarios. For a small scale event, a gas leakage warning for example, only one or a few nodes would detect it. Among five mechanisms described in the previous section, the priority queueing is not effective since all emergency packets are likely to belong to one class. In addition, the rate control of important class traffic by local congestion detection does not help much, since the number of nodes emitting important packets is small and thus the possibility of congestion is expected to be small. If multiple paths are established from the source node to the BS, collisions may occur among emergency packets traversing different paths. In such a case, the rate control by backpressure is activated. ACM is the most effective among the five, because loss of emergency packets would be mainly caused by collisions with normal packets. In addition, the retransmission is necessary, since an emergency packet is not protected from normal packets until an assured corridor is established. Figure 3.6 is an intuitive sketch to show how much each mechanism contributes to the fast and reliable transmission of emergency packets in a small scale event. As an assured corridor is gradually established as emergency packets are forwarded to the BS, the contribution of ACM becomes larger.

On the other hand, in the event of a large scale emergency such as an earthquake, many nodes detect the emergency and emit a variety of sensor information. Figure 3.7 illustrates the degree of contribution of the mechanisms for the case of a large scale event. Since most of the nodes are involved in transmission of emergency packets as source nodes in the *EMG_SEND* state or forwarding nodes in the *EMG_FORWARD* state, an assured corridor to suppress the emission of normal packets does not help much. On the contrary, mechanisms to mitigate congestion within a corridor are effective. The priority queueing mechanism offers differentiated forwarding services to emergency packets in accordance with their class. Rate control is first applied locally at a source node to mitigate local congestion among neighboring source nodes. Furthermore, congestion among emergency packets traversing different paths is solved by the backpressure mechanism.

All of these reactions against different types of emergencies emerge as a consequence of the autonomous and simple behavior of nodes, which is determined only by their states and emergency packets received from neighbors. Thus UMIUSI is a self-organizing and distributed protocol. There is neither a mechanism to identify the type and scale of an emergency nor an explicit rule to choose and coordinate mechanisms.

3.4 Simulation Model

We implemented UMIUSI on the ns-2 network simulator package for performance evaluation. The simulation experiments are conducted in three stages. First, we clarify basic behavior of ACM with a tree-based and broadcast-based network. The performance of UMIUSI is then evaluated. Finally, we compare the performance of UMIUSI with that of another service differentiation protocol. In the simulation experiments, 200 (referred as low density) or 500 (high density) nodes are uniformly and randomly distributed in a 20 m × 20 m two-dimensional region with a BS at its center. IEEE 802.15.4 [40] non-beacon mode is used as the MAC protocol and the transmission range of radio signals is set to 2.5 m.

We employ a general broadcast-based or tree-based routing protocol for the underlying network layer. In both routing protocols, we assume that each node knows its own hop distance from the BS. In the broadcast-based routing, a packet

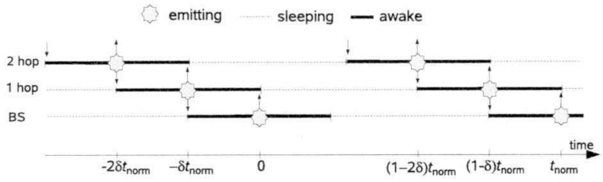

emitting
sleeping
awake
2 hop
1 hop
BS
time
$-2\delta t_{\mathrm{norm}}$
$-\delta t_{\mathrm{norm}}$
0
$(1-2\delta)t_{\mathrm{norm}}$
$(1\text{-}\delta)t_{\mathrm{norm}}$
t_{norm}

Figure 3.8: The transmission sequences in normal operation.

contains the sensor data and the hop distance of the sender. A node forwards a packet, if the hop distance of the sender is larger than that of itself. Otherwise, it simply drops the packet. Since a packet can be received and forwarded by multiple nodes in the sender's vicinity, the broadcast-based routing can be categorized into multipath routing protocols without explicit path establishment. An example of such a routing protocol is the synchronization-based data gathering scheme [41]. In the tree-based routing, every node chooses a next-hop node among neighbors which are closer to the BS by one hop, based on the received signal strength. This is equivalent to choosing the nearest one-hop-closer node in the simulation experiments.

The schedule of transmission is shown in Fig. 3.8 for normal operation. In the *NORMAL* state, a node sends packets at regular intervals of t_{norm}. The instant that a node at n hops from the BS sends a packet is earlier than the instant that an $(n-1)$-hop node sends a packet by δt_{norm}. It means that it takes $(n-1)\delta t_{\mathrm{norm}}$ for sensor data of n-hop node to reach the BS. Here, δ is a coefficient which governs the interval of packet emission between nodes of adjacent hops. Based on the sleep schedule, an n-hop node wakes up at δt_{norm} before the timing of its packet emission to receive packets from preceding $(n+1)$-hop nodes. It aggregates the received data with its own sensor data and then it sends the packet at the time when $(n-1)$-hop nodes wake up. Once it overhears the packet of an $(n-1)$-hop node at δt_{norm} after the emission, it goes to sleep. The detailed discussion on applying ACM to the synchronization-based data gathering scheme is presented in Appendix A.

In our simulation experiments, t_{norm} and δ are set at 10 seconds and 0.1. Before sending a packet, the random backoff of 10 ms at maximum is applied in the network layer to ease the collision situation. The size of sensor information is 6 bytes. Since we do not assume data fusion, N sensor information amounts to $6N$ bytes. The maximum size of the payload of a packet is limited to 78 bytes due to the limitation of IEEE 802.15.4 and sensor information exceeding this limitation is discarded.

We also implemented a stochastic forwarding scheme for comparison purposes. One example of such scheme is Adaptive Forwarding Scheme (AFS) proposed in [42]. In this scheme, a node which receives a packet decides whether it forwards or drops the received packet according to a probability. If the forwarding probability is equal to one, a node forwards all packets. The forwarding probability can

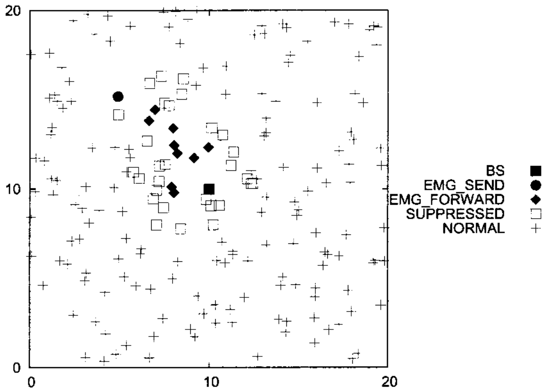


Figure 3.9: An "assured corridor" in the broadcast-based network.

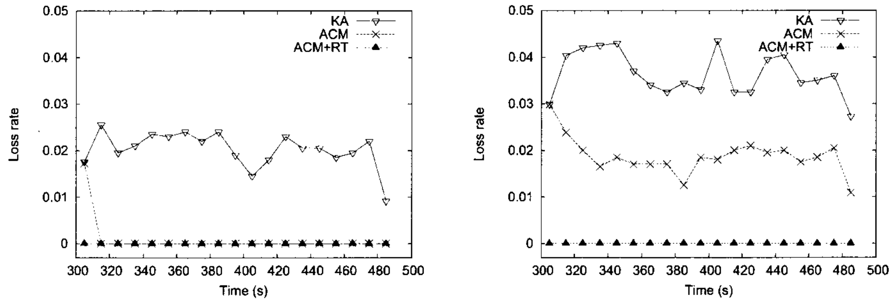
be greater than one, which means that a node stochastically retransmits a packet in addition to its first emission. In our simulation experiments, only *EMG_SEND* and *EMG_FORWARD* nodes follow this scheme to improve the reliability of the transmission of emergency packets. Although the forwarding probability is dynamically updated by feedback from the BS in AFS, we used a fixed probability. For fair comparisons, we conducted a number of simulation experiments with different static forwarding probabilities to find the values which offered the almost same delivery ratio as UMIUSI did. The suppression of normal packets, priority queueing, and rate control mechanisms adopted in UMIUSI are not applied in the experiments of AFS.

3.5 Results and Discussion

In this section, some results of simulation experiments are presented. Note that emergency packets sent before an assured corridor is established are not taken into these results, because the loss rate and delay of these packets depend largely on the underlying routing algorithm or data gathering scheme.

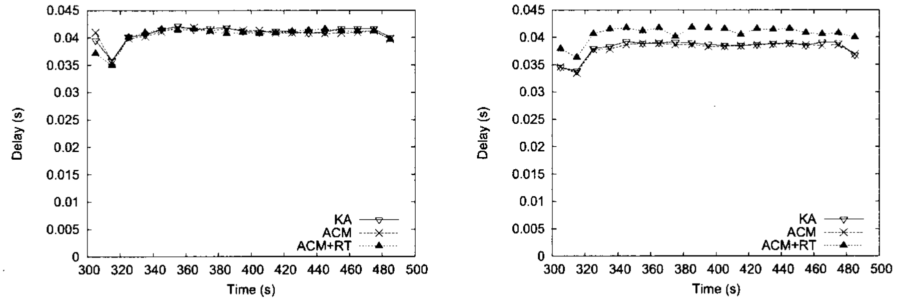
3.5.1 Evaluation of ACM

For the evaluation of ACM, the low density scenarios with the tree-based and broadcast-based routing are adopted. Each simulation experiment lasts for 500 seconds including 300 seconds for initialization without any emergency. After that, each of randomly chosen 1, 2, 4 or 8 nodes moves to the *EMG_SEND* state at randomly chosen time in following 10 seconds. They begin sending emergency packets at the rate of 2 packets/s, *i.e.*, $t_{\mathrm{cri}} = 0.5$ seconds. Each of them goes back to the *NORMAL* state at 180 seconds after it moves to the *EMG_SEND* state. The same experiment is repeated for 100 times with different node layouts. Fig. 3.9 shows a snapshot in one of the simulation experiments with one *EMG_SEND* node.



(a) (b)

Figure 3.10: The loss rate of emergency packets in (a) a tree-based network and (b) a broadcast-based network with one *EMG_SEND* node.



(a) (b)

Figure 3.11: The delay of emergency packets in (a) a tree-based network and (b) a broadcast-based network with one *EMG_SEND* node.

For comparison purposes, we considered three variants of the mechanism. One is called as KA (keep awake), in which only the *EMG_SEND* and *EMG_FORWARD* states are applied and no suppression of normal packets is conducted. Another is called ACM, in which an assured corridor is established by suppressing emission of normal packets, but lost packets are not recovered by retransmission. The other is called ACM+RT, in which retransmission is applied in addition.

Loss rate of emergency packets

First, we consider the case of one *EMG_SEND* node. The loss rate of emergency packets against simulation time is shown in Fig. 3.10. The loss rate is defined as the ratio of the number of emergency packets not received by the BS to the number of those sent from the source node after a corridor is completely established.

In the tree-based network without suppression (KA in Fig. 3.10(a)), about 2 % of emergency packets are lost due to collision with normal packets. By keeping surrounding nodes quiet (ACM), the loss rate becomes almost zero. In the broadcast-base network (Fig. 3.10(b)), the loss rate with KA is larger than in the tree-based network, since there occur additional collisions among emergency packets traversing multiple paths. On the other hand, with ACM, although the initial value is the same as that of KA, the loss rate drops gradually in about 20 seconds. The reason why the loss rate does not become zero is that suppressing transmission of normal packets can not prevent collisions among emergency packets in a corridor. With retransmission, *i.e.*, ACM+RT, no packet loss occurs. However, the total number of emergency packets that *EMG_SEND* and *EMG_FORWARD* nodes send including retransmission is larger than that of ACM by 15 % and this increase leads to additional energy expenditure.

Delay of emergency packets

As for the end-to-end delay (Fig. 3.11), which is defined as the time taken for an emergency packet emitted by a source node to arrive at the BS, we can observe that the delay of ACM is a little smaller than that of KA. In ACM, since surrounding nodes are quiet, the number of backoff due to contention in the MAC layer is smaller than in KA [43].

The delay of ACM+RT becomes larger than the others for retransmission in the broadcast-based network as shown in Fig. 3.11(b). However, in the tree-based network (Fig. 3.11(a)), such drawback is not observed. As we saw in Fig. 3.10(a), the suppression is highly effective in the tree-based network, which means that there is little need of retransmission. On the contrary, in the broadcast-based network, there are collisions among emergency packets which incur retransmission. Retransmission is conducted in only 0.4 % cases of packet transmission in the tree-based network, while 11 % in the broadcast-based network.

Comparing Fig. 3.11(a) and Fig. 3.11(b), we can see that the delay of KA and ACM is larger in the tree-based network than in the broadcast-based network. The reason is that a random backoff is applied before packet emission in the network layer in the experiments. The BS in the broadcast-based network receives multiple emergency packets with the same sequence number which traverse different paths. Among them, the first packet which arrives at the BS is taken into account in the delay. On the other hand, in the tree-based network, there is only one path for an *EMG_SEND* node. Thus the delay in the tree-based network becomes larger than in the broadcast-based network.

Multiple emergency nodes

Next we consider cases where multiple nodes detect an emergency event and move to the *EMG_SEND* state at the same time. The loss rate of emergency packets are plotted against the number of *EMG_SEND* nodes in Fig. 3.12. The more the number of *EMG_SEND* nodes is, the more frequently collisions occur. More than 25 % of emergency packets are lost in the cases of eight *EMG_SEND* nodes without retransmission. This is because that emergency packets originated from different source nodes collide with each other in the same or merged assured corridor.

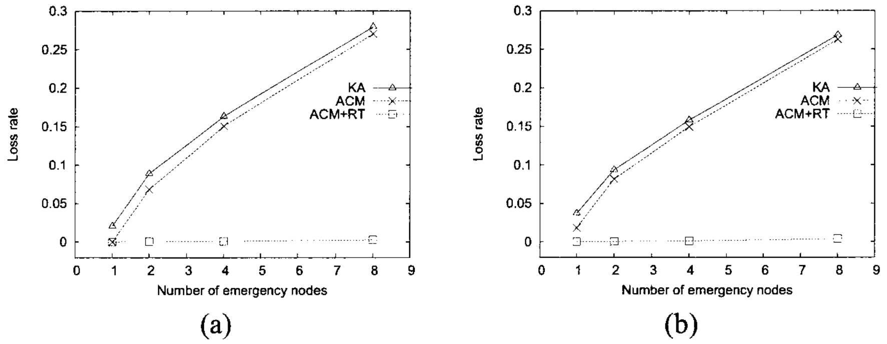


Figure 3.12: The loss rate of emergency packets in (a) a tree-based network and (b) a broadcast-based network with multiple *EMG_SEND* nodes.

