IEICE TRANS. COMMUN., VOL.E95–B, NO.9 SEPTEMBER 2012

PAPER Special Section on Emerging Technologies and Applications for Ad Hoc and Wireless Mesh Networks

Robust and Resilient Data Collection Protocols for Multihop Wireless Sensor Networks

Daichi KOMINAMI[†a)], ***Nonmember***, **Masashi SUGANO**[††b)], ***Member***, **Masayuki MURATA**[†c)], ***Fellow***, ***and*** **Takaaki HATAUCHI**[†††d)], ***Member***

SUMMARY Robustness is one of the significant properties in wireless sensor networks because sensor nodes and wireless links are subjected to frequent failures. Once these failures occur, system performance falls into critical condition due to increases in traffic and losses of connectivity and reachability. Most of the existing studies on sensor networks, however, do not conduct quantitative evaluation on robustness and do not discuss what brings in robustness. In this paper, we define and evaluate robustness of wireless sensor networks and show how to improve them. By computer simulation, we show that receiver-initiated MAC protocols are more robust than sender-initiated ones and a simple detour-routing algorithm has more than tripled robustness than the simple minimum-hop routing algorithm.

key words: *robustness, resilience, sensor network, simulation*

1. Introduction

Recent advances in wireless and micro-electromechanical technologies have directed considerable attention toward ad hoc networks. Among ad hoc networks, wireless sensor networks are expected to be useful for a wide range of applications, since they have the ability to monitor a location without the need for an infrastructure. However, wireless sensor networks have critical technical problems that remain to be solved. In particular, the robustness of sensor networks is of significant concern because sensor nodes and wireless links are subject to frequent failures. In wireless sensor networks, sensor nodes fails due to harsh environmental conditions and energy depletion [1].

Robustness is then the property of maintaining or recovering performance in the face of these uncertain environmental variations, as illustrated in Fig. 1. These environmental variations entail changes of the route along which data packets are delivered to the sink node, and these route changes can lead to loss of end-to-end reachability and heavy traffic load concentration. Without adequate robustness against environmental variations, once the network conditions have been severely perturbed, system performance can fall to critical levels. Numerous approaches have been suggested to optimize sensor networks, but such approaches typically incur severe performance degradation after topological changes, since an ideal situation is assumed. To tackle these problems, mechanisms that monitor network conditions and leveraging the information on the network have been found to be effective. Although various fault tolerant techniques for wireless sensor networks were discussed by Paradis and Han [2], quantitative evaluation of robustness in wireless sensor networks has been rarely examined.

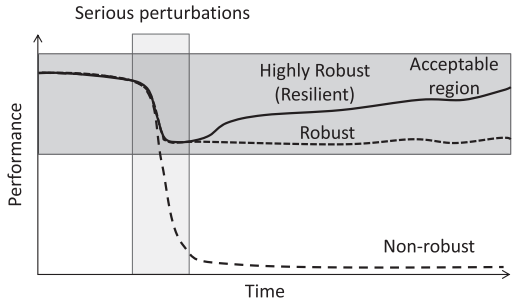


Fig. 1 Robust and non-robust systems.

In this paper, we separate robustness into two properties "robustness" and "resilience", which "maintain" and to "recover" performance in the face of uncertain environmental variations, respectively. Unless otherwise stated, we use the word performance in referring to the packet delivery ratio hereafter. We define these properties in a quantitatively evaluable form. With regard to environmental variations, we consider abrupt increases of traffic load, random failure of sensor nodes, and failure of a sink node.

The contributions of this paper include the following. First, we discuss how robustness and resilience are introduced and improved in the media access control (MAC) and network layers of a sensor system. For the MAC layer, we focus on the difference between robustness of sender-initiated and receiver-initiated MAC protocols. We show that this difference is essentially between the hard-state and soft-state [3], [4], and that the latter has higher robustness. Moreover, we show that resilience in the MAC layer is obtained from the adaptive setting of appropriate duty cycles. For the network layer, we address two points for robustness and resilience improvement: detour routing over a mesh net-

Manuscript received January 5, 2012.
Manuscript revised March 26, 2012.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.
[††]The author is with School of Knowledge and Information Systems, College of Sustainable System Sciences, Osaka Prefecture University, Sakai-shi, 599-8531 Japan.
[†††]The author is with Fuji Electric Co., Ltd., Hino-shi, 191-8502 Japan.
a) E-mail: d-kominami@ist.osaka-u.ac.jp
b) E-mail: sugano@rehab.osakafu-u.ac.jp
c) E-mail: murata@ist.osaka-u.ac.jp
d) E-mail: hatauchi-takaaki@fujielectric.co.jp

DOI: 10.1587/transcom.E95.B.2740

work and management of routing tables. We demonstrate that soft-state management of routing tables has greater resilience than the hard-state management, and that robustness is enhanced by the existence of multiple candidates of next-hop nodes over a mesh sensor network.

The rest of this paper is organized as follows. We define robustness and resilience quantitatively in Sect. 2. In Sects. 3 and 4, we then discuss robustness in the MAC layer and in the network layer, respectively. We present the simulation results in Sect. 5, and we conclude the paper in Sect. 6.

2. Quantative Definitions of Robustness and Resilience

As stated in Sect. 1, we define robustness and resilience as the properties that "maintain" and "recover" performance in the face of unexpected environmental variations, respectively. In this section, we intuitively propose quantitative expressions for robustness and resilience based on Fig. 1 and discuss how to improve them.

