Robustness of Receiver-driven Multi-Hop Wireless Network with Soft-State Connectivity Management

Daichi Kominami*, Masashi Sugano†, Masayuki Murata* and Takaaki Hatauchi‡
*Graduate School of Information Science and Technology, Osaka University
1-5, Yamadaoka, Suita-shi, Osaka 565-0871, Japan
Email: {d-kominami,murata}@ist.osaka-u.ac.jp
†School of Comprehensive Rehabilitation, Osaka Prefecture University
3-7-30, Habikino, Habikino-shi, Osaka 583-8555, Japan
Email: sugano@rehab.osakafu-u.ac.jp
‡Fuji Electric Systems
1, Fujicho, Hino-shi, Tokyo 191-8502, Japan
Email: hatauchi-takaaki@fesys.co.jp

***Abstract*—Energy saving and ensuring robust data collection are the big subjects in realization of wireless sensor networks. In the intermittent receiver-driven data transmission (IRDT) protocol, which aims to save energy and get high reliability, communication between nodes commences when multiple receiver nodes transmit their own IDs intermittently and a sender node receives them. In this paper, we focused on the analogy between this periodic ID-transmission and a periodic message in the soft-state management. Soft state is often considered to have robustness against failures, therefore, we introduce it to IRDT for constructing a robust network. We propose a soft-state management of routing tables in IRDT, where each node uses the periodic ID-transmission not only for communication but also for update of a routing table. By computer simulation, we show that IRDT can achieve a 43.5% improvement in robustness against a sink node failure. Moreover, we show that the receiver-driven asynchronous intermittent transmission protocol suits for the soft-state management through comparison with the sender-driven asynchronous intermittent transmission protocol.**

***Keywords*-Sensor Network; Intermittent Transmission; Soft State; Robustness**

I. Introduction

Recent advances in wireless and micro-electromechanical (MEMS) technologies direct considerable attention to ad hoc networks. Among ad hoc networks, sensor networks are expected for a wide range of applications as they have sensing ability without infrastructure. However, wireless sensor networks have critical technical problems that remain to be solved, one of which is saving energy in sensor nodes with limited battery life. Various approaches for saving energy have been proposed, for example, miniaturizing sensor nodes, media access control (MAC) with sleep control, and multi-hop routing [1-3].

In particular, considerable energy can be saved through intermittent operation, in which wireless nodes sleep and wake up periodically. We call this wake-up interval 'intermittent interval'. This power-saving operation is based on the fact that sleeping nodes consume significantly less energy than idling nodes [4]. In intermittent operation, nodes must control wake-up times in order to communicate with each other. There are two types of control method operation to control wake-up times; synchronous [5-7] and asynchronous [8-11]. For saving energy and getting scalability, the latter is superior because it doesn't need synchronization with other nodes [12]. *Intermittent receiver-driven data transmission* (IRDT) protocol, which aims to save energy and to get high reliability, makes use of the intermittent *receiver-driven* asynchronous media access control (MAC). Note that IRDT is developed and actually used for meter products [10]. Furthermore, we are proposing this technique to IEEE 802.15 Task Group 4 as a part of the standard protocol for smart meter systems [13]. In [14], we clarified the performance of IRDT by comparing with *low power listening* (LPL) protocol [9], which is a *sender-driven* asynchronous intermittent MAC protocol.

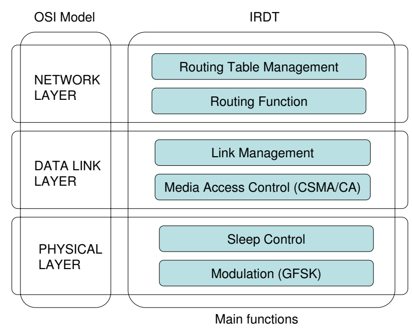


Fig. 1. Protocol layer and main functions of IRDT

IRDT provides for the functions from physical layer to network layer as shown in Figure 1. In physical layer, IRDT uses GFSK modulation to obtain tolerance for noise and manages a sleep controller which switches wireless device on and off. Data link layer controls a link management between two nodes, where each node can establish the link with multiple nodes, which enables IRDT to construct a mesh network. In asynchronous MAC, a sender node has to wait until a receiver awakes. IRDT can reduce that time by using multiple receiver nodes, which also reduces energy consumption. Specifically, each receiver sends its own ID to inform other nodes that they are ready to receive a data