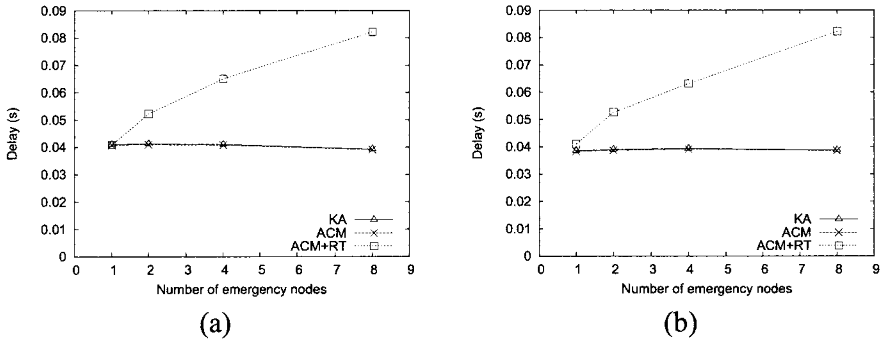
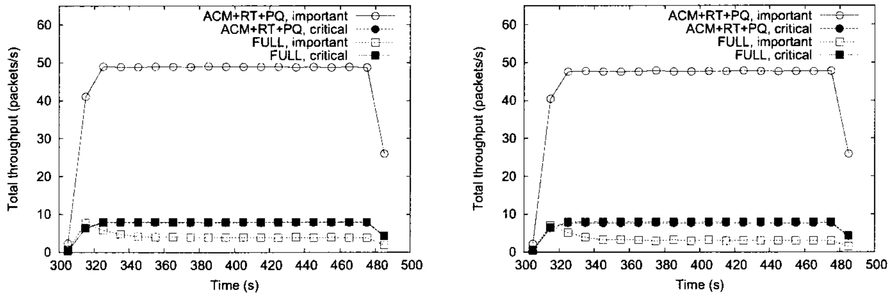


Figure 3.13: The delay of emergency packets in (a) a tree-based network and (b) a broadcast-based network with multiple *EMG_SEND* nodes.

In comparing results of KA and ACM, the effect of suppression of normal packets in reduction of loss rate becomes slightly smaller as the number of *EMG_SEND* nodes increases in both the tree-based and broadcast-based networks. With the help of retransmission, in ACM+RT, the loss rate is less than 0.3 % and 0.4 % with eight *EMG_SEND* nodes in the tree-based and the broadcast-based networks respectively.

Fig. 3.13 shows the delay of emergency packets against the number of *EMG_SEND* nodes. For KA and ACM, results are close since most of nodes are in *EMG_SEND* or *EMG_FORWARD* states and suppression of normal packets is not effective. In addition, the delay slightly decreases as the number of *EMG_SEND* nodes increases in both the tree-based and broadcast-based networks. The reason for this can be explained as follows. In calculating the delay, we take into account only emergency packets that successfully arrive at the BS. Therefore, there is a bias in favor of emergency packets emitted by *EMG_SEND* nodes closer to the BS than those of distant *EMG_SEND* nodes, for their less loss rate. For supporting this, the per-hop delay of KA and ACM, which is not shown because of space limitations, slightly increases as the number of *EMG_SEND* nodes becomes larger due to con-



(a)

(b)

Figure 3.14: The total throughput of emergency packets with (a) 200 nodes and (b) 500 nodes (large scale event).

tention in the MAC layer among more *EMG_SEND* and *EMG_FORWARD* nodes. On the contrary, the delay of ACM+RT increases with the number of *EMG_SEND* nodes reflecting more frequent retransmission due to collisions among emergency packets within a corridor.

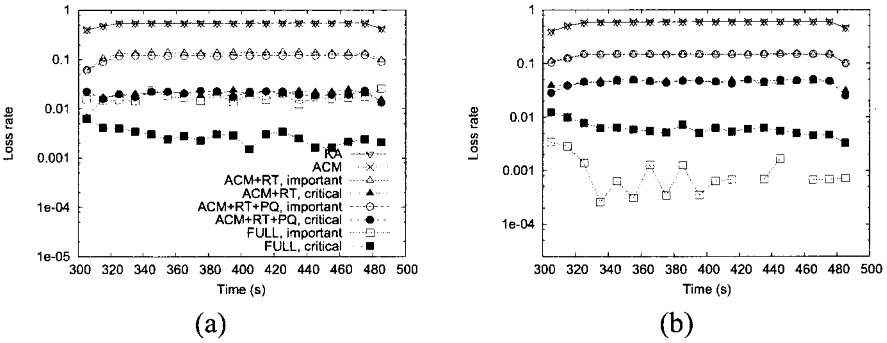
In concluding the above results, we can say that suppression of normal packets contributes reduction of the loss of emergency packets. With retransmission, most of emergency packets from a source node can reach the BS at the probability of higher than about 99.6 %. However, it is accomplished at the sacrifice of the increased delay, which is proportional to the number of *EMG_SEND* nodes, for contention among emergency packets.

3.5.2 Evaluation of UMIUSI

In order to evaluate the effect of mechanisms comprising UMIUSI, we compared five combinations of the mechanisms. In addition to the three variants of the previous subsection, the experiments of ACM+RT+PQ (priority queueing), in which the priority queueing is additionally applied, and FULL, which employs all of the mechanisms described in Section 3.2 including the rate control mechanisms, are conducted. The broadcast-based routing is employed for these experiments.

For the experiments of a small scale event with one *EMG_SEND* node, priority queueing and rate control have no effects, since there is no important class traffic. Thus the results of ACM+RT+PQ and FULL are the same as those of ACM+RT. The results of three variants, KA, ACM, and ACM+RT have been shown in Figs. 3.10 and 3.11.

We conducted another series of 100 simulation experiments with 32 emergency nodes, where four are of the critical class and the others are of the important class. After 300 seconds of initialization, each of the 32 nodes moves to the *EMG_SEND* state at a randomly chosen time within the following 10 seconds. Each of them stays in the *EMG_SEND* state for 180 seconds and then returns to the *NORMAL* state. The initial emission interval of important packets $t_{\rm imp}$ is set at 0.5 seconds



(a)

(b)

Figure 3.15: The loss rate of emergency packets with (a) 200 nodes and (b) 500 nodes (large scale event). The legends in (b) are the same as in (a).

and the emission interval of critical packets t_{cri} is fixed at 0.5 seconds. For the AIMD rate control, the parameter α in Eq.(3.1) is 0.05 and β in Eq.(3.2) is 0.5 taken from [39].

Throughput

The total throughput of the critical and important class averaged over 100 experiments is illustrated in Fig. 3.14. Here the total throughput is defined as the number of emergency packets per second received by the BS. It seems that there is no difference between the throughput of the critical class with and without the rate control. This is because the total traffic of the critical class is 8 packets/s and, as shown later, the loss rate of critical packets is as low as around 2 % even without the rate control. Similarly, little difference was observed between the throughput of ACM+RT and that of ACM+RT+PQ for both traffic classes, thus the results of ACM+RT are not shown in the figure.

In FULL of the low density scenario (Fig. 3.14(a)), the total throughput of the important class decreases to 3.8 packets/s while that in ACM+RT+PQ is 49 packets/s. In the high density scenario (Fig. 3.14(b)), the total throughput of the important class in FULL is about 80 % of that in the low density scenario reflecting the smaller sending rate due to more frequent collisions.

Loss rate

In a large scale event, a lot of nodes transmit urgent information at the same time. Here, there are two factors which affect the reliability of transmission; collision and buffer overflow. A lot of emergency packets generated at the same time can cause buffer overflow at *EMG_FORWARD* nodes, which leads to a bursty loss of important packets. Buffer overflow does not matter for the critical class, because the number of critical class nodes is limited at deployment as stated in Section 3.2.

By comparing KA and ACM on the loss rate shown in Fig. 3.15, it can be seen that suppression of normal packets has little effect, since most of collisions occur among emergency packets. The scheduled retransmission makes the loss rate of the critical class lower than that of the important class in ACM+RT of both low and high density scenarios. The rate control further reduces the loss rate of critical packets. Note that these observations support our discussion in Section 3.3 referring Fig. 3.7.

By the rate control, the loss rate of important packets also decreases in both scenarios. Since the emission interval of important packets is increased by the rate control, a node has more time to retransmit lost packets. For example, in the low density scenario, the total emission rate of 3.8 packets/s means that the averaged emission interval of important packets at a source node is 7.4 seconds. The interval is long enough for an *EMG_FORWARD* node to retransmit a packet four times, following the schedule shown in Fig. 3.4. On the other hand, without the rate control, an *EMG_FORWARD* node can retransmit a packet only once.

On the contrary, losses of critical class packets due to collisions can not be fully recovered by retransmission. Since UMIUSI does not regulate the emission rate, or the emission intervals, of the critical class, a critical class packet has a fewer number of retransmissions than an important class packet. The decrease of the loss rate of the critical class in FULL is mainly for less collisions with important packets. With the help of the rate control mechanisms, the loss rate of the critical class is reduced to about 0.25 % from around 2 % (ACM+RT+PQ) in the low density scenario shown in Fig. 3.15(a). Considering this together with the results of Fig. 3.14(a), we can say that we sacrifice 92 % of the throughput of the important class in order to make the loss rate of the critical class one-eighth.

Figure 3.15(b) shows the loss rate in the high density scenario. The high density leads to more frequent collisions. However, at the same time, the possibility of buffer overflow decreases. A packet emitted by a node would be received at any one of several next-hop nodes. On receiving an acknowledgement, the node can discard the packet and the buffer occupancy decreases. This is the reason why the loss late of the important class in FULL is much lower in the high density scenario than in the low density scenario while they are almost the same without the rate control. On the other hand, the negative effect of increased collisions affects the critical class in the high density scenario, since there is no overflow in both of the low and high density scenarios.

Delay

Figure 3.16 shows the delay of emergency packets in a large event. First, the suppression of normal packets does not help since the number of *NORMAL* nodes is small. Next, the scheduled retransmission increases the delay of both classes, but offers the differentiated service. The priority queueing has little effect in our simulation experiments. It is expected to contribute more under a scenario with bursty important class traffic such as image or sound. The rate control mechanisms decrease the delay of the critical class traffic to about 70 ms. The reason for the large variation of FULL is that some packets occasionally experienced delay of

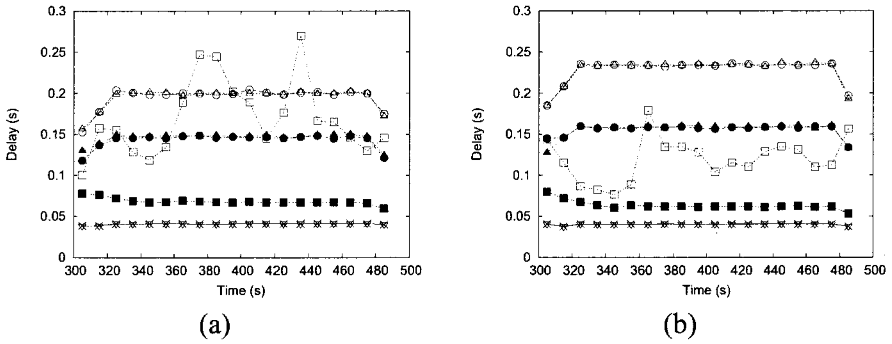


Figure 3.16: The delay of emergency packets with (a) 200 nodes and (b) 500 nodes (large scale event). The legends are the same as in Fig. 3.15(a).

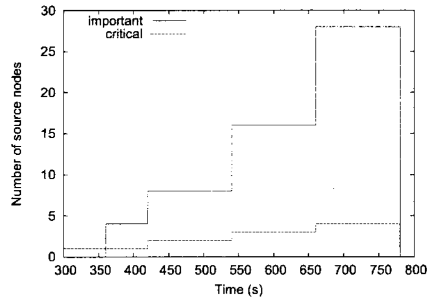


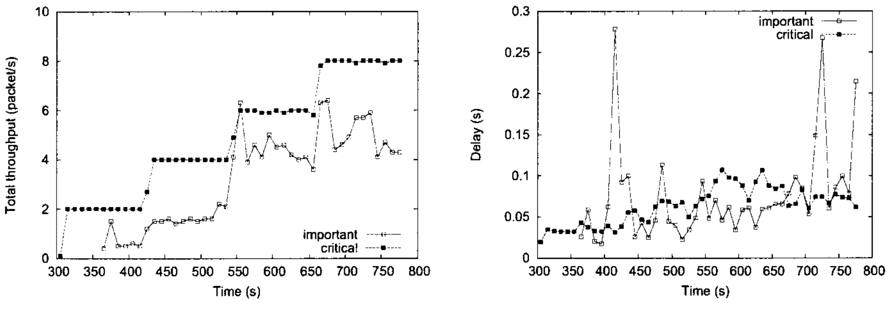
Figure 3.17: The expanding emergency scenario.

more than 10 seconds for the long emission interval.

3.5.3 Expanding emergency

In order to evaluate UMIUSI in a transient situation, we consider another scenario of an event whose scale is initially small, but develops to be larger one as time passes, such as a fire. The number of source nodes of emergency packets in this scenario is illustrated in Fig. 3.17. One critical class node first detects an event at $t = 300$ seconds. Three important class nodes detect the event at $t = 360$ seconds, and after that, some of other nodes become source nodes of emergency packets in addition at $t = 540$ and 660 seconds. They go back to the *NORMAL* state at $t = 780$ seconds. The low density layout is employed.

The total throughput of emergency packets is shown in Fig. 3.18(a). We observe that the total throughput of the critical class is retained at 2 packets/s per node, regardless of the number of source nodes of the important class. The delay of critical packets (Fig. 3.18(b)) is also maintained relatively constant throughout this scenario. The loss rates of critical and important packets are 0.3 % and 0.07 % respectively. We can see that UMIUSI keeps the loss rate and delay of critical packets relatively low while the emergency is dynamically expanding.



(a) (b)

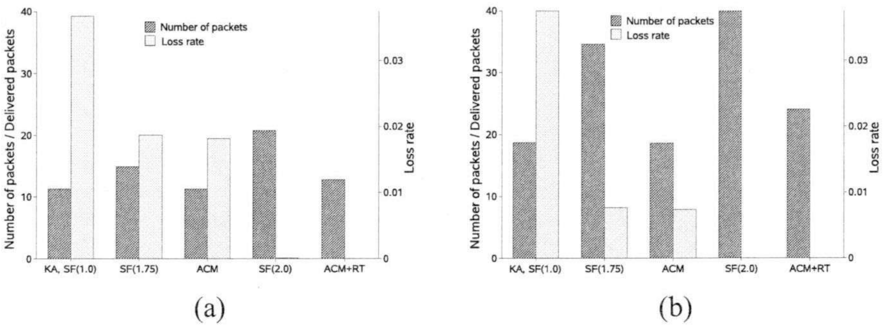
Figure 3.18: The (a) total throughput and (b) delay of emergency packets in the spreading event scenario.