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

$$Robustness = \frac{|\overline{C_{before}} - \overline{C_{after}}|}{\overline{C_{before}}}, \quad (1)$$

$$Resilience = T_{recovery} - T_{variation}, \quad (2)$$

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

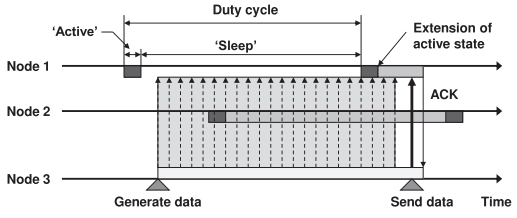
In order for improvement of robustness, retransmission mechanisms are of important. In the MAC layer, the one-to-one message retransmission advances robustness of the data delivery, and the network layer can enhance robustness by utilizing alternative and detour paths. These mechanisms keep the packet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

3. Robustness and Resilience in MAC Layer Protocols

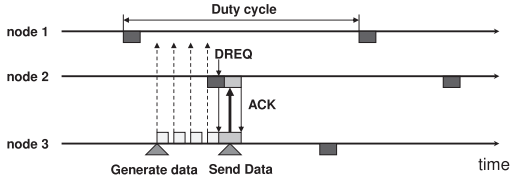
Considerable importance is placed on the energy efficiency of MAC layer protocols in sensor networks [5], and many duty-cycle MAC protocols have been proposed [6]–[13]. Therefore, we examine duty-cycle MAC protocols in this paper. Power-saving operation in duty-cycle MAC protocols is based on the fact that sleeping nodes consume significantly less energy than idling nodes [14]. However, since nodes turn off their wireless interfaces, the nodes must control their wake-up timings in order to communicate with other nodes. According to whether the sender or receiver initiates communications, duty-cycle MAC protocols are respectively classified into two types: sender-initiated [6]–[10] and receiver-initiated MAC protocols [11]–[13]. In the subsequent sections, we describe both sender-initiated and receiver-initiated MAC protocols, and show that the difference between these two types is essentially between hard- and soft-states. Furthermore, the "soft-state", which is often referred to in network protocol designs [15]–[18], is important for robustness improvement.

3.1 Sender-Initiated MAC Protocols

B-MAC [7] is the basis of *low power listening* (LPL) protocols in which receiver nodes periodically probe the state of the channel (Fig. 2(a)). Figure 2(a) presents an instance where node 3 (the sender) is ready to send a data packet to node 1 (the receiver). If the channel is idle, the receiver returns to the sleep state after probing. In contrast, if the channel is busy, preparations are made to be ready for data



(a) B-MAC



(b) X-MAC

Fig. 2 Sender-initiated MAC protocols.

reception. After receiving intended data, node 1 returns an acknowledgement (ACK) message. To activate the channel and initiate communication, the sender sends a continuous preamble over a period of time that is longer than the duty cycle. The sender then sends the data after sending the preamble. A number of shortfalls are found in using this protocol. As the duty cycle increases, each sender node occupies the channel for a longer period of time during preamble transmission. Such occupation of the channel then interferes with communication between neighboring nodes. Moreover, preamble transmission from the sender consumes the power of unrelated receivers, and is known as the overhearing problem.

X-MAC [8] (Fig. 2(b)) was designed to solve the overhearing problem of B-MAC. To prevent the sender preamble in B-MAC from occupying the channel, X-MAC continuously transmits short preambles to which the ID of a certain receiver is appended. The receiver node then replies with an early ACK when the appended ID corresponds to its own. After receiving this early ACK, the sender transmits the data packet and waits for the ACK of the data. Thus, receivers that detect unrelated short preambles can resume their sleep state soon after the end of data reception and the overhearing problem generated in B-MAC by continuous preamble transmission is solved.

3.2 Receiver-Initiated MAC Protocols

Intermittent receiver-driven data transmission (**IRDT**) [12] is a receiver-initiated MAC protocol that was developed and is actually used for products with meters [19]. IRDT complies with the receiver initiated transmission (RIT) mode in the IEEE draft 802.15.4e standards [20] and extends RIT, giving greater energy efficiency and reliability. In our previous study [12], we clarified the performance of IRDT by comparing this performance with that of the sender-initiated MAC protocol, *energy-aware adaptive LPL* [9]. As shown in Fig. 3(a), receivers that are ready to receive data transmit small packets containing their ID in order to inform the senders. A sender waits for an appropriate receiver's ID, and after acquiring this ID, the sender establishes a link with the receiver by returning a send request (SREQ). After getting a request acknowledgement (RACK) for the SREQ, the sender then transmits the data packet and finishes communication following receipt of a data acknowledgement (DACK). Another receiver-initiated Protocol is the receiver-initiated MAC (**RI-MAC**) [11] which is a simple type of RIT. In RI-MAC, the sender transmits the data packet immediately after receiving an appropriate ID (Fig. 3(b)).

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

1. Periodical ID transmissions can interfere with other nodes' communication. To avoid these collisions, receiver-initiated MAC protocols exploit channel clear assessment before transmitting an ID, and a node terminates transmission of the ID if the channel condition is busy.
2. When a receiver transmits its ID and multiple senders possess data for the receiver, transmission from different senders of multiple SREQs in IRDT or multiple data packets in RI-MAC may result in collision. To avoid these collisions, both RI-MAC and IRDT use collision detection and exponential backoff.