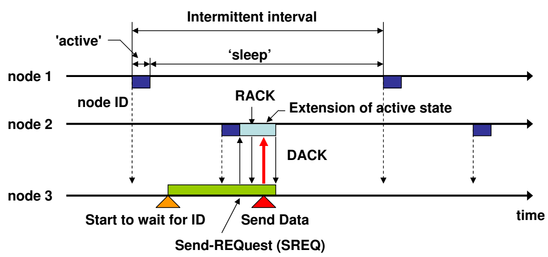


Fig. 2. Link management in IRDT

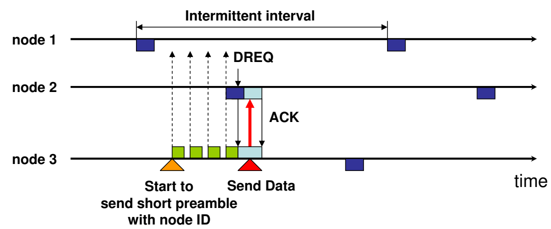


Fig. 3. Link management in AX-MAC

packet as shown in Figure 2. A sender node waits for a receiver's ID and when it acquires an ID from an appropriate receiver, it establishes a link with the receiver by returning a send request (SREQ) packet. After getting an acknowledge packet for the SREQ (RACK), the sender transmits a data packet and finishes communication following receipt of an acknowledge packet for the data (DACK). In this way, a sender node can communicate with one or more receivers flexibly, which can improve the communication reliability and save considerable energy. Therefore, in network layer, the routing protocol is designed to use multiple receiver nodes flexibly and effectively. However, the management of routing tables in IRDT is not designed to deal with the emergency. Thus, when route-changes caused by the wireless channel fluctuations or node failures occur, a system based on IRDT may not work properly long after the route-changes. In particular, connectivity to at least one sink node is very important in data collecting application which is main operation example of sensor networks.

The critical problem other than energy saving is maintaining performance against environmental changes. For that, connectivity assurance against route-changes caused by instable radio conditions and failures or energy depletions of nodes is required. Once significant route-changes occur, data sent from sensor nodes can't be collected correctly and the performance of the whole system eventually degrades. Particularly, the quick response of the routing table is indispensable in the situation where correct information of the route is absolutely necessary such as a failure of the destination node. As mentioned above, IRDT cannot deal with urgent route-changes because of its routing protocol. In this paper, we focus on the similarity between periodic ID-transmissions in IRDT and periodic messages in the soft-state management. In order to improve robustness of IRDT, we introduce the soft-state management of routing information to IRDT. Here, we define 'robustness' as the property that can maintain a packet collection ratio of a sensor network system even though critical route changes occur. Soft state is one of the methods for managing a node's state, in which node sends refresh message periodically to maintain another node's state. The node keeps its own state as long as such refresh messages arrive, but when the node can't receive a refresh message within a given time period, it changes into default state. This soft-state mechanism is generally noted to have robustness [15]. Then we evaluate robustness of IRDT with a soft-state management through a comparison with IRDT with a hard-state by computer simulation. We also compare with a sender-driven asynchronous intermittent MAC with a soft-state management and show that the receiver-driven protocol has compatibility to the soft-state management.

The rest of this paper is organized as follows. In Section II, we briefly present related work and in Section III, we show the overview of IRDT. Then, we explain the soft-state connectivity management in IRDT in Section IV and present the simulation results in Section V. Finally, we conclude our paper in Section VI.

II. Related Work

A. Intermittent asynchronous MAC protocols

B-MAC [3] is a basis of LPL protocols, where receiver nodes intermittently check the channel condition. If the channel is idle, they sleep again, and if busy, they start to be ready for data receptions. After sending the preamble packet, a sender transmits a data packet. One problem in this protocol is that long preamble packets occupy the channel, which obstructs neighbors' transmission. Moreover, many nodes waste energy due to unrelated sender's preamble, which is called overhearing problem. Other problem is that each sender node has only a specific node with which communication is possible. **X-MAC** [8] was designed to solve the overhearing problem. A sender transmits a short preamble packet continuously which includes a receiver node's ID. A receiver node replies an early acknowledge (early ACK) packet to the ID packet addressed for itself. The sender node transmits a data packet after receiving the early ACK. Receivers that detect unrelated short preamble can sleep soon after this reception finished. **Attribute-based X-MAC** (AX-MAC) protocol was proposed in [16] where sender nodes can use multiple receiver nodes by including a sender's ID in a short preamble. When a receiver gets a short preamble packet from an appropriate sender, it returns a data request (DREQ) packet as shown in Figure 3. After getting a DREQ, a sender node transmits a data packet. Comparing Figure 2 with Figure 3, these link-management procedures are fairly similar to each other, but the first packet to initiate communication is different.