3.5.4 UMIUSI and stochastic forwarding

To compare the performance of UMIUSI with that of the stochastic forwarding, the number of emergency packets needed to deliver one emergency packet to the BS, which is calculated by dividing the total number of packet emission at *EMG_SEND* and *EMG_FORWARD* nodes by the number of emergency packets successfully delivered to the BS, is shown in Fig. 3.19 as well as the loss rate.

The stochastic forwarding scheme with the forwarding probability of one, denoted as SF(1.0), is equivalent to KA. In the low density scenario, by increasing the forwarding probability to 1.35, denoted as SF(1.35), the stochastic forwarding scheme can attain the same level of reliable transmission of emergency packets as ACM. However, it involves more packets than ACM by 33 %. Furthermore, SF(2.0) puts more load on a WSN than ACM+RT by 84 % to avoid the loss of emergency packets. The delay of emergency packets in SF(1.35) is larger than ACM and comparable to ACM+RT, and the delay in SF(2.0) is about 7 % larger than ACM+RT (not shown in the figure). Similar results are obtained in the high density scenario.

For a large scale event with 32 *EMG_SEND* nodes, the loss rate of emergency packets in FULL is 0.6 % and 0.1 % for the critical and important class respectively in the high density scenario. However, the loss rate of the stochastic forwarding is at its minimum of 13.9 % with the forwarding probability of 2.1, and it can not offer the same level of reliability as UMIUSI does. Increasing the forwarding probability merely results in heavier congestion, and decreasing it also leads to larger loss rate due to less chance of retransmission. This result supports our claim that a single comprehensive mechanism can not adapt to the variation of the scale of an event.



(a)

(b)

Figure 3.19: The total number of packet emission and the loss rate with (a) 200 nodes and (b) 500 nodes (small scale event).

Table 3.1: The power consumption of nodes during 10 minutes. (mAh)

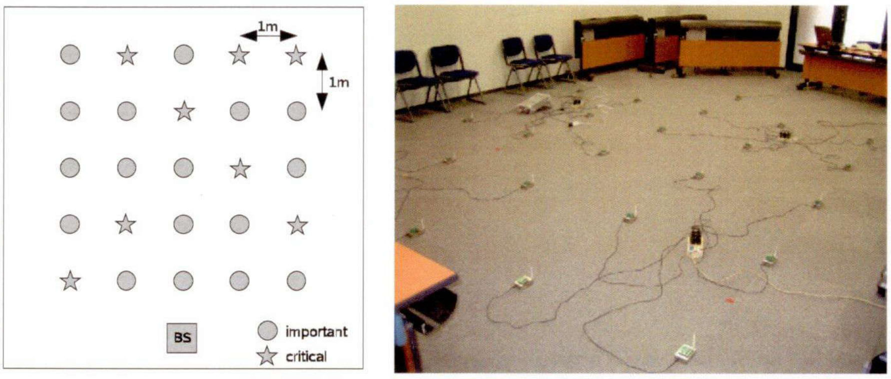
State	Node 1	Node 2	Node 3
NORMAL	8.786	8.814	8.759
EMG_FORWARD	8.795	8.834	8.778
EMG_SEND	8.795	8.836	8.779
SUPPRESSED	8.804	8.849	8.783

3.6 Practical Experiments

Following the simulation experiments, we implemented UMIUSI onto off-the-shelf sensor nodes provided by OKI Electric Industries Co., Ltd. to conduct practical experiments using two testbeds. Testbed A consisted of 26 nodes including a BS which were put in a 10 m × 6 m room. In Testbed B, 47 nodes were deployed over a floor of a building for feasibility demonstration.

In the experiments, IEEE 802.15.4 non-beacon mode was employed for the MAC layer. The payload size of an emergency packets was 16 bytes including packet header with a class identifier, dummy sensor data, and a time stamp. For the network layer, we adopted the synchronization-based data gathering scheme [44] modified to ignore uni-directional links. It employs a tree-based routing, and timing of packet emission is the same among nodes of the same hop distance from the BS. In a normal state, all nodes adopt a sleep schedule. Nodes on the same hop distance wake up at the same time and receive packets from one-hop distant node. Then, they send packets to next-hop nodes. Finally, after overhearing packets emitted by the next-hop nodes, they go back to a sleep mode. In the experiments, we set the interval of normal packet emission t_{norm} at 10 seconds, and the offset between emissions of adjacent nodes at 1 second. Routes from nodes to the BS dynamically changed for variations in radio environment.

For evaluation of transmission of emergency packets, we made one (small scale event) or eight (large scale event) nodes detect an event and moved to the *EMG_SEND* state. Each of them was scheduled to emit emergency packets at an



(a) (b)

Figure 3.20: (a) A schematic view and (b) a photograph of Testbed A.

interval of 0.5 seconds, but the actual interval was about 0.58 seconds due to implementation constraints. Source nodes went back to normal operation with the sleep schedule 10 minutes later.

Table 3.1 summarises the amounts of power consumed by randomly chosen three nodes in 10 minutes for each of the four states of ACM in a small scale event on Testbed A. It was measured by a digital power meter (Yokogawa Electric WT210) attached to the DC input of a node. We find that the difference in power consumption among four states is relatively small compared to the consumed power. Therefore the active-to-sleep ratio in normal operation would determine the lifetime of a node. A node adopts three AAA alkaline cells with serial connection. One cell has the capacity of about 1 Ah. Thus, if we apply a sleep schedule of the active-to-sleep ratio of 1/600, *e.g.*, being active for one second in ten minutes, the lifetime of a node can be estimated as 11,400 hours (= 1.30 years). Once an emergency occurs and a node stays in either of *EMG_SEND*, *EMG_FORWARD* or *SUPPRESSED* states for 3 minutes for example, it shortens the lifetime of a node by 30 hours. Developing a sleep schedule for these states is one of our future works.

3.6.1 Fundamental experiments

In Testbed A, 25 sensor nodes were arranged in a 5×5 grid topology with separation of 1 m (Fig. 3.20). A BS was put beside the grid with 1 m separation. The transmission power was set to -27 dBm. The average delivery ratio of normal packets over 10 hours experiments was around 80 %.

In order to evaluate the effect of mechanisms comprising UMIUSI, we compared five variants of combination of the mechanisms as well as in the simulation experiments. For the variants with retransmission, the first retransmission was scheduled at 0.1 seconds after the first transmission for the critical class and 0.2 seconds for the important class, respectively. A binary backoff scheme was ap-

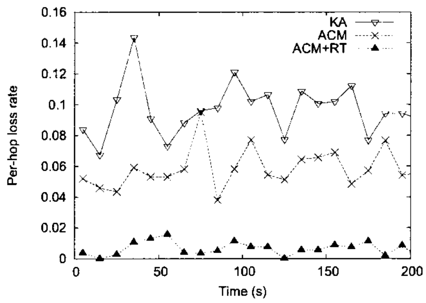


Figure 3.21: The per-hop loss rate of emergency packets (small scale event).

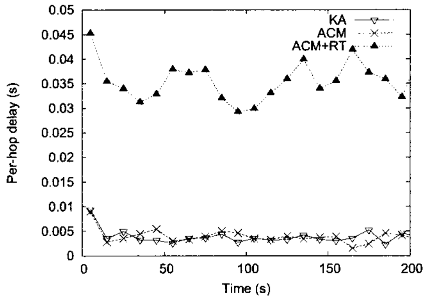


Figure 3.22: The per-hop delay of emergency packets (small scale event).

plied to following retransmissions. In FULL, local congestion detection was done by monitoring packet reception rate at each node. When a node received more than 20 packets in recent 2 seconds, it considered that the wireless channel was highly loaded. We observed that the number of packet losses rose up sharply when the traffic exceeded this threshold in preliminary experiments. For the AIMD rate control, the parameters for multiplicative decrease and additive increase were the same as in the simulation experiments.

Since the hop distance from a node to the BS dynamically changed during the experiments, we employ the per-hop loss rate and per-hop delay as evaluation metrics. Letting n the hop distance from a source node to the BS and p_k $(k = 1, 2, \cdots, n)$ the per-hop loss rate in transmission of emergency packets at k-th hop, the loss rate P_n observed at the BS is given by

$$P_n = 1 - Q_n = 1 - \prod_{i=1}^{n}(1 - p_i)$$

where Q_n is the delivery ratio observed at the BS. In the experiments, packet losses were detected at the BS using a sequence number in the header of an emergency packet to obtain P_n. Then, assuming p_k is identical for all hops $(p_1 = p_2 = \cdots = p_n = p)$, the per-hop loss rate p is defined as

$$p = 1 - \sqrt[n]{1 - P_n} = 1 - \sqrt[n]{Q_n}.$$

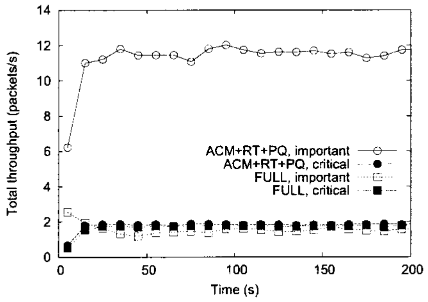


Figure 3.23: The total throughput of emergency packets (large scale event).

The per-hop delay d is defined as

$$d = D_n / n,$$

where D_n is time taken from emission of an emergency packet at a source node to reception of the packet at the BS. Note that, since only a clock on the hundred milliseconds scale was provided for the application layer, time error in D_n observed was 100 ms at maximum.

Small scale event

In the scenario of a small scale event, one critical class node became a source node of critical packets and went back to normal operation 10 minutes later. We conducted experiments twice for each of randomly chosen eight nodes.

Figure 3.21 shows the per-hop loss rate of emergency packets averaged over the 16 experiments. In these experiments, suppression of normal packets became effective immediately after a node began emitting critical packets, since the hop distance to the BS was so small, 1.6 on average, that it did not take long to establish an assured corridor. With retransmission, the per-hop loss rate was further improved to be less than 1 %.

However, retransmission led to increase of the per-hop delay (Fig. 3.22). Even if collision was mostly avoided by the suppression, packet losses frequently occurred (see ACM in Fig. 3.21) due to random channel errors. Therefore, a number of retransmissions were needed to recover those lost packets, which resulted in increase of the delay.

Large scale event

For the experiments of a large scale event, we considered eight sets of seven important class nodes and one critical class node. These eight nodes were moved to the *EMG_SEND* state in about 10 seconds and went back to the *NORMAL* state about 10 minutes later. We conducted experiments twice for each of the eight sets.

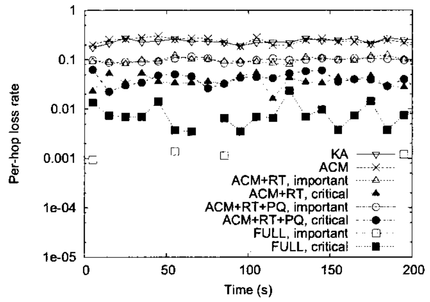


Figure 3.24: The per-hop loss rate of emergency packets (large scale event).

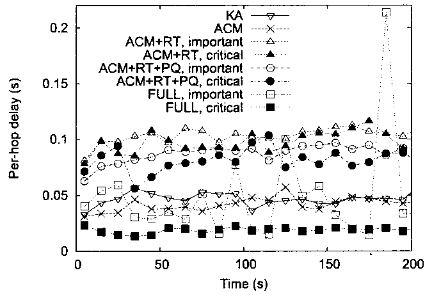


Figure 3.25: The per-hop delay of emergency packets (large scale event).

First, the total throughput of the critical and important class averaged over the experiments is illustrated in Fig 3.23. Little difference was observed between the total throughput of ACM+RT and that of ACM+RT+PQ for both classes, thus results of ACM+RT are not shown in Fig. 3.23. With the rate control mechanisms, we can see that the total throughput of the important class decreased around 1.5 packets/s in 30 seconds while that without the rate control in ACM+RT+PQ was kept high at 11.5 packets/s.

The per-hop loss rate of emergency packets is illustrated in Fig. 3.24. Suppression of normal packets had little effect on reliability in a large scale event as can be seen in comparison between KA and ACM. Adding the priority queueing mechanism to ACM+RT was also not much helpful in this experiment settings, because paths of important and critical class traffic seldom overlapped with each other. In FULL, the per-hop loss rate of the critical class gradually decreased as the emission rate of important class was regulated by the rate control mechanisms. Since an important class packet had more chances to be retransmitted than a critical class packet due to the prolonged interval of emission by rate control, the per-hop loss rate of the important class was smaller than that of the critical class.

Figure 3.25 shows the per-hop delay of emergency packets. In FULL, the per-hop delay of the critical class gradually decreased in the first 30 seconds as the important class traffic was regulated by the rate control mechanisms. The emission interval of important class packets was prolonged by the rate control mechanisms,

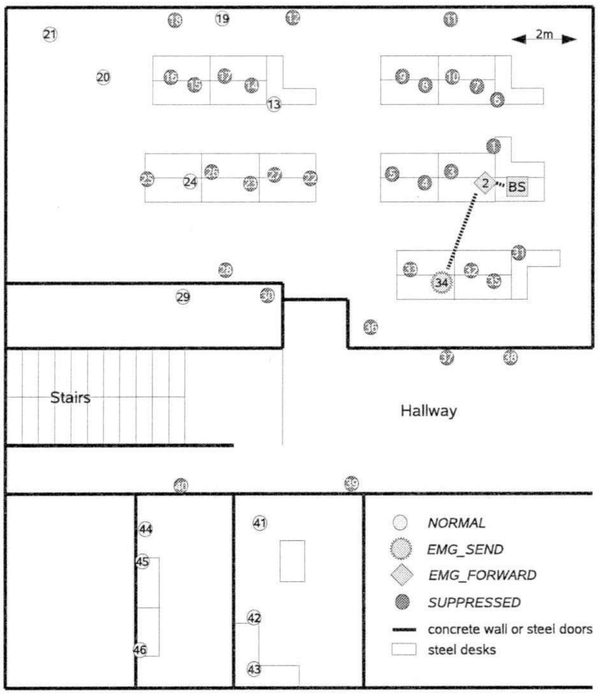


Figure 3.26: A small scale event in Testbed B.

thus waiting time to be retransmitted at an *EMG_FORWARD* node could be as long as a few seconds. Such occasional large delay caused the large variation in the per-hop delay of the important class in FULL.