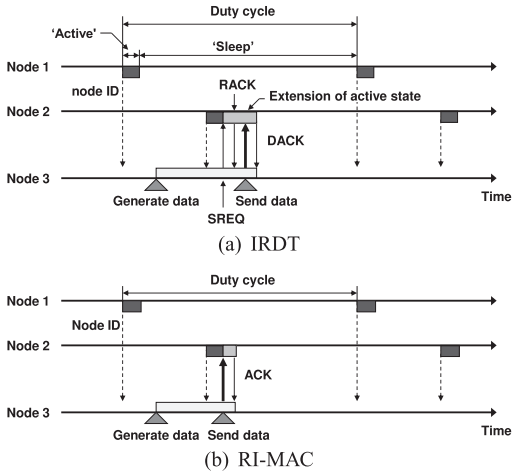
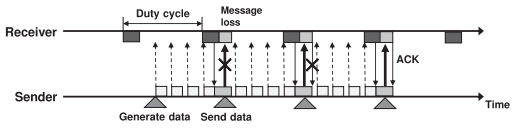


Fig. 3 Receiver-initiated MAC protocols.

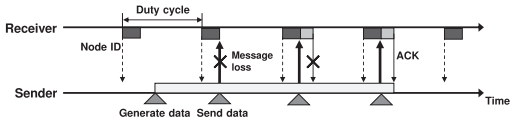
3.3 Difference in Robustness and Resilience between Sender-Initiated and Receiver-Initiated MAC Protocols

In the previous section, we defined robustness and resilience as those properties that maintain and recover performance when environmental changes occur. Since the main role of the MAC layer is one-to-one data communication, we do not consider a node failure and a link failure. Instead, we consider environmental changes due to sharp increases in traffic load, which incur congestion and message collisions as we mentioned in Sect. 1. For robustness and resilience to traffic increases in the MAC layer, changes to the receiver's condition must be detected. To maintain performance, senders should retransmit data packets only if the receiver's normal operation is confirmed, and this requires monitoring.

Although a retransmission mechanism is naturally applicable to both sender-initiated and receiver-initiated MAC protocols as shown in Fig. 4, detecting changes in the receiver's condition is nontrivial for sender-initiated MAC protocols. To monitor the receiver's condition, a sender must transmit messages to the receiver in sender-initiated MAC protocols; however, when no response is given by the receiver, the sender cannot distinguish between failure of the receiver and failure of message reception. Conversely, in receiver-initiated MAC protocols, the receiver periodically transmits its ID and shows evidence of its existence. If a sender waiting for a particular receiver's ID does not receive this ID for a period of one or more duty cycles, the



(a) X-MAC



(b) RI-MAC

Fig. 4 Retransmission-procedure examples.

sender conjectures that the receiver has failed or — since the receiver does not transmit an ID when its buffer is full — is congested. Eventually, senders in sender-initiated MAC protocols must retransmit data packets repeatedly until they achieve success, since they cannot know the receiver's condition. In receiver-initiated MAC protocols, senders will retransmit data if they receive the receiver's ID, and senders will discard their data packets if they do not receive this ID for a period of one or more duty cycles.

This procedure in receiver-initiated MAC protocols is similar to soft-state protocols. In soft-state protocols, periodical refresh messages are used, and a node that receives an intended refresh message maintains its state for as long as such refresh messages arrive. When the node cannot receive a refresh message within a given time period, it returns to its default state. As Lui et al. [3] stressed, soft-state protocols are highly robust to unanticipated fluctuations. In contrast, because senders cannot get information about a receiver's condition in sender-initiated MAC protocols, senders continue to transmit preambles as if the receivers were operating normally, similar to hard-state protocols.

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

4. Robustness and Resilience in Network Layer Protocols

Here, our focus is on robustness and resilience to route changes induced by severe environmental changes. To ensure robustness and resilience to route changes caused by node failure or energy depletion, both connectivity assurance between adjacent nodes and reachability confidence from sensor nodes to the sink node are required. To maintain performance when node failure occurs, aggressive use of detours and alternate routes is shown to be useful. In more severe cases, such as failure of the destination node, data sent from sensor nodes cannot be correctly collected and the performance of the system eventually degrades. Here, the quick response of routing tables is indispensable for resilience. In this section, we demonstrate the robustness resulting from multipath detour routing over a mesh network and the resilience resulting from soft-state management of routing tables.

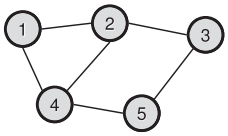


Fig. 5 Simple network model.

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

(a) Node 2's routing table

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

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

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

(b) Routing tables received by node 2 from neighbors

Fig. 6 Routing tables of node 2 in Fig. 5.

4.1 Management of Routing Tables for A Simple Distance Vector Routing

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

To begin, we explain the routing tables. In following description, we use the simple network model shown in Fig. 5. We refer to the node with a unique ID k as node k, and we define $H(m, n)$ as the hop count from node m to node n. Initially, node n registers in the routing table that $H(n, n)$ is zero. When node n receives any type of message (e.g., a HELLO message or messages in MAC layer) sent from node m, node n registers on its routing table that $H(m, n)$ is one; that is, node n refers to node m as a neighboring node. To calculate the minimum hop counts for nodes with distances greater than one hop away, each node must iteratively exchange its routing table with its neighboring nodes. This table exchange is performed with constant period, T_i. All routing tables are given a table sequence number (TSN) that is used to determine whether to exchange routing tables,

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

Fig. 7 Hop matrix table of node 2 in Fig. 5.