B. Soft-state protocols and their robustness analysis

RSVP is a protocol for the QoS guarantee [17]. Receivers send Resv messages to their senders periodically for the reservation of the network resource and when the Resv message doesn't reach during the fixed time, the sender's state is initialized into default state. **SIP** uses the soft-state

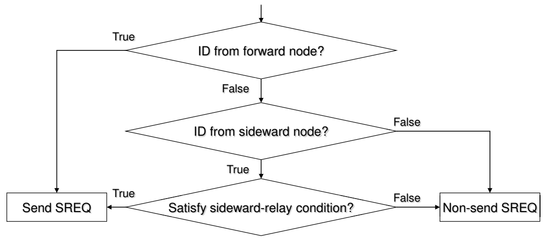


Fig. 4. An example of the routing function

session control [18]. Each node periodically sends the location information to the server for the session establishment. When the location information is not registered during the fixed time, it becomes invalid. John C.S. Lui et al. stressed the need for the quantitative evaluation of robustness in [15]. They modeled a hard-state protocol and a soft-state protocol and compared them quantitatively. They concluded that when network conditions can anticipate, hard state can attain better performance, but when unpredictable, soft state can suppresses drastic increases of the communication cost in case of some troubles. The vagueness of the concept of soft state is also recommended in [19]. Authors proposed a formal model for the soft-state communication based on a probabilistic delivery model and evaluated the tradeoff between performance overhead and robustness.

III. Overview of IRDT

A. MAC protocol

As for the MAC protocol of IRDT, its procedure of connection is explained in Section I and carrier sense multiple access with collision avoidance (CSMA/CA) is used in sending any packets. Here, the sender's decision regarding whether or not to send an SREQ packet is made on the basis of its routing protocol. Figure 2 shows the example of communication among three nodes. In this figure, node 3 gets an ID packet from node 2 and decides to send an SREQ packet to node 2 according to its own routing function.

B. Routing protocol

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

A routing table contains hop counts from a maker-node of the table to all nodes in the network and in order to make own routing table, each node has to exchange own table with its neighbors. In IRDT, all nodes periodically wake up and wait for ID packets for a short time. When a node receives an ID packet in this period, this node registers on its routing table that the hop count to the sender of the ID is one. We call this interval 'sampling interval' and this period 'sampling period' (denoted by T_i and T_p respectively).

The routing function is a logic function and the routing table is used in this function. Sender nodes decide whether to return an SREQ packet according to this function and an example of the routing function is shown in Figure 4. In this figure, 'forward node' is a neighbor node whose hop count to the destination node is smaller than own hop count. Likewise, 'sideward node' has the same hop count. The function in Figure 4 assumes the minimum hop routing, however detours are also used when the condition of sideward relay is satisfied. Easy example of the sideward-relay condition is that 'true' is returned at the probability of 25%.

IV. Soft State Connectivity Management in IRDT

Here, we present the soft-state management for improving robustness of IRDT network. First, we explain what means "state is soft or hard". We define that as the state is softer, the state should be more sensitive to environmental changes. By contraries, hard state responds slowly to changes in environment. Therefore, T_i in the following sections is small for the soft-state management.

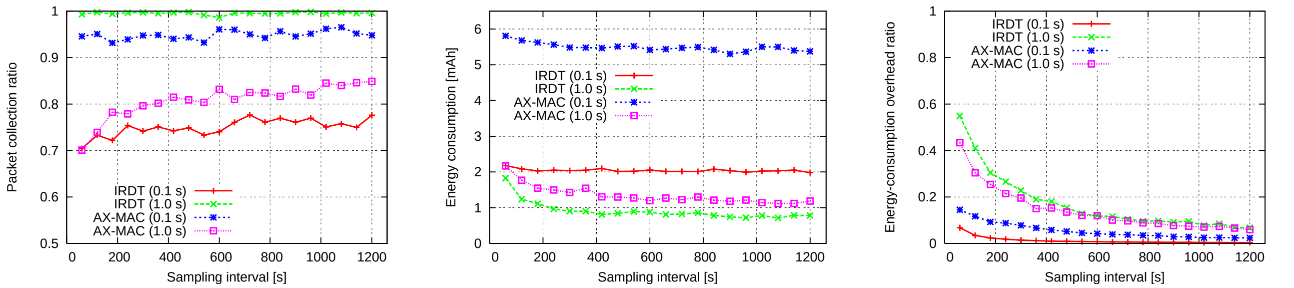
A. Soft-state management of the neighbor relationship