3.6.2 Feasibility demonstration

The purpose of the experiments in Testbed B is to verify feasibility of UMIUSI in practical settings. In Testbed B, 46 sensor nodes and a BS were deployed on a floor of several rooms and a hallway in a concrete building, as illustrated in Fig. 3.26. All of the five mechanisms of UMIUSI (FULL) were used, and parameters were the same as in Testbed A other than the transmission power of -7 dBm.

Small scale event

Figure 3.26 shows a snapshot where Node 34 detected an emergency. Node 34 reached the BS via Node 2 by a path established by the synchronization-based data gathering scheme. Therefore, Node 2 was in the *EMG_FORWARD* state. Nodes indicated by dark circles could hear radio signals of Nodes 2 and 34 and moved to the *SUPPRESSED* state. Due to shadowing and fading, the geographical proximity does not necessarily correspond to neighbor relation. The average per-hop loss rate over eight experiments setting Nodes 4, 7, 16, 25, 30, 34, 41, and 46 an *EMG_SEND* node was 0.37%. The average hop distance to the BS was 2.75.

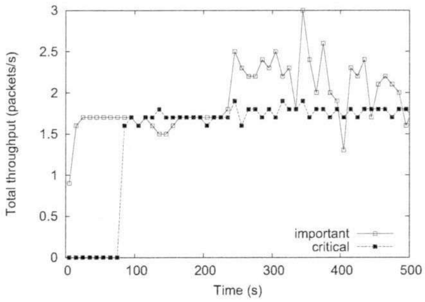


Figure 3.27: The total throughput of emergency packets in Testbed B.

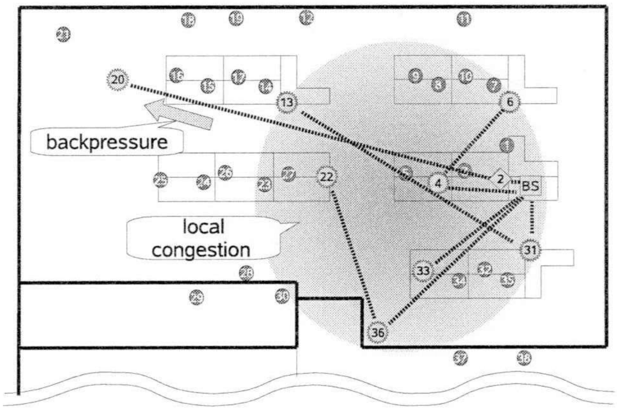


Figure 3.28: A snapshot in the large scale event in Testbed B ($t = 400$ s).

Large scale event

Next we considered a scenario where a small scale emergency grew to become large, such as a fire. In this scenario, Node 33 first detected an event, moved to the *EMG_SEND* state, and began sending important packets at time $t = 0$. At $t =$ 80 seconds, Node 4 of the critical class next detected the event, followed by Nodes 22 and 36 at $t = 240$ seconds and Nodes 6, 13, 20, and 31 at $t = 340$ seconds. These six nodes were of the important class.

When Nodes 4 and 33 were in the *EMG_SEND* state, emergency packets were directly transmitted to the BS. As shown in Fig. 3.27, the average throughput was about 1.7 packets/s for both nodes. After Nodes 22 and 36 detected the event at $t = 240$ seconds, local congestion was detected and the three *EMG_SEND* nodes of the important class, *i.e.*, Nodes 33, 22, and 36, reduced the emission rate in order to mitigate congestion. The total throughput of the important class was controlled around 2.3 packets/s. At $t = 340$ seconds, other four nodes newly began to emit important packets and this caused congestion around Node 2. To reduce the important class traffic at its source, Node 2 sent a backpressure to Node 20 (Fig. 3.28). As a result, the total throughput of the important class was kept about the same level, and there was no loss of critical class packets throughout this ten minutes experiment. The loss rate of important class packets was 0.6 %.

As shown in this experiment, traffic control developed from path-level to network-level adapting to the growing scale of emergency without any centralized control in UMIUSI.

3.7 Conclusion

Urgent sensor information is needed to be transmitted preferentially in a WSN used as a social infrastructure. In this chapter, we presented a WSN control protocol for fast and reliable transmission of urgent information. In this control protocol, Sensor information is categorized into three traffic classes: normal, important, and critical. In order to prioritize transmission of the critical class, five simple fully-distributed mechanisms, *i.e.*, assured corridor mechanism (ACM), retransmission, priority queueing, rate control by local congestion detection, and rate control by backpressure, working in different spatial and temporal levels are installed onto each node and the collective control of these mechanisms offers preferential transmission of urgent information adapting to the scale of emergency. In ACM, the path of emergency packets is protected from collisions with normal packets by making surrounding nodes refrain from sending normal packets. We verified through simulation and practical experiments that UMIUSI successfully improved the delivery ratio and the delay of emergency packets regardless of the scale of emergency.

Chapter 4

A Sensor Network Protocol for Automatic Meter Reading

4.1 Overview

In this chapter, we focus our attention on a WSN deployed for automatic meter reading (AMR) in a large-scale apartment building, in which there are hundreds of apartments. In such a building, meters are attached to pipes and cables in each apartment to monitor consumption of water, gas, electricity, and so on. A water, gas, or electricity company hires and sends personnel to houses and buildings to collect meter readings once a month in most cases.

In these days, those companies consider to adopt a WSN for meter reading, since it is costly to hire many personnel to cover all houses and buildings of customers and it becomes difficult for outsiders to enter modern apartment buildings for security reasons. Such a WSN for automatic meter reading consists of meters equipped with a radio transceiver operated on battery power supply and a gateway server, *i.e.*, BS, connected to a monitoring station of a company through a regional wired or wireless network. Since a meter is usually stored in a meter box or pipe shaft, radio signal is heavily disturbed and the range of transmission is relatively small. Therefore, a WSN is sparse, where the average number of neighboring nodes, *i.e.*, nodes in the range of radio signals, is a few. Monthly meter reading does not put too strict restriction on the delay requirement as far as meter reading is collected from all meters in a building. In addition to usual metering, some of latest meters are capable of detecting an abnormal event, such as gas leakage. On the contrary to usual meter reading, such urgent information must be transmitted to the BS immediately, *e.g.*, within 10 seconds, once an abnormal event happens.

In this chapter, we consider a protocol for AMR, which can collect meter reading from all meters at predetermined intervals, *e.g.*, once a month, and transmit urgent information from a meter to a BS within the specified delay bound, *e.g.*, 10 seconds. We assume that internal timers of nodes are synchronized by a certain synchronization protocol, such as proposed in [45, 46, 47, 48, 49]. In addition, a

WSN is maintained by a certain topology control protocol, together with a health check mechanism conducted once a day, for example. Our focus is rather on sleep and transmission scheduling for energy-efficient operation of a WSN. A low duty cycle leads to the longer lifetime of a WSN, where nodes rarely wake up to send and receive packets. However, since a node has to wait for the next-hop node to wake up in order to send a packet, unorganized and unscheduled sleeping causes unacceptable end-to-end delay for urgent information transmission. We aim to satisfy both requirements of bounded end-to-end delay and low duty cycle.

In the rest of this chapter, we refer a parent or child node as a direct neighbor of a node which is one hop closer or further to the BS than the node itself, respectively. A next-hop node is one of parent nodes to which a node sends a packet and a preceding node is one of child nodes from which a packet is received. The parent-child relationship is determined by a routing or topology control protocol.

4.2 Detailed Description of Sensor Network Protocol for Automatic Meter Reading (SNPAMR)

As mentioned above, making nodes sleep is one of the primary techniques to save energy consumption, but it increases delay. Sleep scheduling is one possible way to cope with this problem. By coordinating awake periods of nodes beforehand, we can have a node send a packet without unnecessarily waiting for its next-hop to become active. To detect an event and notify a BS of it, a node has to wake up at least once per the delay bound, denoted as $d_{\max}$ hereafter. Then, all nodes on the path from the detecting node to the BS must wake up at appropriate timing so that the urgent information is relayed to the BS immediately. In most of TDMA-based scheduling, the duration is divided into time slots and a slot is assigned to each node. A node can occupy the wireless channel during the assigned slot and send a packet without being disturbed by other nodes. This scheduling enables efficient usage of the wireless channel, but it is inefficient from an energy point of view. Since a next-hop node does not know which preceding node has a packet to send, it must be awake and listen to the wireless channel in all of slots assigned to its preceding nodes. In a WSN, such idle listening is the major drain of energy, especially in an AMR where traffic is normally generated only once per day for health check and once per month for meter reading.

Our approach to shorten this idle listening is to assign time slots not for sending but for receiving. In our sensor network protocol for automatic meter reading (SNPAMR), every node is active during t_s at an interval of $d_{\max}$ for possible packet reception. The total number N of slots is given by

$$N = d_{\max}/t_s. \tag{4.1}$$

We define slotID $0 \le k \le N-1$ so that a smaller slotID corresponds to an earlier slot in a cycle of $d_{\max}$. The duration t_s of a time slot is determined to be long enough for MAC layer to deliver a packet to the next-hop including carrier sense, MAC level acknowledgement, and retransmissions. For example, its typical value would be between 100 ms and 200 ms for IEEE 802.15.4 and smaller for IEEE 802.11. If t_s = 100 ms and $d_{\max}$ = 10 seconds, the duty cycle is 1/100. Even lower duty cycle can be achieved by employing an energy-aware low duty MAC protocol such as [8]. In SNPAMR, a further node from a BS is assigned an earlier slot with a smaller slotID. As far as every node has a next-hop node which has a slot of a larger slotID, a packet originating at any node can reach the BS within $d_{\max}$.

The details of SNPAMR are as follows:

1. When node i is newly deployed in a WSN, it tries to discover neighbor nodes by a neighbor discovery protocol employed. We do not discuss the details in this thesis.

2. When node i discovers neighbor j, node j notifies node i of its ID, level l_j, which corresponds to the hop distance from the BS, and slotID k_j. We assume that the BS has a power supply and does not sleep. So that a level 1 node can choose slotID ranging from 0 to $N-1$, the BS declares slotID N. The level l_j of the BS is 0. Note that node i may discover two or more neighbors. Node i stores the ID, level, and slotID of neighbors in its neighbor table. Neighbors are listed in ascending order of the level as the first key and the slotID as the second key (see Fig. 4.1). A time synchronization process could be conducted in this stage.

3. After completing the neighbor discovery process, node i determines, based on the neighbor list, its own level l_i and slotID k_i by

$$l_i = l^1 + 1, \tag{4.2}$$

$$k_i = f(k^1), \tag{4.3}$$

   where l^1 and k^1 are the level and slotID of the node at the top of the neighbor list. $f(k)$ is called the slot assignment function (SAF), which gives a smaller value than k to assign an earlier time slot to node i than its parent nodes. The slotID is determined following the slot assignment probability distribution function (SAPDF), which will be discussed later. Entries of nodes with level $l_i + 1$ and nodes with level l_i and slotID equal or smaller than the maximum among level $l_i - 1$ nodes, *i.e.*, parent nodes, are removed from the neighbor table. After determining its level l_i and slotID k_i, node i sends these information to its neighbor nodes of the same level at their slots. A neighbor node which receives this time slot notification adds an entry to its neighbor table if slotID in the notification is greater than the maximum among those of parent nodes.

4. Once node i determines its time slot, it operates in accordance with a sleep schedule. Node i wakes up at slotID k_i, keeps active for t_s to receive packets from its preceding nodes, and goes back to sleep. If it receives a packet

during this slot or it has a packet to send, it wakes up again at the next-hop's time slot, *i.e*, k^1, and sends the packet. If it has both a packet of meter reading and an event notification packet, the event notification packet is sent first (priority queueing). We assume that an acknowledgement and retransmission process is conducted in the MAC layer.

5. If node i fails in transmitting a packet at the time slot assigned to the first next-hop node, node i tries sending the packet at slotID k^2 for the second next-hop node in the neighbor list. If the transmission fails again, it tries third one. It repeats this procedure until the transmission succeeds or fails at the last next-hop node in the neighbor table.

The reason why we choose a parent node with the minimum slotID as the first next-hop is that we can have more chances to send by doing so. To have even more chances, it is possible to list neighbors in ascending order of slotID, ignoring their level. However, it implies that a node tries a neighbor of the same hop distance at an earlier timing and it leads to the larger number of hops to the BS and lower reliability. The detailed discussion on this aspect can be found in [36] and the method proposed to select a next-hop can be incorporated into our protocol.

In SNPAMR, only a sender and a receiver are active during packet transmission, unless the identical time slotID is assigned to two or more neighboring nodes in their vicinity. In this sense, the path from a source of urgent information to the BS is protected from the interference of surrounding nodes. Considering network-level contention control discussed later in the next subsection, we have four mechanisms which work in different spatial levels, which is compliant with our design methodology stated in Chapter 2: node-level priority queuing, neighbor-level retransmission, path-level sleep scheduling, and network-level contention control, in SNPAMR.

Now, we are going to demonstrate a sample behavior of SNPAMR based on Fig. 4.1 where $N = 100$. Initially, the level and slotID of nodes are set to infinity on deployment, and those of the BS are 0 and 100, respectively. First, node A conducts neighbor discovering and find the BS, nodes B and C. Since nodes B and C are not initialized yet at this time (infinite level and slotID), node A ignores them. The level and slotID of the BS, (0, 100), are sent to node A, then it sets its own level to 1, and determines its slotID according to the employed SAF. Here, let us assume that the slotID of node A is set to 94. Next, node B discovers neighbor nodes, the BS and node A, and they notify node B of their levels and slotIDs. Node B stores these information into its neighbor table in ascending order of the level. Node B determines its level and slotID from those of the BS, which is at the top of the neighbor table, as 1 and 98, respectively. The entry of node A in node B's neighbor table is discarded at this time, because node A's slotID 94 is smaller than that of the parent node, *i.e.*, the BS. Node B notifies node A of its slotID, but node A also does not add an entry of node B to its neighbor table for the same reason. Now, node C discovers its neighbor nodes A and B, and it determines its own level and slotID, 2 and 91 respectively, from those of node A at the top of the table.