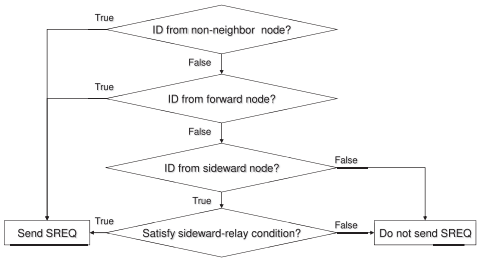


Fig. 8 Flowchart of routing function.

and TSN is incremented when the node's routing table is updated. Figure 6(a) shows an example of the routing table of node 2 in which the minimum numbers of hop counts from node 2 to all nodes have been registered. This table is calculated using routing tables of node 2's neighbors, as shown in Fig. 6(b).

Each node's corresponding hop matrix table (denoted M) is then represented by an $N \times N$ matrix (Fig. 7), where N is the number of nodes in the network. Here, we define r_{ij} as the element in row i and column j of M such that i corresponds to the receiver node ID and j to the destination node ID. Each r_{ij} is assigned an integer value that indicates the type of relays to destination node i by way of receiver node j as follows. Given sender node n whose destination is node i, if node n receives an ID from node j, node n compares $H(n, i)$ in its routing table with $H(j, i)$ in the routing table received from node j. If $H(n, i) - H(j, i) = 1$, then node n sets r_{ij} to equal to one. If $H(n, i) - H(j, i) = 0$, then r_{ij} is set equal to two, and if $H(n, i) - H(j, i) = -1$, then r_{ij} is set equal to three. Otherwise, node j is not a neighbor of node n and r_{ij} is set to zero. In addition, we define "forward", "sideward", and "backward" nodes. For node n with destination node i, if r_{ij} is one, then node j is a forward node. In a similar manner, if r_{ij} is two, then node j is a sideward node; if r_{ij} is three, then node j is a backward node; and if r_{ij} is zero, then node j is a non-neighboring node. An example of the hop matrix table of node 2 in the five node network shown in Fig. 5 is given in Fig. 7. The elements in this hop matrix table are calculated based on the routing tables shown in Fig. 6.

4.2 Detour Routing over a Mesh Network

Many studies have been conducted on routing protocols in wireless sensor networks [21]. The majority of these studies use single-path routing algorithms in which all nodes forward data to a single predetermined node according to a metric such as energy efficiency. However, in the case of a link error or node failure, controlling detours and alternative routes is considered to be effective [22]. To examine the robustness of networks, we assume a multihop wireless mesh sensor network and a hop-by-hop routing algorithm. In our single-path routing, each node forwards data packets to one of the forward nodes registered in its hop matrix table. To explain our routing procedure, we define the "routing function".

A routing function is a logic function that determines the transmission of a data packet. The flowchart of an example routing function is shown in Fig. 8. The function in this figure assumes a routing based on a minimum hop routing, where detours occur when a "sideward-relay condition" is satisfied. Thus, alternative routes exists in minimum hop routing, and a detour is employed by selecting a sideward node as the next hop. An example sideward-relay condition is that "true" is returned when a node fails to transmit a data packet to all of its forward nodes. Note that we append a TTL value in each data packet to prevent the sideward relays from causing a routing loop.

4.3 Connectivity and Reachability Management

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

Under DVR, each node has a routing table in which the hop counts are registered from all nodes in the network. When node i receives a message from node j, node i registers that $H(i, j)$ is in its routing table, and we call this a neighbor relationship. Neighbor relationships in a node's routing table can thus be maintained by probing a message. To manage the relationships, we add a time stamp to each item in a routing table. Each node waits for a message for length of time T_p every T_i, and when the node gets a message during T_p, it updates the time stamp that corresponds to the sender of the message to the current time (in this paper, T_p is always set to the same value as the duty cycle). This procedure is similar to sender-initiated MAC protocols, which periodically probe the wireless channel. To maintain neighbor relationships in sender-initiated MAC protocols, a node periodically broadcasts data packets containing its routing table to its neighboring nodes. Moreover, in receiver-initiated MAC protocols, a node has to periodically broadcast its routing table, since it probes for an ID to transmit data packets. Therefore, by adding a TSN into an ID message or a short preamble, a receiver can inform the sender whether it requires the routing table identified by the TSN. If the receiver does not need the routing table, it does not transmit an SREQ message, data packet, or data

required (DREQ) message. Since strong dependence on past conditions prevents quick responses to sudden changes, when a node does not receive a message from a neighbor within nT_i (where n is constant), the node sets the hop count associated with the former neighbor to infinity, and we call this soft-state connectivity management. After sampling, the node recalculates its routing table by using the tables received from its neighbors.

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

5. Simulation Results

We evaluate robustness and resilience in the MAC and network layers by using an event-driven simulator written in visual C++, where all results are averaged over 300-time simulations. We employ the disk model of communication between nodes, in which the strength of the radio signals does not deteriorate, and — unless packet collisions occur — a transmitted packet is assumed to be received by nodes within the communication range. In addition, our evaluation is made on safe side; if a collision with other messages occurs while a message is being received, the messages are simply discarded. The parameters in our simulation are set to the values shown in Table 1.