Each node in IRDT has a routing table in which the hop counts from all nodes in the network are registered. The neighbor relationship in the routing table can be maintained by sampling an ID packet. In the sampling period, when a node gets an ID packet, the node sets the hop count from the sender of the ID to one. Here, we propose the soft-state management of the neighbor relationship. We add a time stamp to each item in the routing table. Each node waits for an ID packet for T_p every T_i. Then, each node updates the time stamp to the current time when it gets an ID during sampling period. Note that when a node waiting for an ID packet in order to transmit a data packet receives an ID packet, the node also sets the hop count to one and updates the time stamp. When a node cannot get the ID packet from a neighbor within T_i, the node sets the hop count to the former neighbor to infinity. After the sampling, a node recalculates the hop count by using the routing tables received from neighbors.

B. Soft-state management of the neighbor's routing table

In IRDT, neighbor nodes' routing tables are necessary for each node to make own routing table. Here, we also introduce the soft-state management of the routing table. We add a time stamp to the routing table as well as the neighbor relationship management. When a node cannot get the routing table from its neighbor within T_i, the node deletes the neighbor's routing table. For maintaining the neighbors' routing tables, we also use the T_p and T_i. Then, the management of the neighbor relationship and the management of the neighbors' routing table are done at once. This is because if these managements are separately controlled, the recalculating of own routing table bring unexpected results.

Now, we describe the procedure of exchanging the routing tables. As a premise, routing tables are exchanged during sampling period. Note that only a time stamp is updated when the node doesn't need exchanging the routing table from the ID-sender. Here, the table sequence number (TSN) is added to the routing table and this number is used for the determination whether to update the neighbor's routing table. TSN is incremented when own routing table is changed and



(a) Packet collection ratio

(b) Average energy consumption

(c) Overhead of energy consumption for the soft-state management

Fig. 5. The basic performance of IRDT and AX-MAC: changing the sampling interval

TABLE I
PARAMETER SETTINGS

Parameter	Value
Transmission speed	100 kbps
Communication range	100 m
Data packet generation rate	0.003 packets/s
Current consumption (TX)	20 mA
Current consumption (RX)	25 mA
Current consumption (SLEEP)	0 mA
Packet size (ID, SREQ, DREQ, TBEX, TBNX)	24 byte
Packet size (RACK, NACK, DACK)	22 byte
Packet size (DATA)	128 byte

the latest TSN is included in an own ID packet. Specifically, tables are exchanged according to the following procedures.

1) When a node receives an ID packet during a sampling period, it checks the TSN contained in the ID packet. The node also examines the TSN of the routing table from the ID-sender.
    a) If the node has the neighbor's latest table, it updates the time stamp of the neighbor's table and returns a table non-exchange (TBNX) packet. The node that receives a TBNX packet addressed for itself also updates the time stamp and return to ID-sampling.
    b) If the node has an old table, the node and the ID-sender exchange tables each other as follows.
2) The node transmits a table exchange (TBEX) packet that demands to exchange tables. TBEX packet includes two TSN values. One is the old TSN of the neighbor's routing table from the ID-sender (denoted by TSN1) and the other is the latest TSN of own table (denoted by TSN2).
3) When the ID-sender receives a TBEX packet, it compares the TSN1 with the TSN of own table. Then, it transmits only the difference between the previous own routing table whose TSN corresponds to TSN1 and the present own routing table as a table packet. For this purpose, all nodes maintain a history of own routing table. Furthermore, the ID-sender checks the TSN2 included in the TBEX packet. If the ID-sender also has need to acquire the TBEX-sender's latest table, it adds the old TSN of the table from the TBEX-sender in the table packet.
4) After receiving the table packet, the node sends a table packet if necessary. If there is no need for transmitting a table packet, the node comes to wait for an ID packet again.