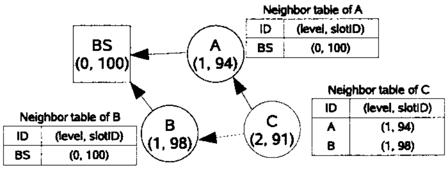


Figure 4.1: An example of level and slotID assignment in SNPAMR.

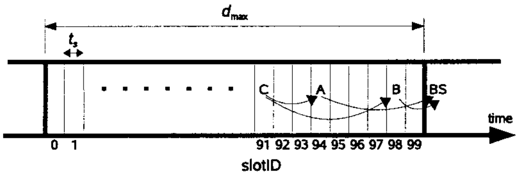


Figure 4.2: Scheduled packet transmission.

Once the level and slotID are determined, each node undergoes the sleep schedule. If node C has data to send for detecting an event, reception of a packet from its preceding node, or reading the meter, it sends the data to node A at slotID $k = 94$ when its first next-hop node A is active (Fig. 4.2). If the transmission fails, it tries to send the data to B at $k = 98$. One may think that the BS could be heavily congested because all level 1 nodes try sending a packet at $k = 100$, which corresponds to $k = 0$ of the next cycle. We can avoid this problem, for example, by giving level 1 nodes even slotIDs and having them transmit a packet at the following time slot of an odd slotID, under a reasonable assumption that the BS has a power supply and does not sleep. In the example of Fig. 4.2, node A and B can send a packet to BS at $k = 95$ and 99 respectively.

Note that, in SNPAMR, sleep scheduling of nodes and routing are integrated in terms of the time slot assignment. Each node utilizes only its neighbor's information, thus this protocol is fully-distributed and self-organizing.

4.3 Contention Degree

4.3.1 Definition

In SNPAMR, it is possible that two or more nodes transmit packets at the same time, since time slots are not for transmission but for reception. Moreover, since each node determines its own slotID in a distributed manner, two or more nodes could have the same slotID. Such duplicated slot assignment increases the possibility of simultaneous transmission and contention for wireless channel among multiple nodes. In addition, MAC layer could not finish delivery of a packet within t_s. These could degrade the reliability of transmission and lead to extra energy consumption in the MAC layer. Therefore, we need to take into account the degree of contention, more specifically, balance the degree of contention among slots, in

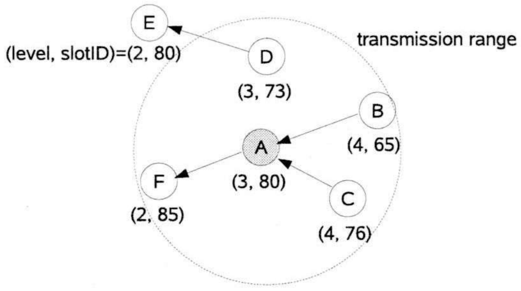


Figure 4.3: An example of the contention degree.

determining SAPDF.

In order to evaluate the intensity of contention at a node, we define the contention degree C of a node as the number of neighbors which can transmit a packet at the time slot given to the node. In other words, if a node is assigned slotID k, C is the number of neighbors which have a next-hop assigned slotID k. Note that C is not necessarily equal to the number of its preceding or child nodes, because C of a node may include its neighbors whose next-hop is another node having the same time slot. For example, in Fig. 4.3, slotID $k = 80$ is assigned to node A, whose next-hop is node F with $k = 85$. The next-hop of nodes B and C is node A, in other words, nodes B and C are preceding nodes of node A. Thus these two nodes can transmit a packet at $k = 80$. Meanwhile, node D, which is the same level as node A, has its next-hop node E to which $k = 80$ is assigned. Thus node D can also transmit a packet at $k = 80$. Therefore, node A has three direct neighbors which can send a packet at its time slot, namely nodes B, C, and D, and consequently its contention degree is 3. While we consider only the first next-hop to compute the contention degree in the following discussion, it is straightforward to expand this idea to involve the second and more next-hops by weighing contribution to the contention degree, 1.0 for the first next-hop and 0.1 for the second next-hop for example.

4.3.2 Contention degree in a grid network

First, we calculate the expected contention degree in a grid network, considering the arrangement of apartments in a building. In a grid network shown in Fig. 4.4, each circle represents a node and a number inside denotes its level. In this case, the BS is located at the center. Node B has two child nodes, X·and Y. Thus, the contention degree of B, C_B, will be either 0, 1, or 2. Nodes X and Y determine their next-hop in accordance with a smaller slotID among nodes A and B, and B and C, respectively. Therefore, C_B is determined by comparing the slotIDs of nodes A, B, and C. For example, for $k_A > k_B > k_C$, $C_B = 1$, since node X chooses node B and node Y chooses node C as their first next-hop.

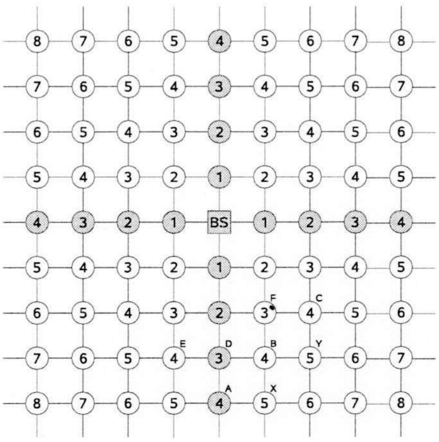


Figure 4.4: A grid network.

Now letting q the probability that two nodes of level l have the same slotID,

$$P(k_A = k_B) = P(k_B = k_C) = q, \tag{4.4}$$

then,

$$P(k_A < k_B) = P(k_A > k_B) = P(k_B < k_C) = P(k_B > k_C) = (1 - q)/2. \tag{4.5}$$

Therefore,

$$P(k_A > k_B > k_C) = P(k_A > k_B)P(k_B > k_C) = \{(1 - q)/2\}^2. \tag{4.6}$$

Calculating the probability of all nine combinations of k_A, k_B, and k_C and their probabilities, the expected contention degree of node B, $E(C_B)$, is given by

$$E(C_B) = 1 + q.$$

Next we consider nodes represented by a gray circle in Fig. 4.4 with three child nodes and one parent node. For example, node D has child nodes A, B, and E. We can apply similar discussion as above for nodes B and E. In addition, node D is the only parent of node A, so node A has node D as its next-hop. Therefore, the expected contention degree of node D, $E(C_D)$, is given by,

$$E(C_D) = 2 + q.$$

We refer a gray node as "m-node" (main stream node) and an open-circle node as "b-node" (branch stream node) in the rest of this chapter. Since the discussion above on C_A and C_B can be applied for all m-nodes and b-nodes, respectively, we define the expected contention degree of m-nodes, $E(C_m)$, and of b-nodes, $E(C_b)$, as

$$E(C_m) = 2 + q, \tag{4.7}$$

$$E(C_b) = 1 + q, \tag{4.8}$$

If a BS is located at the corner of the network, $E(C_m)$ and $E(C_b)$ are defined as

$$E(C_m) = \frac{3+q}{2}, \tag{4.9}$$

$$E(C_b) = 1 + q, \tag{4.10}$$

in the same way.

4.4 Slot Assignment Probability Distribution Functions (SAPDF)

4.4.1 Evaluation metrics

A larger contention degree means more contention, which leads to unreliable transmission and more energy consumption. Therefore we want to have the contention degree as low as possible. In addition, in order to equalize energy consumption among nodes, contention degree should be similar among nodes. In SNPAMR, a node determines its slotID from that of the first next-hop node in a stochastic way by a slot assignment functions (SAF) $f(k)$. In the rest of this chapter, we explore SAFs which give lower and more homogeneous contention degree over a network. We employ following four evaluation metrics for SAFs.

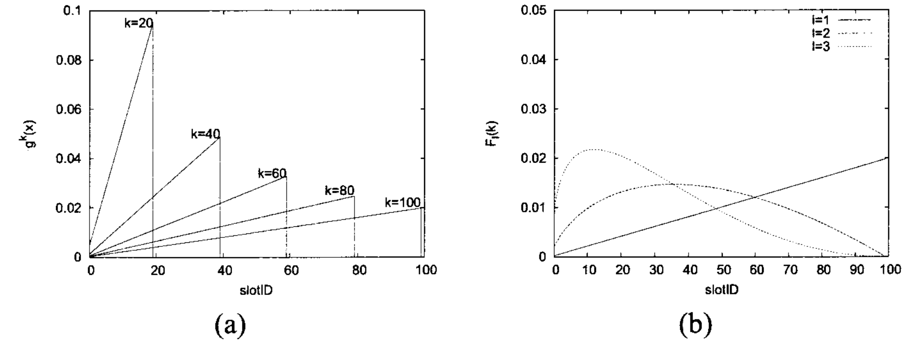
- $E_l(C)$. The average contention degree of level l nodes. Lower $E_l(C)$ means less contention.
- $V_l(C)$. The variance of the contention degree of level l nodes. Lower $V_l(C)$ means more homogeneous contention.
- p_{empty}. The ratio of unassigned time slots. Lower p_{empty} means more effective utilization of time slots.
- q_{isolated}. When all of parent nodes are assigned a time slot of slotID 0, a node cannot obtain its time slot. Such node is called an isolated node. Since slot assignment is done from the center of a network toward the edge, nodes at the edge tend to be isolated. q_{isolated} is the ratio of isolated nodes to all nodes in a network.

4.4.2 Examples of SAPDF

In this subsection, we consider the slot assignment probability distribution function (SAPDF) $g^k(x)$ which determines SAF $f(k)$. SAPDF $g^k(x)$ gives the probability that slotID x is assigned to a node when its first next-hop's slotID is k.

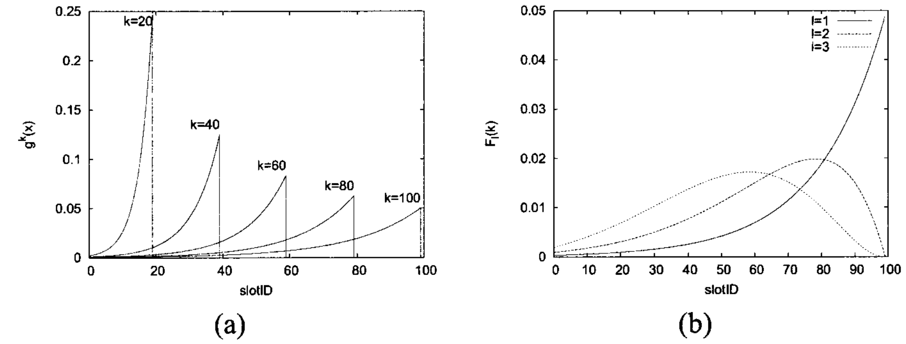
Letting $F_l(x)$ denote the probability distribution of nodes (PDN) that a node of level l is assigned slotID x,

$$F_1(x) = g^N(x), \tag{4.11}$$



(a) (b)

Figure 4.5: The (a) SAPDF and (b) PDN of LINEAR.



(a) (b)

Figure 4.6: The (a) SAPDF and (b) PDN of EXPONENTIAL.

$$F_{l+1}(x) = \sum_{k=0}^{N-1} g^k(x)F_l(k). \tag{4.12}$$

Note that this expression ignores, for simplicity, the process of choosing the next-hop node with the minimum slotID. Therefore, it corresponds to choosing a next-hop randomly among parent nodes.

We show four typical examples of a SAPDF and investigate the characteristics of each function in a grid network.

K-1 If the slotID of the next-hop node of a node is k, slotID $k-1$ is assigned to the node,

$$g^k(x) = \begin{cases} 1 & (x = k-1) \\ 0 & (x \neq k-1). \end{cases} \tag{4.13}$$

According to this SAPDF, all level l nodes have the identical slotID $N - l$. Therefore, K-1 leads to the worst case scenario.

L-BOUND The lower bound L_l of the slotID for level l node is predetermined. The slotID is randomly chosen between L_l and $k-1$,

$$g^k(x) = \begin{cases} 1/(k-L_l) & (L_l \leq x \leq k-1) \\ 0 & \text{otherwise}. \end{cases} \tag{4.14}$$

LINEAR The slotID is randomly chosen between 0 and $k-1$ according to linear distribution as shown in Fig. 4.5(a),

$$g^k(x) = \begin{cases} a_k(x+1) & (0 \le x \le k-1) \\ 0 & (k \le x \le N-1) \end{cases} \tag{4.15}$$

where

$$a_k = \frac{2}{k(k+1)}. \tag{4.16}$$

The PDNs $F_l(k)$ of LINEAR for level 1 through 3 calculated by Eqs.(4.11), (4.12), and (4.15) are shown in Fig. 4.5(b).

EXPONENTIAL The slotID is randomly chosen between 0 and $k-1$ following the exponential distribution,

$$g^k(x) = \begin{cases} e^{-\lambda_k\{k-(x+1)\}} - e^{-\lambda_k(k-x)} & (0 \le x \le k-1) \\ 0 & (k \le x \le N-1). \end{cases} \tag{4.17}$$

An example of SAPDF and PDNs of EXPONENTIAL for level 1 through 3 are shown in Fig. 4.6. A smaller value of coefficient λ_k gives a more gentle slope, which means that nodes are distributed more widely over time slots. However it introduces more isolated nodes, because the probability of assigning smaller slotID becomes larger. We can deduce λ_k which makes the ratio of isolated nodes of level l less than $1 - P^\star$ as

$$\lambda_k > -\frac{\log(1 - \sqrt[l-1]{P^\star})}{k-1}. \tag{4.18}$$

Refer to the next subsection for details. If $k = 1$, the slotID given must be 0,

$$g^1(x) = \begin{cases} 1 & (x = 0) \\ 0 & (1 \le x \le n-1). \end{cases} \tag{4.19}$$

The duplication probability q in Eqs.(4.7) and (4.8) for each of four SAPDFs can be derived as

$$q = \sum_{x=0}^{N-1} \{g^k(x)\}^2. \tag{4.20}$$

For example, the probability q for all four SAPDFs at a node with a next-hop node of slotID $k = 100$ is summarized in Table 4.1. In K-1, all nodes of the same level have the same slotID, thus $q = 1$. In L-BOUND, q heavily depends on L_1. For EXPONENTIAL, we set λ_k to $11.5/(k-1)$ in the table, which means that the ratio of isolated nodes at level 10 is kept less than 0.01 %.

Table 4.1: The q values of four SAPDFs at $k = 100$.