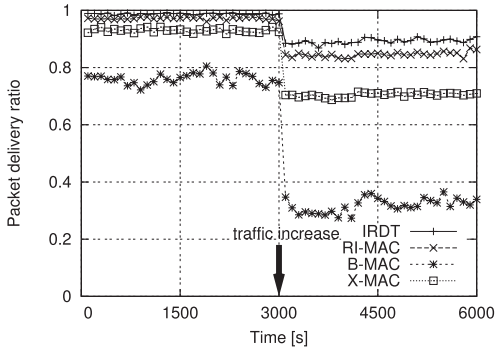
5.1 Robustness of the MAC Layer

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

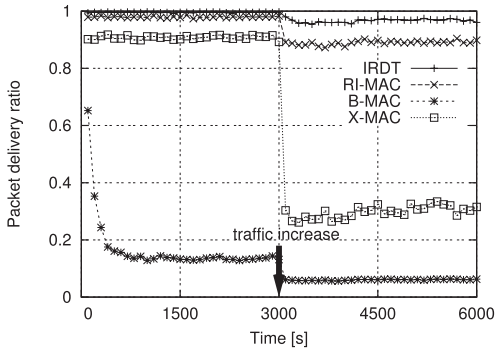
Table 1 Parameter settings.

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

Figures 9 and 10 show the packet delivery ratio and energy consumption for each 100 s of the two receiver-initiated MAC protocols (IRDT and RI-MAC) and two sender-initiated MAC protocols (B-MAC and X-MAC). Except for B-MAC, the packet delivery ratios and energy consumptions of the MAC protocols are not considerably different to each other before 3000 s. In contrast, after the addition of extra nodes, the packet delivery ratio of B-MAC and X-MAC decrease greatly due to message collisions, but the receiver-initiated MAC protocols show good robustness. This is essentially due to the receiver's link-establishment procedure in the receiver-initiated protocols. In sender-initiated asynchronous MAC protocols, since data transmission is initiated at an arbitrary timing, message collisions

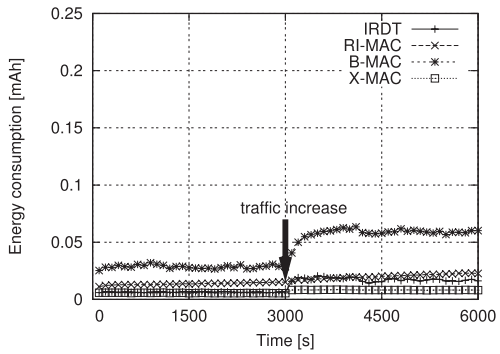


(a) T_d equals 2.0 s

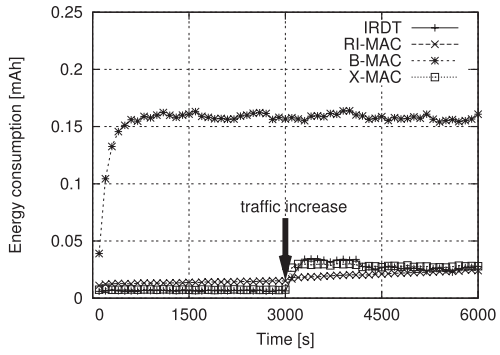


(b) T_d equals 10 s

Fig. 9 Robustness of a packet delivery ratio in the MAC layer.

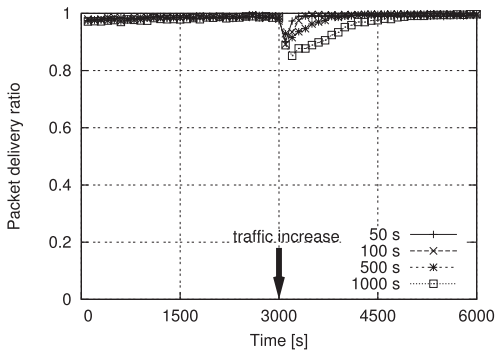


(a) T_d equals 2.0 s

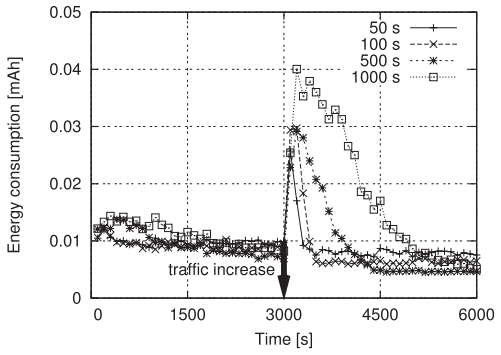


(b) T_d equals 10 s

Fig. 10 Robustness of energy consumption in the MAC layer.



(a) Packet delivery ratio



(b) Energy consumption

Fig. 11 Resilience of IRDT.

Table 2 *Robustness* of MAC layer protocols.

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

are essentially inevitable, especially when there are many senders. Meanwhile, in receiver-initiated MAC protocols, since data transmission is conducted after a receiver's ID transmission, message collisions can be avoided in some way. As above-mentioned, RI-MAC and IRDT utilize exponential backoff algorithm to establish a link between a sender and a receiver. In terms of energy consumption, we cannot easily compare the robustness among the four MAC protocols since their packet delivery ratios are different. By definition, B-MAC with T_d of 10 s has the most robust energy consumption but it consumes much more energy. RI-MAC is more robust on average due to our use of the binary exponential backoff mechanism of carrier sense multiple access with collision avoidance for data transmission. After several backoff trials, a sender drops its data packet in RI-MAC, which reduces congestion. The *Robustness* values of the MAC protocols, the relative change of 100-second average performance before and after variations defined in Sect. 2, are listed in Table 2. This shows receiver-initiated MAC protocols have about twice robustness of sender-initiated ones.