Finally, we discuss the TBEX packet collisions. When an ID packet reaches two or more nodes under sampling, TBEX packets or TBNX packets are returned simultaneously. For avoiding this collision, we add random value to T_i.

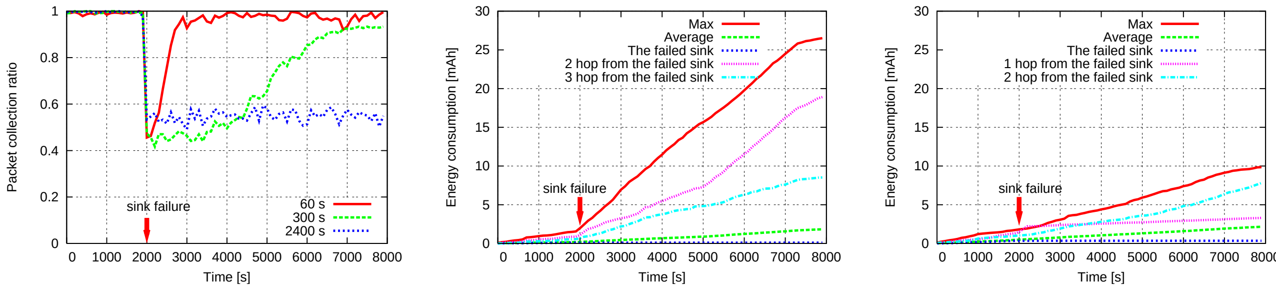
V. Simulation Results

We evaluate the basic performance and robustness of IRDT with the soft-state management by using an event-driven simulation program with visual C++. The network model shapes a square with 500 m of side length where 100 sensor nodes are randomly deployed and 2 sink nodes are set on the right top and left bottom of the network respectively. We assume that data packets are generated by each sensor node according to Poisson process and are sent to the sink node by multi-hop relay. The simulation commences after initializing phase in which each node exchanges routing table with neighbor nodes sufficiently and finishes at 8000 second. In our reception model, when collisions with other packets occur while a packet is being received, the packets are always discarded. The intermittent interval is set to 0.1 s or 1.0 s and T_p is set to the same value of the intermittent interval and other parameters are set as shown in Table I. The sideward-relay conditions of IRDT and AX-MAC is that the sender returns an SREQ packet at the probability of 25%. We also assume that the number of histories of the routing table in each node is sufficiently large.

In general, comparing T_i of the soft-state management and T_i of the hard-state management, the soft-state management uses shorter T_i in order to respond flexibly to changes in the network, namely in order to acquire robustness. However, it seems that shorter T_i leads larger overhead of the energy consumption. We identify shorter T_i as 'soft state' and longer T_i as 'hard state'. Then, we change T_i in our simulation and evaluate the performance.

A. Performance evaluation

First, in order to compare the impact on applying the soft-state management to the receiver-driven method and



(a) Packet collection ratio

(b) Energy consumption with hard state

(c) Energy consumption with soft state

Fig. 6. Robustness against the sink node's failure

the sender-driven method, we compare the IRDT and AX-MAC when both protocols use the soft-state management. We evaluate the packet collection ratio, the average energy consumption, and the overhead of the energy consumption for the soft-state management. The packet collection ratio is calculated by dividing the number of the packets received at the sink node by the number of all generated packets. The average energy consumption is obtained to divide the sum of the energy consumption of all nodes by the number of nodes and the overhead of energy consumption for the soft-state management is the proportion of the sum of energy consumed for ID-samplings and table exchanges to whole energy of nodes. In AX-MAC, each node continues to send short preamble packets for T_p every T_i which include own ID number and the TSN of own table. A receiver that catches a short preamble packet returns a TBEX packet and exchanges routing tables if necessary as is the case with IRDT.

In regard of the traffic overhead for the soft-state management, AX-MAC with 1.0 s intermittent interval degrades its packet collection ratio notably when the sampling interval is 60 s as illustrated in Figure 6(a). Because even the sink node transmits short preambles every T_i in AX-MAC, the sink node can't receive a data packet all that time. Especially, IRDT with 1.0 s intermittent interval can always get the highest packet collection ratio, more than 98% irrespective of the sampling interval.