	K-1	L-BOUND	LINEAR	EXPONENTIAL
q	1	$1/(N - L_1)$	0.013	0.058

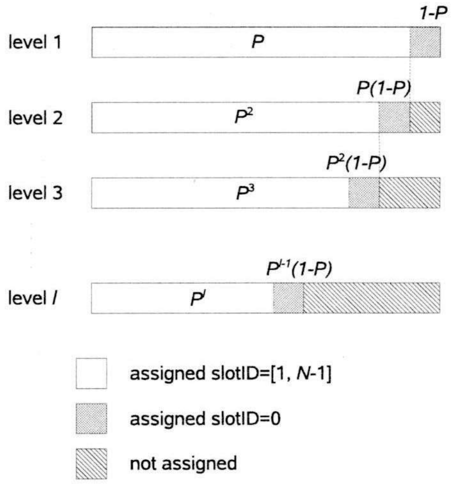


Figure 4.7: The ratio of isolated nodes.

4.4.3 Parameter Determination of EXPONENTIAL SAPDF

In the EXPONENTIAL SAPDF, parameter λ_k determines the distribution of slotIDs. With small λ_k, slotIDs are widely distributed, with which probability q becomes small and the contention degree becomes low. However, as an adverse effect, it generates many isolated nodes by assigning a small slotID to a low level node. Therefore, we need to find appropriate λ_k which guarantees the maximum ratio of isolated nodes of level l at $1 - P^\star$.

We define P as the probability that slotID $k = [1, N-1]$ is assigned to a node of level 1, namely,

$$P = \sum_{i=1}^{N-1} F_1(i) \tag{4.21}$$

$$1 - P = F_1(0). \tag{4.22}$$

Among level 2 nodes, those have a level 1 node with slotID $k = 0$ as their next-hop cannot obtain a valid slotID. Therefore, they are isolated. For simplicity, ignoring a case that a level 2 node has two or more parent nodes, we assume that $1 - P$ of level 2 nodes are isolated. Among level 2 nodes with a parent node with non-zero slotID, P get slotID $k = [1, N-1]$ and $1 - P$ get slotID $k = 0$. Therefore, among all level 2 nodes, the probability of getting slotID $k = [1, N-1]$ and $k = 0$ are P^2 and $P(1-P)$ respectively. $1 - P$ of level 2 nodes are isolated as shown in

Fig. 4.7. Repeating this procedure with the assumption

$$P = \sum_{i=1}^{N-1} F_l(i) \tag{4.23}$$

$$1 - P = F_l(0), \tag{4.24}$$

at level l, the ratio of nodes with a valid slotID is P^{l-1}, and this must be larger than $P^\star$,

$$P^{l-1} > P^\star \tag{4.25}$$

Therefore,

$$P > \sqrt[l-1]{P^\star}. \tag{4.26}$$

Substituting Eq.(4.23), we get

$$\sum_{i=1}^{N-1} F_l(i) > \sqrt[l-1]{P^\star}. \tag{4.27}$$

From Eq.(4.12), the left hand side of Eq.(4.27) becomes

$$\begin{aligned}\sum_{i=1}^{N-1} F_l(i) &= \sum_{i=1}^{N-1} \left(\sum_{k=0}^{N-1} g^k(i) F_{l-1}(k) \right) \\ &= \sum_{k=0}^{N-1} \left(F_{l-1}(k) \sum_{i=1}^{N-1} g^k(i) \right). \end{aligned} \tag{4.28}$$

If

$$\sum_{i=1}^{N-1} g^k(i) > \sqrt[l-1]{P^\star} \tag{4.29}$$

holds,

$$\begin{aligned}\sum_{i=1}^{N-1} F_l(i) &> \sqrt[l-1]{P^\star} \sum_{k=0}^{N-1} F_{l-1}(k) \\ &= \sqrt[l-1]{P^\star}. \end{aligned} \tag{4.30}$$

Therefore, Eq.(4.29) is a sufficient condition for Eq.(4.27). For the EXPONENTIAL SAPDF, from Eq.(4.17),

$$\sum_{i=1}^{N-1} g^k(i) = 1 - e^{-\lambda_k (k-1)}. \tag{4.31}$$

Substituting Eq.(4.31) into Eq.(4.29), we finally get

$$\lambda_k > -\frac{\log(1 - \sqrt[l-1]{P^\star})}{k-1}. \tag{4.32}$$

If the slotID of a next-hop node is $k = 1$, we have to assign $k = 0$ to the node. Then,

$$g^1(x) = \begin{cases} 1 & (x = 0) \\ 0 & (1 \leq x \leq N-1). \end{cases} \tag{4.33}$$

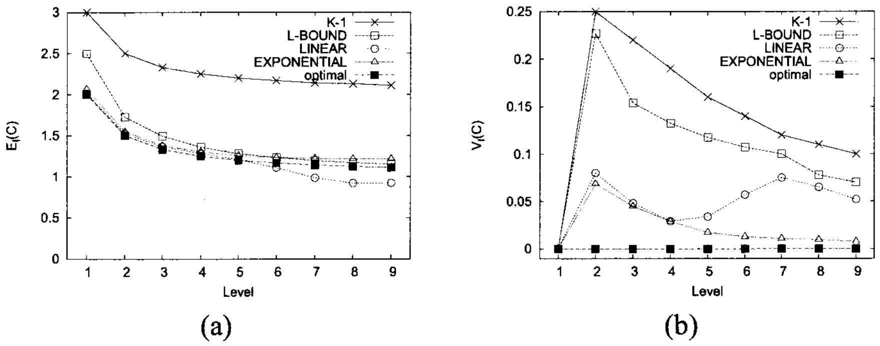


Figure 4.8: The (a) mean and (b) variance of the contention degree.

Table 4.2: Evaluation metrics for four SAPDFs.

	K-1	L-BOUND	LINEAR	EXPONENTIAL
p_{empty} (%)	90.0	11.8	65.9	29.4
q_{isolated} (%)	0	0	41.0	0.00818

4.4.4 Simulation experiments

Now, we compare the four SAPDFs by using four evaluation metrics defined in Section 4.4.1. 220 nodes are arranged in a grid network centered at a BS. The maximum hop count of a node, *i.e.*, level value, is 10. The number of time slots, N, is set to 100. For L-BOUND, the lower bound of slotID L_l for each level is determined taking into account the number of nodes belonging to the level. For the grid network, the number of nodes of level l is $4l$, thus L_l is determined as,

$$\frac{L_{l-1} - L_l}{L_l - L_{l+1}} = \frac{l}{l+1} \tag{4.34}$$

where $L_0 = N$ and $L_{10} = 0$. For EXPONENTIAL, $\lambda_k = 11.5/(k-1)$.

The simulation experiments are repeated 500 times and the results are averaged to obtain the four evaluation metrics. For the average contention degree $E_l(C)$ and the variance of contention degree $V_l(C)$ of level l, we first calculate the average contention degree $\overline{C}_i$ of each node over 500 simulation experiments, and then take the mean and variance of $\overline{C}_i$ over nodes in each level,

$$E_l(C) = E[\overline{C}_i : i \in S_l] \tag{4.35}$$

$$V_l(C) = V[\overline{C}_i : i \in S_l]. \tag{4.36}$$

where S_l is a set of all level l nodes.

In the optimal case, the contention degree of all nodes of level l should be equal to the average number of child nodes per node, thus

$$E_l^{\text{opt}}(C) = (l+1)/l, \tag{4.37}$$

$$V_l^{\text{opt}}(C) = 0, \qquad (4.38)$$

considering that the number of level l nodes is $4l$ in the grid network.

The results are shown in Fig. 4.8 and Table 4.2. Note that $E_1(C)$ of four SAPDFs are well consistent with Eq.(4.7) and Table 4.1. We can see that the graph of EXPONENTIAL is very close to that of the optimal case and the best among four SAPDFs. L-BOUND gives low contention degree at higher level nodes and high slot utilization, but it has higher variance than LINEAR or EXPONENTIAL does. In LINEAR and EXPONENTIAL, as the level increases, slotIDs of b-nodes tend to be smaller than those of m-nodes. It is because that a b-node chooses a parent node with smaller k in determining its own slotID, whereas an m-node does not. For having a smaller slotID, a b-node is chosen more often than an m-node. Then, the difference between the contention degree of m-nodes and that of b-nodes becomes smaller as the level increases. However, in L-BOUND, the dependency on next-hop's slotID is not so strong as in LINEAR or EXPONENTIAL, because a node randomly selects a slotID within a predetermined time slot range. This is the reason why the variance of L-BOUND is much larger than those of LINEAR and EXPONENTIAL. Moreover, the performance of L-BOUND would heavily depend on L_l, which could be a disadvantage in a real deployment, where the node distribution cannot be predicted well.

In LINEAR, $E_l(C)$ decreases and $V_l(C)$ increases at high level nodes. It is caused by isolated nodes. In Fig. 4.5(b), we can see that the peak of PDN for level 3 node is at $k = 12$, which means that higher level nodes have little choice of slotIDs. As a result, 41 % of nodes are isolated as shown by q_{isolated} in Table 4.2. This reduces the number of nodes joining the schedule and the contention degree at higher level nodes becomes lower.

4.5 Optimization of Exponential Distribution for a Grid Network

Next we propose a method to make m-nodes and b-nodes have the equal contention degree by tuning parameters of the EXPONENTIAL SAPDF $g^k(x)$. We first look for a necessary condition for the homogeneous contention. Then, we consider how to determine the parameter of the EXPONENTIAL SAPDF to satisfy the necessary condition. Finally we verify the performance using simulation experiments.

4.5.1 Stochastic analysis

As we discussed in Section 4.3.2, m-nodes have higher contention degree than b-nodes for having more child nodes. We consider to make the contention degree same or similar among m-nodes and b-nodes. The basic idea is to employ a different parameter of the EXPONENTIAL SAPDF for m-nodes and b-nodes, so that a node which has both m-nodes and b-nodes as its parents to choose a b-node as its next-hop node with a higher probability. In the discussion in Section 4.3.2, we assumed that the probability that node X chose m-node A as its next-hop node was equal to the probability of choosing b-node B by using the same SAPDF for both nodes. Now, we employ a larger coefficient of the exponential distribution for m-nodes to get a larger slotID than those of b-nodes.

In a grid network shown in Fig. 4.4, k_A, k_B, k_C, and k_E are slotIDs of node A, B, C, and E, respectively. Letting $p = P(k_A > k_B) = P(k_A > k_E)$ the probability that an m-node of level l has a larger slotID than that of a b-node of the same level l, $q_1 = P(k_A = k_B) = P(k_A = k_E)$ the probability that an m-node of level l has an equal slotID to that of a b-node of level l, and $q_2 = P(k_B = k_C)$ the probability that two b-nodes of level l have the identical slotID, we obtain the expected contention degree $E(C_A)$, $E(C_B)$, and $E(C_C)$ of m-node A, b-node B which has an m-node as its neighbor, and b-node C which does not have any m-node as its neighbors, respectively. After similar calculation as in Section 4.3.2, we get

$$E(C_A) = 3 - 2p, \tag{4.39}$$

$$E(C_B) = p + q_1 + \frac{1+q_2}{2}, \tag{4.40}$$

$$E(C_C) = 1 + q_2. \tag{4.41}$$

For the equal contention degree, $E(C_A) = E(C_B) = E(C_C)$, erasing q_2, we obtain

$$2p + q_1 = \frac{3}{2}. \tag{4.42}$$

This is a necessary condition for the identical contention degree among nodes of level l. For a grid network with a BS at a corner, this condition becomes

$$\frac{3}{2}p + q_1 = 1. \tag{4.43}$$

Now, let us assume that an m-node and a b-node have their next-hop node with slotID k and their slotIDs are assigned by SAPDFs g_m^k and g_b^k respectively. The probability p in Eq.(4.42) that the m-node obtained a greater slotID than one assigned to the b-node can be calculated as

$$p = \sum_{x=1}^{N-1} \left(g_m^k(x) \sum_{i=0}^{x-1} g_b^k(i) \right). \tag{4.44}$$

The probability q_1 in Eq.(4.42) that the same slotID is assigned to the two nodes is given as

$$q_1 = \sum_{x=0}^{N-1} g_m^k(x) g_b^k(x). \tag{4.45}$$

In order to assign a larger slotID to an m-node, we employ a larger coefficient for g_m^k than g_b^k in the EXPONENTIAL SAPDF,

$$g_m^k(x) = \begin{cases} e^{-r\lambda_k\{k-(x+1)\}} - e^{-r\lambda_k(k-x)} & (0 \le x \le k-1) \\ 0 & (k \le x \le N-1) \end{cases} \tag{4.46}$$

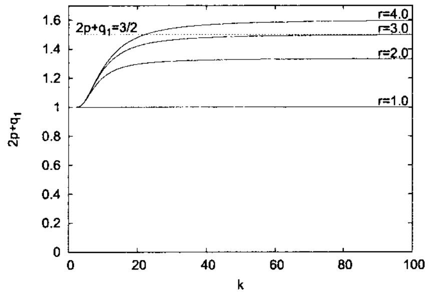


Figure 4.9: The parameter r and $2p + q_1$.

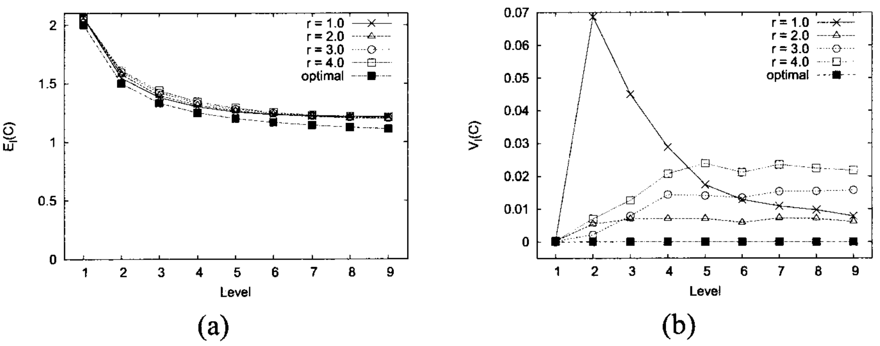


Figure 4.10: The (a) mean and (b) variance of the contention degree with the parameter tuning.

$$g_b^k(x) = \begin{cases} e^{-\lambda_k\{k-(x+1)\}} - e^{-\lambda_k(k-x)} & (0 \le x \le k-1) \\ 0 & (k \le x \le N-1) \end{cases} \tag{4.47}$$

where $r \ge 1$.

Using Eq.(4.44), (4.45), (4.46), and (4.47), we obtain $2p + q_1$, which is the left hand side of Eq.(4.42), illustrated in Fig. 4.9 for $r = 1.0$, 2.0, 3.0, and 4.0, where $\lambda_k = 11.5/(k-1)$ and $N = 100$. We can see that $r = 3.0$ is the best among the four values to satisfy Eq.(4.42).