5.2 Resilience of the MAC Layer

As discussed in Sect. 3.3, we examine resilience of the receiver-initiated MAC protocols. Particularly, IRDT is evaluated using a similar simulation to that for the robustness measurements. However, here, $\lambda = 0.005$ in Poisson process and T_d is fixed to 2.0 s. To improve resilience, when the sink node (receiver node) detect congestion, its duty cycle is changed every 50 s, 100 s, 500 s, or 1000 s. Specifically, during this interval, if the rate exceeds 0.05 at the sink node that a collision is detected immediately after transmitting an ID, the sink node decreases its duty cycle by 0.2 s. Conversely, if the rate at the sink node is below 0.02, the sink node increases its duty cycle by 0.2 s. In all other cases, the sink node does not change its duty cycle.

Figure 11 shows the packet delivery ratio and energy consumption of IRDT for each 100 s and the associated *Resilience* values are listed in Table 3 (where R described in Sect. 2 is 98[%]). From the simulation results, a short interval for changing the duty cycle increases the resilience of the network performance. Note that after node additions, not only the packet delivery ratio, but also the energy consumption shows better performance due to the selection of an appropriate duty cycle.

5.3 Robustness of the Network Layer

Unlike the MAC layer, link and node failures increasingly

Table 3 *Resilience* (98% recovery) of the receiver-initiated MAC protocol, IRDT.

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

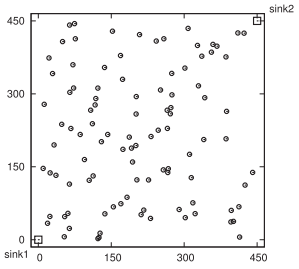
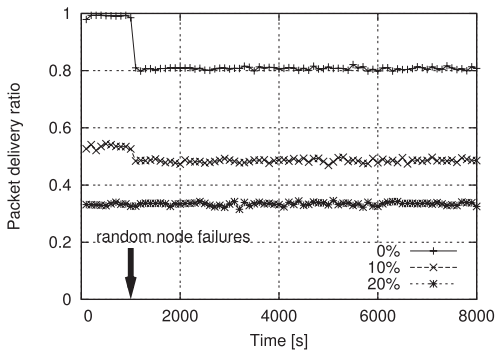


Fig. 12 An example of network model in which 100 sensor nodes and 2 sink nodes are deployed over a 450 m × 450 m square field.

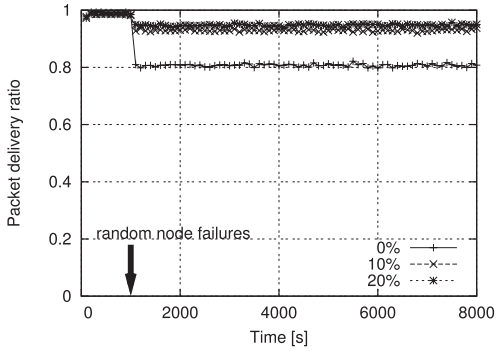
pose severe problems rather than individual link congestion in the network layer. In this section, we evaluate the robustness and resilience to node failures and we investigate the *Robustness* and *Resilience* (*R* is 90[%]) values of the packet delivery ratio and the energy consumption. Two types of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

As shown in Fig. 12, we use a network model in a square (450 m × 450 m) area to conduct our evaluation. One hundred sensor nodes, represented by circles, are randomly deployed within this area and two sink nodes, represented by squares, are positioned in the bottom left and top right corner of the network. Each sensor node generates data packets according to Poisson process with $\lambda = 0.003$ and these packets are sent to the nearest sink node by multihop relay. In our evaluation for robustness and resilience of the network layer, we use IRDT as a MAC layer protocol. The simulation commences after an initializing phase in which each node exchanges its routing table with its neighboring nodes and the simulation ends after 8000 s.

In order to investigate robustness itself, all nodes do not exchange routing tables after initializing phase, but utilize alternative and detour paths. The sideward-relay condition used for detour routing is that the sender returns an SREQ message with a fixed probability (0%, 10%, 20%). The robustness of the packet delivery ratio is shown in Fig. 13, in which multiple nodes fail at 1000 s. After the failure, the packet delivery ratio falls when nodes do not use sideward relays. However, note that after random node failures, more than 80% of the data packets are still delivered correctly because each node with a failed forward node can use alternative forward nodes. With sideward relays, the packet delivery ratio after the random failures does not considerably decrease, because each node can use a detour by controlling sideward relays. Therefore, the influence of multiple node failure is small in such cases.



(a) TTL equals the hop count

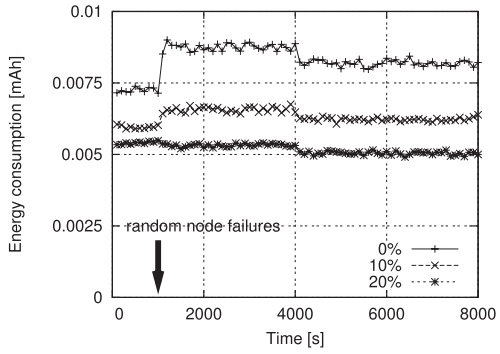


(b) TTL equals threefold the hop count

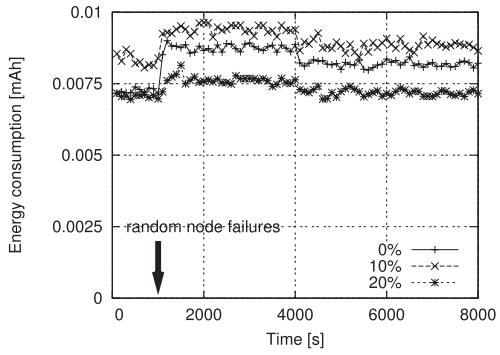
Fig. 13 Robustness of packet delivery ratio in the network layer.