4.5.2 Simulation experiments

In order to evaluate the contribution of the parameter tuning, we conduct simulation experiments using the same grid network as in Section 4.4.4. The number of time slots N is set to 100. The EXPONENTIAL SAPDFs by Eq.(4.46) and (4.47) with $\lambda_k = 11.5/(k-1)$ are used for m-nodes and b-nodes respectively.

The mean $E_l(C)$ and variance $V_l(C)$ of the contention degree at level l are plotted in Fig. 4.10 by changing r. $E_l(C)$ slightly increases with r, since q_1 in Eq.(4.40) increases. $V_2(C)$ dramatically decreases with $r = 3.0$ compared to the case without the tuning ($r = 1.0$), which means that the contention degree is well equalized among an m-node and b-nodes. At level 3 or above, $r = 2.0$ gives

Table 4.3: Evaluation metrics and r in a grid network.

r	1.0	2.0	3.0	4.0
p_{empty} (%)	29.4	31.9	33.7	35.1
q_{isolated} (%)	0.008	0.006	0	0

better results than $r = 3.0$ does. A b-node has two parent nodes and chooses one with the smaller slotID as its next-hop node, while an m-node has only one parent which is also an m-node. Therefore, even in the case without the parameter tuning, slotIDs of b-nodes tend to be smaller than those of m-nodes as the level increases. Increasing r enhances this effect even further, thus p becomes too large to satisfy Eq.(4.42) with $r = 3.0$ or more at higher levels.

Table 4.3 shows p_{empty} and q_{isolated} for each of four r values. As r increases, p_{empty} also increases, since m-nodes are distributed within a narrower range of time slots, in other words, more packed.

4.6 Conclusion

In this chapter, a WSN protocol for automatic meter reading in a large-scale apartment building was proposed. To accomplish both of low duty cycle and delay-bounded transmission of urgent information, the interval of length of the delay bound is first divided into time slots. A node wakes up at its assigned slot to receive packets from child nodes and at a slot of its parent node to send a packet if it has. A node determines its time slot in a distributed and stochastic manner, so that every node has a parent node with a larger slotID. Consequently, a packet originating at an arbitrary node can be relayed to the BS within the delay bound. It was shown that the EXPONENTIAL SAPDF enabled less and more identical contention degree at all level nodes in a grid network and the results were near-optimal.

Chapter 5

Conclusion

The WSN technology is a key for creating our safe, secure, and comfortable living environment. WSNs used as a social infrastructure have to be capable to deliver urgent information, such as a fire alarm, fast and reliably. Because of a large number of deployed sensor nodes in a WSN, control mechanisms of a WSN should be fault tolerant, energy-efficient, and scalable. In addition, adaptability is a necessary factor for such a WSN, since the scale of an emergency event is unpredictable and dynamically changing. A centralized control is not feasible to satisfy these requirements, and a distributed and self-organizing control is preferable.

We proposed a design methodology of a WSN architecture for urgent sensor information in which several simple and fully-distributed mechanisms working in different spatial and temporal levels are combined instead of developing a complicated monolithic protocol. The collective control of these mechanisms offers preferential transmission of urgent information adapting to the scale of emergency.

We also showed two WSN protocols which are designed following the methodology. In UMIUSI protocol, sensor information is categorized into three traffic classes: normal, important, and critical. In order to prioritize transmission of the critical class, five simple mechanisms, *i.e.*, the assured corridor mechanism (ACM), retransmission, priority queueing, rate control by local congestion detection, and rate control by backpressure, collaborate consistently. In our newly developed ACM, an assured corridor is eventually established from a source node to the BS. In the corridor, all nodes keep awake for fast transmission of emergency packets. Beside the corridor, all nodes refrain from transmission of normal packets to avoid disturbing transmission of emergency packets in the corridor. The other nodes stay in normal operation. We verified through simulation and practical experiments that UMIUSI protocol successfully improved the delivery ratio and the delay of emergency packets regardless of the scale of emergency.

Another WSN protocol for automatic meter reading, in which low duty cycle and bounded delay of an event notification are guaranteed, was proposed. In this protocol, one working cycle, whose length is equal to the delay-bound, is divided into a number of time slots, and each node determines its own time slot to be active in a distributed manner. By employing an appropriate SAPDF, earlier time slots are given to nodes further from the BS, which enables for urgent event notification to be delivered to the BS within the delay-bound. The EXPONENTIAL SAPDF gives less intense and more homogeneous contention over a grid network than other three SAPDFs. We further tune the parameter of the EXPONENTIAL SAPDF to give an even better contention situation. We verified the performance of these SAPDFs through simulation experiments, and the results showed that our protocol successfully provided the near-optimal contention situation.

Although a lot of problems to be solved are left as open issues, we believe that our research results will promote the advances of the WSN technology and help people to enjoy their safe, secure, and comfortable future life.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, March 2002.

[2] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pages 28–39, Los Angeles, California, USA, November 2003.

[3] R. Rajagopalan and P. K. Varshney. Data aggregation techniques in sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 8(4):2–17, 2006.

[4] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *IEEE Network*, 18(4):45–50, 2004.

[5] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE computer and Communications Societies (INFOCOM 2002)*, volume 3, pages 1567–1576, New York, NY, USA, June 2002.

[6] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03)*, pages 171–180, Los Angeles, California, USA, November 2003.

[7] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pages 95–107, Baltimore, MD, USA, November 2004.

[8] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys '06)*, pages 321–334, Boulder, Colorado, USA, November 2006.

[9] L. F. W. van Hoesel and P. J. M. Havinga. A lightweight medium access protocol (lmac) for wireless sensor networks. In *Proceedings of the 1st international conference on networked sensing systems (INSS 2004)*, Tokyo, Japan, June 2004.

[10] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03)*, pages 181–192, Los Angeles, California, USA, November 2003.

[11] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6):6–28, December 2004.

[12] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, May 2005.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom 2000)*, pages 56–67, Boston, Massachusetts, United States, August 2000.

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 2000)*, volume 2, pages 1–10, Hawaii, USA, January 2000.

[15] Y. Yu, D. Estrin, and R. Govindan. Geographical and energy-aware routing: A recursive data dissemination protocol for wireless sensor networks. Technical Report UCLA-CSD TR-010023, UCLA Computer Science Department, May 2001.

[16] D. Chen and P. K. Varshney. QoS support in wireless sensor networks: A survey. In *Proceedings of the International Conference on Wireless Networks (ICWN 2004)*, pages 227–233, Las Vegas, Nevada, USA, June 2004.

[17] M. Younis, K. Akkaya, M. Eltoweissy, and A. Wadaa. On handling QoS traffic in wireless sensor networks. In *Proceedings of the 37th Annual Hawaii International Conference on System Scieneces (HICSS 2004)*, Hawaii, USA, January 2004.

[18] S. Pack, J. Choi, T. Kwon, and Y. Choi. Application aware data aggregation in wireless sensor networks. In *Proceedings of the 1st IEEE International Workshop on Adaptive Wireless Networks (AWiN)*, November 2005.

[19] B. Deb, S. Bhatnagar, and B. Nath. ReInForM: Reliable information forwarding using multiple paths in sensor networks. In *Proceedings of 28th Annual IEEE conference on Local Computer Networks (LCN 2003)*, pages 406–415, Bonn, Germany, October 2003.

[20] Y. Sankarasubramaniam, B. Akan, and I. F. Akyilidiz. ESRT: Event-to-sink reliable transport in wireless sensor networks. In *Proceedings of the 4th ACM International symposium on Mobile ad hoc networking and computing (MobiHoc 2003)*, pages 177–188, Annapolis, Maryland, USA, June 2003.

[21] H. Dubois-Ferrière, D. Estrin, and M. Vetterli. Packet combining in sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys '05)*, pages 102–115, San Diego, California, USA, November 2005.

[22] H. Luo, Z. Zhang, and Y. Liu. Recoda: reliable forwarding of correlated data in sensor networks with low latency. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing (IWCMC '07)*, pages 278–283, Honolulu, Hawaii, USA, August 2007.

[23] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. A spatiotemporal communication protocol for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 16(10):995–1006, 2005.

[24] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, pages 2646–2657, Miami, Florida, USA, March 2005.

[25] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell. CODA: congestion detection and avoidance in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03)*, pages 266–279, Los Angeles, California, USA, November 2003.

[26] B. Hull, K. Jamieson, and H. Balakrishnan. Mitigating congestion in wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pages 134–147, Baltimore, Maryland, USA, November 2004.

[27] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '06)*, pages 63–74, Pisa, Italy, September 2006.

[28] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft. Overload traffic management for sensor networks. *ACM Transactions on Sensor Networks*, 3(4):18, 2007.

[29] C.-F. Chiasserini and M. Garetto. Modeling the performance of wireless sensor networks. In *Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM 2004)*, pages 220–231, Hong Kong, March 2004.

[30] O. Dousse, P. Mannersalo, and P. Thiran. Latency of wireless sensor networks with uncoordinated power saving mechanisms. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '04)*, pages 109–120, Tokyo, Japan, May 2004.

[31] R. Cohen and B. Kapchits. An optimal algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network. In *Proceedings of the 26rd Anuual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2007)*, pages 258–266, Anchorage, Alaska, USA, May 2007.

[32] W. Lai and I. C. Paschalidis. Sensor network minimal energy routing with latency guarantees. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '07)*, pages 199–208, Montreal, Quebec, Canada, September 2007.

[33] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pages 20–27, Los Angeles, California, USA, April 2005.

[34] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel. Delay efficient sleep scheduling in wireless sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, pages 2470–2481, Miami, Florida, USA, March 2005.

[35] A. Keshavarzian, H. Lee, and L. Venkatraman. Wakeup scheduling in wireless sensor networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '06)*, pages 322–333, Florence, Italy, May 2006.

[36] Y. Gu and T. He. Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links. In *Proceedings of the 5th international conference on Embedded networked sensor systems (SenSys '07)*, pages 321–334, Sydney, Australia, November 2007.

[37] Y. Wu, S. Fahmy, and N. B. Shroff. Energy efficient sleep/wake scheduling for multi-hop sensor networks: Non-convexity and approximation algorithm.

In *Proceedings of the 26rd Anuual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2007)*, pages 1568–1576, Anchorage, Alaska, USA, May 2007.

[38] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: a reliable transport protocol for wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA 2002)*, pages 1–11, Atlanta, Georgia, USA, September 2002.

[39] A. Woo and D. E. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking (MobiCom 2001)*, pages 221–235, Rome, Italy, July 2001.

[40] IEEE 802.15.4. Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks, 2003.

[41] N. Wakamiya and M. Murata. Synchronization-based data gathering scheme for sensor networks. *IEICE Transactions on Communications*, E88-B(3):873–881, March 2005.

[42] S. Bhatnagar, B. Deb, and B. Nath. Service differentiation in sensor networks. In *Proceedings of the 4th International Symposium on Wireless Personal Multimedia Communications (WPMC 2001)*, Aalborg, Denmark, September 2001.

[43] K. Leibnitz, N. Wakamiya, and M. Murata. Modeling of IEEE 802.15.4 in a cluster of synchronized sensor nodes. In *Proceedings of the 19th International Teletraffic Congress (ICT-19)*, pages 1345–1354, Beijing, China, August 2005.

[44] S. Kashihara, N. Wakamiya, and M. Murata. Implementation and evaluation of a synchronization-based data gathering scheme for sensor networks. In *Proceedings of IEEE International Conference on Communications, Wireless Networking (ICC 2005)*, pages 3037–3043, Seoul, Korea, May 2005.

[45] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the 5th symposium on Operating systems design and implementation (OSDI '02)*, pages 147–163, Boston, Massachusetts, USA, December 2002.

[46] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03)*, pages 138–149, Los Angeles, California, USA, November 2003.

[47] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pages 39–49, Baltimore, MD, USA, November 2004.

[48] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava. Estimating clock uncertainty for efficient duty-cycling in sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys '05)*, pages 130–141, San Diego, California, USA, November 2005.

[49] T. Herman and C. Zhang. Stabilizing clock synchronization for wireless sensor networks. In *Proceedings of the 8th international symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2006)*, Dallas, Texas, USA, November 2006.

[50] L. Chenyang, B. M. Blum, T. F. Abdelzaher, and H. Tian. RAP: a real-time communication architecture for large-scale wireless sensor networks. In *Proceedings of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2002)*, pages 55–66, San Jose, California, USA, September 2002.

[51] K. Akkaya and M. Younis. Energy and QoS aware routing in wireless sensor networks. *Cluster Computing*, 8(2-3):179–188, July 2005.

[52] A. Mahapatra, K. Anand, and D. P. Agrawal. QoS and energy aware routing for real-time traffic in wireless sensor networks. *Computer Communications*, 29(4):437–445, February 2006.

[53] B. Deb, S. Bhatnagar, and B. Nath. Information assurance in sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications (WSNA 2003)*, pages 160–168, San Diego, California, USA, September 2003.

Appendix A

Application of Assured Corridor Mechanism to the Synchronization-based Data Gathering Scheme

A.1 Synchronization-based Data Gathering Scheme

Although the assured corridor mechanism (ACM) does not depend on any specific routing algorithm, we employ the synchronization-based data gathering scheme [41, 44] to evaluate the behavior of our mechanism in this thesis. The synchronization-based data gathering scheme is proposed to accomplish energy-efficient data gathering in a WSN without any centralized control. Combined with ACM, it enables a distributed and self-organizing control of fast and reliable transmission of urgent sensor information.

Figure A.1 illustrates how sensor data propagate from all nodes to the BS. A number in each circle, *i.e.*, node, corresponds to the number of hops from the BS, which is maintained by each node as a level value. Figure A.2(a) shows timings that level $n-2$, $n-1$, and n nodes wake up, receive and send packets, and go to sleep. Here, the interval of data gathering is given as t_{norm}.

In the synchronization-based data gathering scheme, sensor nodes with the same level value behave in synchrony. When so-called global synchronization is accomplished, an interval between packet emission of level i nodes and that of level $i-1$ nodes becomes δt_{norm}. Now assume that the most distant nodes are at n hops from the BS. First, all level n nodes wake up and then broadcast a packet at the same time. This broadcasting is scheduled slightly before timing of packet emission of level $n-1$ nodes by δt_{norm}. At this time, all level $n-1$ nodes wake up. They receive packets from level n nodes in their vicinity and aggregate or fuse the received data with their own data. Then, they broadcast a packet containing the aggregated sensor data slightly before timing of packet emission of level $n-2$

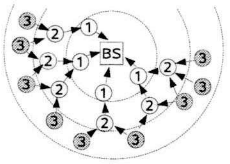


Figure A.1: The synchronization-based data gathering scheme.

nodes by δt_{norm}, so that level $n - 2$ nodes, which are awake at this time, can receive the packet. At the same time, level n nodes also receive packets emitted by level $n - 1$ nodes to maintain the synchronization and then go to sleep. Therefore all sensor nodes need to be awake for $2\delta t_{\mathrm{norm}}$ during the interval of data gathering as illustrated in Fig. A.2(a). For further details of the synchronization-based data gathering, refer to [41].

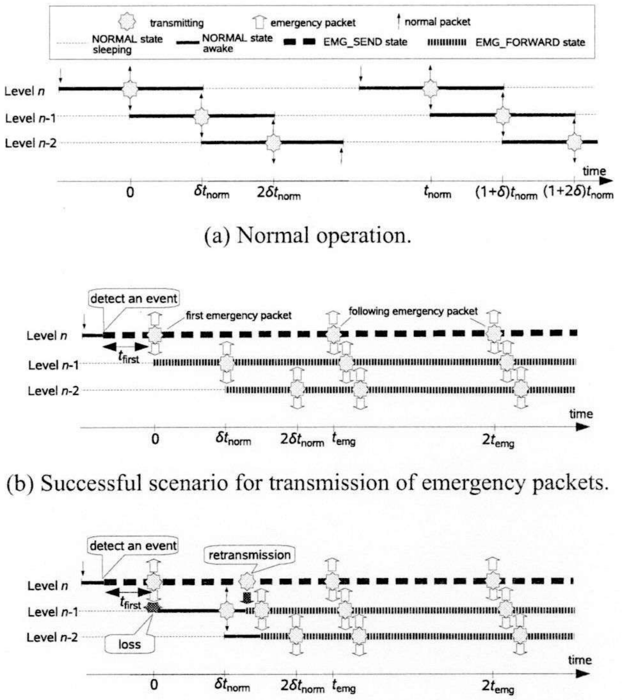
In the rest of this chapter, we call one-level smaller nodes of a node as "parent nodes" and one-level larger nodes as "child nodes." As easily imagined from their names, the mechanism proposed in this paper can directly be applied to a tree-based data gathering scheme as shown in Fig. 3.2(a). In the case of MANET-type schemes, where one or more paths are explicitly built for communication between a node and a BS, parent nodes correspond to the next-hop nodes to the BS and child nodes correspond to the preceding nodes.

A.2 In Emergency

Transmission of urgent information in the synchronization-based data gathering scheme with ACM is depicted in Fig. A.2(b). When a node detects an emergency event, it moves into *EMG_SEND* state. It defers emission of the first emergency packet for t_{first} until its next timing of packet emission ($t = 0$, in Fig. A.2(b)), since its parent nodes in the *NORMAL* state are asleep at the moment of event detection and can not receive any packet until they wake up. To minimize this delay, one possible way is to have a mechanism to wake up parent nodes by sending a wake-up signal. ACM can be combined with such a mechanism, however, a special hardware for the wake-up mechanism is needed on every node, which leads to additional production cost.

When the first emergency packet is broadcast at $t = 0$, parent nodes move to the *EMG_FORWARD* state while neighboring nodes of the same or a larger level move to the *SUPPRESSED* state. By hop-by-hop broadcasting of the first emergency packet, all intermediate nodes move to the *EMG_FORWARD* state.

If the first emergency packet is lost at one of intermediate nodes due to collision for example, the transmission is delayed for t_{norm} until the next timing of packet emission. Letting t^{i}_{relay} time taken for an *EMG_FORWARD* node of level i to wait for level $i - 1$ nodes to wake up, the total delay of the first emergency packet



(a) Normal operation.

(b) Successful scenario for transmission of emergency packets.

(c) Scenario with retransmission of the first emergency packet.

Figure A.2: Transmission sequences in the synchronization-based data gathering scheme.

originating at level n is given by,

$$D_n = t_{\text{first}} + \sum_{i=1}^{n-1} t^i_{\text{relay}} + kt_{\text{norm}}, \tag{A.1}$$

where k is the number of packet loss events.

The maximum of t_{first} is t_{norm}, and all intermediate nodes have to wait for δt_{norm}, if they are in the *NORMAL* state,

$$\max(t_{\text{first}}) = t_{\text{norm}}, \tag{A.2}$$

$$\max\left(\sum_{i=1}^{n-1} t^i_{\text{relay}}\right) = (n-1)\delta t_{\text{norm}}. \tag{A.3}$$

Thus the maximum of delay D_n for the first emergency packet originating at an *EMG_SEND* node of level n to reach the BS is given by

$$\max(D_n) = t_{\text{norm}} + (n-1)\delta t_{\text{norm}} + kt_{\text{norm}}. \tag{A.4}$$

The *EMG_SEND* node keeps sending emergency packets at an interval of t_{emg} after its emission of the first emergency packet. These following emergency packets are forwarded to the BS immediately along the corridor. If δt_{norm} is relatively large compared to the transmission delay of following emergency packets, a following emergency packet may catch up the preceding first emergency packet at an intermediate node. In this case, the following emergency packet takes over the first emergency packet and propagates to the BS establishing a corridor as being treated as the first emergency packet.

A.3 Retransmission

Since a corridor is not established, the first emergency packet is forwarded to the BS without any prioritization and can get lost. Although a following emergency packet succeeds the role of a lost first emergency packet in establishing a corridor, it increases the transmission delay of emergency packets and can be critical to the safety and security of our living environment. Following emergency packets can be lost as well, because of possible collisions among emergency packets.

There are several possibilities to overcome the loss of emergency packets. In this thesis, we take a hop-by-hop acknowledgement and retransmission scheme at a higher layer above MAC. Our scheme does not exclude other techniques and they are helpful to improve the reliability of transmission. For example, we could adopt a MAC protocol with prioritization [50] or a packet-level priority control. In [42, 19], the authors consider service differentiation in terms of delivery ratio based on DiffServ model. Delay-based differentiation is proposed in, for example, [51, 52]. Multipath routing / forwarding is another possibility to improve the reliability of packet transmission [42, 19, 53].

The synchronization-based data gathering scheme inherently enables hop-by-hop acknowledgement since a node receives a packet from a parent node for synchronization. In other kind of schemes, a node can also expect to receive an emergency packet from its parent node at the timing of packet emission of the parent node. A node can confirm the successful transmission of its emergency packet by observing a packet sent by one of its parent nodes. If a node does not receive an emergency packet from its parent node or an emergency packet broadcast by its parent node does not contain urgent information it sent, the emergency packet is considered lost. Then, it retransmits the emergency packet. The retransmission scheme of the first emergency packet in the synchronization-based data gathering scheme is shown in Fig. A.2(c).

First, level n node which detects an emergency event immediately moves to the *EMG_SEND* state. Then, it transmits an emergency packet at the next timing of packet emission, $t = 0$. A level $n - 1$ node, which is a parent node of the level n node, is expected to move to the *EMG_FORWARD* state and broadcast the emergency packet at the next timing of packet emission at $t = \delta t_{\text{norm}}$. However, the first emergency packet can be lost, due to the collision with a normal packet transmitted from a neighbor node or random channel error, for example. In this case, the level $n - 1$ node remains in the *NORMAL* state and broadcast a normal packet at its timing of packet emission, $t = \delta t_{\text{norm}}$ as shown in Fig. A.2(c). Receiving a normal packet from its parent node, the level n node detects the loss and immediately retransmits the emergency packet with a retransmission flag in the packet header. Retransmission is repeated until it receives the emergency packet from any of its parent nodes. Therefore, the duration for retransmission is at most δt_{norm}, *i.e.*, until parent nodes go to sleep. If the next emergency packet originating at the same source node arrives while an emergency packet is waiting for retransmission at an intermediate node, the waiting packet is discarded. This is because that sensor data in the waiting packet is obsoleted by the new data. It is also possible to merge them and generate a new emergency packet depending on an application's requirement.

If a level $n-1$ node receives an emergency packet with a retransmission flag while it is awake, it immediately broadcasts the emergency packet so that level $n-2$ nodes can forward the packet at the next timing of regular emission ($t = 2\delta t_{\mathrm{norm}}$ in Fig. A.2(c)). The emergency packet sent by the level $n-1$ node also confirms the successful reception to the level n node. Since the other nodes in the vicinity of a node retransmitting emergency packets do not transmit any packets during retransmission interval, we can avoid collisions and losses of retransmitted packets.

Now, consider the delay of the first emergency packet for the worst case scenario with retransmission. The maximum delay of transmission is δt_{norm} as far as retransmission succeeds before a parent node, which is still in the *NORMAL* state, goes to sleep (see retransmission from level n node to level $n-1$ node in Fig. A.2(c)). If a node fails in retransmission, it must wait for the next cycle of packet emission at t_{norm} later because its parent node are asleep until that time. Therefore, the maximum of transmission delay with retransmission, D_n^{R}, is given by,

$$\max(D_n^{\mathrm{R}}) = t_{\mathrm{norm}} + (n-1)\delta t_{\mathrm{norm}} + k' t_{\mathrm{norm}} + t_{\mathrm{retrans}}^1. \tag{A.5}$$

where k' is the the number of retransmission failures, which incur delay of t_{norm}. t_{retrans}^1 corresponds to the time for level 1 node to transmit the emergency packet to the BS. Since the BS is always ready to receive a packet, level 1 node can try retransmitting the emergency packet until the next emergency packet catches up, which is at most t_{emg} after it receives the first emergency packet. Therefore,

$$\max(D_n^{\mathrm{R}}) = t_{\mathrm{norm}} + (n-1)\delta t_{\mathrm{norm}} + k' t_{\mathrm{norm}} + t_{\mathrm{emg}}. \tag{A.6}$$

A.4 Simulation Results

We evaluate the synchronization-based data gathering scheme with ACM on the ns-2 network simulator package. The settings of the simulation experiments are the same as in Section 3.5.1. In this section, the results of the first emergency packets, which are transmitted before an corridor is established, are shown. Refer to Section 3.5.1 for the results of following emergency packets.

In ACM, the first emergency packet propagates to the BS while establishing a corridor. Thus the delay of urgent information D_n, which is defined here as the

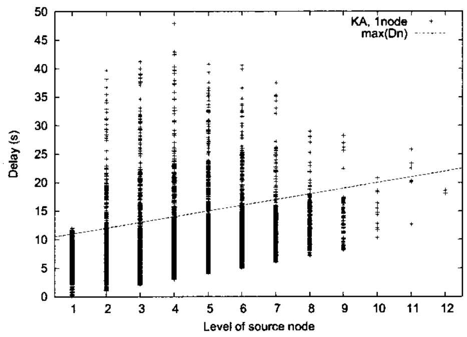


Figure A.3: The delay of the first emergency packets.

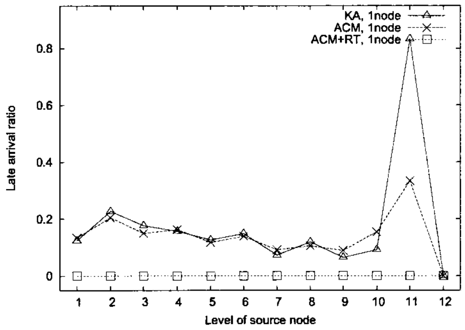


Figure A.4: The late arrival ratio of the first emergency packets.

duration from the time when a level n node detects an emergency event to the time when the BS receives an emergency packet for the first time, indicates the time required to establish an assured corridor.

The delay of first emergency packets of KA with one *EMG_SEND* node is plotted in Fig. A.3 for each level of the source node. A dashed line shows the maximum delay $\max(D_n)$ without loss, *i.e.*, $k = 0$ in Eq. (A.4), given by

$$\max(D_n)_{k=0} = t_{\text{norm}} + (n-1)\delta t_{\text{norm}}. \tag{A.7}$$

As shown in Fig. A.3, the delay exceeds $\max(D_n)_{k=0}$ in about 15 % of the transmission. In these cases, there is at least one packet loss on the way to the BS. Although the lost packet is compensated by a following emergency packet, the loss leads to the increased delay.

The late arrival ratio and the averaged delay of first emergency packets for each source level for KA, ACM, and ACM+RT are shown in Figs. A.4 and A.5 respectively. The late arrival ratio is defined here as the ratio of cases where the delay exceeds $\max(D_n)_{k=0}$ for KA and ACM, and $\max(D_n^R)_{k'=0}$ for ACM+RT substituting $k' = 0$ into Eq.(A.6),

$$\max(D_n^R)_{k'=0} = t_{\text{norm}} + (n-1)\delta t_{\text{norm}} + t_{\text{emg}}. \tag{A.8}$$

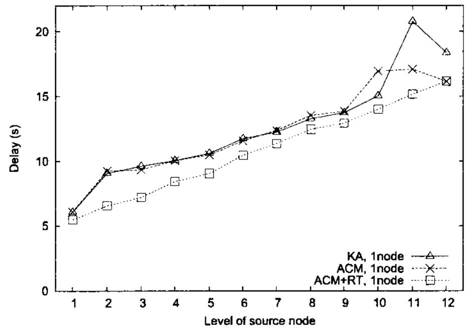


Figure A.5: The delay of the first emergency packets.

As seen in the figures, suppression of transmission of normal packets has little effect for first emergency packets because they collide with normal packets in establishing an assured corridor. In ACM+RT, in contrast to the others, all of the first emergency packets reached the BS within $\max(D_n^R)_{k'=0}$ in Eq.(A.8), which means that all first emergency packets are delivered to the BS within one cycle due to successful retransmission. This is the reason why the delay of ACM+RT is smaller than that of KA and ACM. Since no normal packets are transmitted at the time of retransmission of emergency packets, retransmission does not suffer from collisions with normal packets and is successful unless there is a collision with another emergency packet.

One might think that the absolute value of the delay is too large. However, this delay depends largely on a sleep schedule of a data gathering scheme, since a node must wait for a parent node to wake up in forwarding the first emergency packet. In the case of the synchronization-based data gathering scheme, we can shorten the delay with smaller δ in Eq.(A.4) and Eq.(A.6), but t_{first} is unavoidable without a wake-up mechanism. The results shown in this chapter can be regarded as the expected performance in the worst case scenario. The typical delay of a fire alarm for a home security system is from several tens of seconds to one minute and thus the delay of our mechanism is acceptable under the simulation setting.