In our detour routing, TTL plays a crucial role. Figure 13(a) demonstrates that the use of sideward relays degrades the packet delivery ratio. Degradation occurs because once a node transmits a data packet to a sideward node, the data packet cannot reach either sink node since TTL is set to be the same value as the hop count from the nearest sink node. However, if we set TTL equal to threefold of the hop count from the nearest sink node, over 90% of the data packets reach the sink nodes (Fig. 13(b)). For the energy consumption, the use of sideward relays intuitively expected to increase the total energy consumption, since the total hop count is increased. However, Fig. 14(a) shows the opposite result. The main reason for this contradiction is that the use of sideward relays reduces the time for idle listening of a sender node waiting for an ID from receivers. This idle listening is a dominant factor of energy consumption because the idle-listening time (100 milliseconds to seconds) is much longer than the time for message transmissions (milliseconds). When TTL becomes zero at a relay node (not the sink node), the data is discarded without idle listening. Therefore, in case TTL equals to the hop count, sideward relays shorten the time for idle listening. Conversely, in case TTL equals to threefold the hop count, energy consumption increases due to repeated sideward relays. However, 20% sideward relays consume less energy than 10% sideward relays as shown in Fig. 14(b) because the idle-listening time of a sender gets shorter as the number of multiple receiver candidates increases.

Robustness values in the network layer is listed in Ta-

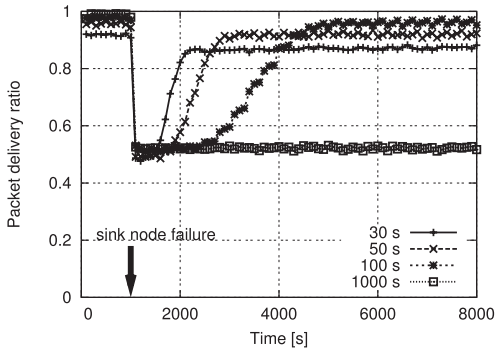


(a) TTL equals the hop count

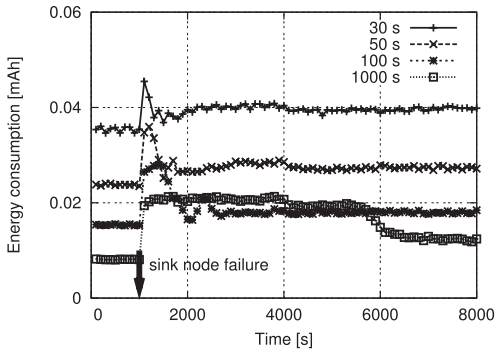


(b) TTL equals threefold the hop count

Fig. 14 Robustness of energy consumption in the network layer.



(a) Packet delivery ratio



(b) Energy consumption

Fig. 15 Resilience of the network layer.

Table 4 *Robustness* of network layer.

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

Table 5 *Resilience* (90% recovery) of the network layer.

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

ble 4 and a more positive use of sideward relays increases the performance robustness. Thus, our detour-routing algorithm has more than tripled robustness than the simple minimum-hop routing algorithm which use alternative paths.

5.4 Resilience of the Network Layer

Finally we investigate resilience to the sink-node (destination-node) failure. In general, since the refresh interval is smaller in a soft-state system, the system has greater flexibility to deal with environmental changes. Namely, with a smaller value of T_i (as described in Sect. 4.3), the network is increasingly resilient to environmental changes. Moreover, shorter T_i potentially leads to a larger overhead energy consumption. Thus, we change T_i (where n is fixed to 3) and evaluate the resilience. Note that all nodes only select a forward node for evaluation on resilience.

The accuracy of each node's routing table is highly significant in the case of sink-node failure. If a node selects the failed sink as a destination, a transmitted data packet wanders around the sink and cannot reach a sink node. As shown in Fig. 15, the packet collection ratio decreases to less than 50% right after the sink failure, because about half of the sensor nodes send data destined for the failed sink. The packet delivery ratio rapidly recovers with shorter T_i, but it does not recover completely because the traffic load of the unfailed sink node gets approximately double. Note that the energy consumption is also recovers after the failure due to the accurate route information. *Resilience* values when T_i is 30 s, 50 s, 100 s, or 1000 s are listed in Table 5. Although the recovery speed is considerably shorter when T_i is 30 s compared when with the other results, its packet delivery ratio before sink-node failure is lowest due to the overhead of table exchanges.

6. Conclusion

In this paper, we quantitatively define robustness and resilience in wireless sensor networks and evaluate them. We also discuss what brings in robustness and resilience and how improve them in the MAC layer and the network layer. Through the computer simulation experiments, we verified that receiver-initiated MAC protocols are compatible with the soft-state mechanism and they are more robust than sender-initiated MAC protocols and we show that adaptive

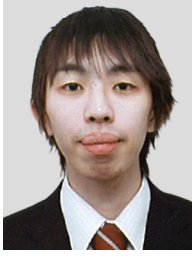
settings of duty cycles achieve good resilience in the MAC layer. As for the network layer, we present leveraging alternative and detour paths bears robustness against random node failures. Monitoring network conditions and highly-frequent exchanges of the monitored information yield great resilience. Especially, the robustness and resilience in the network layer may be able to expect the energy-saving effect. Our study supports to design robust and resilient wireless sensor networks.

Acknowledgment

This research was supported in part by "Global COE (Centers of Excellence) Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and "Grant-in-Aid for Scientific Research (C) 23500097" of the Japan Society for the Promotion of Science (JSPS) in Japan.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Commun. Mag., vol.40, no.8, pp.102–114, 2002.
[2] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," J. Network and Systems Management, vol.15, no.2, pp.171–190, 2007.
[3] J.C. Lui, V. Misra, and D. Rubenstein, "On the robustness of soft state protocols," Proc. IEEE International Conference on Network Protocols (ICNP), pp.1–11, Oct. 2004.
[4] S. Raman and S. McCanne, "A model, analysis, and protocol framework for soft state-based communication," Proc. Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM), pp.15–25, Aug. 1999.
[5] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung, "MAC essentials for wireless sensor setworks," IEEE Commun. Surveys Tutorials, vol.12, no.2, pp.222–248, 2010.
[6] Y. Wei, H. John, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," Proc. International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp.1567–1576, June 2002.
[7] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," Proc. International Conference on Embedded Networked Sensor Systems (Sensys), pp.95–107, Nov. 2004.
[8] M. Buettner, G.V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," Proc. International Conference on Embedded Networked Sensor Systems (Sensys), pp.307–320, Nov. 2006.
[9] R. Jurdak, P. Baldi, and C.V. Lopes, "Adaptive low power listening for wireless sensor networks," IEEE Trans. Mobile Comput., vol.6, no.8, pp.988–1004, Aug. 2007.
[10] I. Rhee, A. Warrier, M. Aia, J. Min, and M.L. Sichitiu, "Z-MAC: A hybrid MAC for wireless sensor networks," IEEE/ACM Trans. Netw., vol.16, no.3, pp.511–524, June 2008.
[11] Y. Sun, O. Gurewitz, and D.B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," Proc. ACM Conference on Embedded Network Sensor Systems (Sensys), pp.1–14, Nov. 2008.
[12] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, "Energy-efficient receiver-driven wireless mesh sensor networks," Sensors, vol.11, no.1, pp.111–137, Jan. 2011.
[13] L. Tang, Y. Sun, O. Gurewitz, and D.B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," Proc. IEEE International Conference on Computer Communications (INFOCOM), pp.1305–1313, April 2011.
[14] "MICA2," available at https://www.eol.ucar.edu/rtf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf
[15] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol," IEEE Netw., vol.7, no.5, pp.8–18, Sept. 1993.
[16] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session initiation protocol," RFC 2543, Internet Engineering Task Force, 1999.
[17] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.G. Liu, and L. Wei, "An architecture for wide-area multicast routing," Proc. Conference on Communications Architectures, Protocols and Applications (SIGCOMM), pp.126–135, Sept. 1994.
[18] M. Handley, C. Perkins, and E. Whelan, "Session announcement protocol," RFC 2974, Internet Engineering Task Force, 2000.
[19] T. Hatauchi, Y. Fukuyama, M. Ishii, and T. Shikura, "A power efficient access method by polling for wireless mesh networks," IEEJ Trans. EIS, vol.128, no.12, pp.1761–1766, Dec. 2008.
[20] F. Kojima, H. Harada, T. Hatauchi, M. Tanabe, K. Sakamoto, A. Kashiwagi, T. Banno, and H. Nishiyama, "Low energy MAC for non-beacon enabled PAN," available at https://mentor.ieee.org/802.15/dcn/09/15-09-0594-01-004e-low-energy-mac-for-non-beacon-enabled-pan.pdf
[21] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," Ad Hoc Networks, vol.3, no.3, pp.325–349, 2005.
[22] H. Alwan and A. Agarwal, "A survey on fault tolerant routing techniques in wireless sensor networks," Proc. 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM), pp.366–371, June 2009.



Daichi Kominami received the M.E. degrees from Osaka University, Japan, in 2010. He is currently a doctoral student at the Graduate School of Information Science and Technology, Osaka University, Japan. His research interest includes performance evaluation of wireless sensor networks.

Masashi Sugano received the M.E. and D.E. degrees in Information and Computer Science from Osaka University, Japan, in 1988 and 1993, respectively. In April 1988, he joined Mita Industrial Co., Ltd. (currently, Kyocera Mita Corporation) as a Researcher. From 1996 to 2003, he was an Associate Professor in Osaka Prefecture College of Health Sciences. From 2003 to 2005, he was an Associate Professor with the Faculty of Comprehensive Rehabilitation, Osaka Prefecture College of Nursing. From 2005 to 2012, he was with the School of Comprehensive Rehabilitation, Osaka Prefecture University, and From April 2009, he has been a Professor. He moved to School of Knowledge and Information Systems, College of Sustainable System Sciences, Osaka Prefecture University in April 2012. His current research interests include performance evaluation of computer communication network, network reliability, and ad hoc and sensor network systems. He is a member of IEEE, ACM, and IPSJ.

Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Science from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Cybermedia Center, Osaka University, and is now with Graduate School of Information Science and Technology, Osaka University since April 2004. He has more than five hundred papers of international and domestic journals and conferences. His research interests include computer communication network architecture, performance modeling and evaluation. He is a member of IEEE and ACM. He is a chair of IEEE COMSOC Japan Chapter since 2009. Also, he is now partly working at NICT (National Institute of Information and Communications Technology) as Deputy of New-Generation Network R&D Strategic Headquarters.

Takaaki Hatauchi received the B.E. degrees in Electronic Engineering from Kinki University, Japan, in 1982. He joined Fuji Electric Co., Ltd. in the same year, and belonged to the electronic developing equipment section, and engaging in the development of a wireless system. His research interests include the technology of low power consumption for the sensor network. He is a member of IEEJ.