An increase in the power consumption of IRDT and AX-MAC can be seen when the sampling interval is short as shown in Figure 5(b). The average energy consumption of IRDT and AX-MAC with 0.1 s intermittent interval don't change very much as the sampling interval increases. The reason of this is that almost all of the energy consumed in each nodes is for the periodical acitive state. Furthermore, the overhearing becomes a very serious problem in AX-MAC. The average energy consumption of AX-MAC with 0.1 s intermittent interval is at least 2.5 times higher than that of IRDT. In the case of the comparatively long intermittent interval such as 1.0 s, the energy consumption is more affected not by the overhearing but by the sampling interval T_i as shown in Figure 5(c). We set T_p to the same value of the intermittent interval, so when T_i is 60 s and the intermittent interval is 1.0 s, each node wakes up for the sampling for 1.0 s every 60 s. This consumes more energy in both of IRDT and AX-MAC.

B. Robustness evaluation

Next, we evaluate robustness of IRDT with the soft-state management against the failure of the sink node. In order to evaluate robustness, we investigate the packet collection ratio 1000 s later from the sink failure. We also define 'recovery speed' as the time that elapses since the failure occurs until the collection ratio recovers to 90% of the ratio immediately before the failure. We investigate the packet collection ratio every 100 second and obtain the recovery speed. Each sensor node selects one sink node as a destination from which the hop count is smallest among all sink nodes according to its own routing table. If two or more nearest sink nodes have the same hop counts, a node selects one of them randomly. At the 2000 s in the simulation, one sink node (denoted by 'failed sink') breaks down. The intermittent interval in IRDT is set to 1.0 s and T_i is set to three values, 60 s, 300 s, and 2400 s,

The accuracy of the routing table of each node is very significant in the case of the sink node's failure. If a node selects the disabled sink as a destination, a transmitted data packet wanders around the sink and cannot get to any sink node. In IRDT, the packet collection ratio decreases to less than 50% right after the sink failure because about half the nodes send data to the failed sink. On the collection ratio 1000 s later from the sink failure, 43.5% improvement can be attained when T_i is 60 s compared with T_i of 2400 s. Moreover, the recovery speed is much shorter when T_i is 60 s comparing with 300 s and with 2400 s as shown in Figure 6(a). When T_i is 60 s, the recovery speed is 12.8% shorter than the recovery speed when T_i is 300 s.

The steep increase of the energy consumption is seen in the hard-state management as shown in Figure 6(b). In this figure, after the sink failure, the neighbors of the failed sink consume most energy among the network because they always wait for an ID packet from the failed sink and cannot get it. The increase spreads stepwisely around the failed sink according to the propagation of the latest routing table. Moreover, routing loops occurs because of the inaccuracy of the routing table. Figure 6(c) indicates that the soft-state management can prevent the energy consumption from growing rapidly. The soft-state management makes it possible for the neighbors of the failed sink to keep their routing table later. The average energy in these figures shows that the soft-state management increases the average energy consumption in all nodes, but it can suppress the increase of the maximum energy consumption after the sink failure.

C. Discussion

First, we discuss a compatibility between IRDT and the soft-state management. IRDT uses a receiver-driven MAC protocol which has a great advantage in a long intermittent interval relative to a sender-driven method. In terms of the average energy consumption, IRDT outperforms AX-MAC when they use the soft-state management. A long intermittent interval can save energy, however, it needs a longer sampling period at least more than one intermittent interval. In the light of the packet collection ratio, IRDT with the 1.0 s intermittent interval can achieve higher performance than AX-MAC at any sampling interval. Then, we consequently say that the soft-state management is very compatible with IRDT.

Second, we discuss robustness of IRDT. IRDT is originally robust because this protocol constructs a mesh network. It is expected that the multiple link failures don't affect on the collection ratio thanks to the alternate pathway. On the critical situation that the sink node breaks down, the packet collection ratio rapidly recovers when the sampling interval is 60 s. The sharp increase of energy consumption is also be dealt with by the soft-state management. It can be said that robustness that IRDT originally has is effective and robustness against emergent changes is improved by the soft-state management.

VI. Conclusion and Future Work

In this paper, we proposed the soft-state management of the neighbor relationship and the routing table for the receiver-driven asynchronous system, IRDT. We also evaluated the basic performance and robustness of IRDT with the soft-state management through comparison with IRDT with the hard-state management and comparison with a sender-driven asynchronous system, AX-MAC. As a result, we verified that IRDT is compatible with the soft-state management in terms of the packet collection ratio and the energy consumption. As for robustness against the sink node failure, the collection ratio 1000 s later after the sink failure improves by 43.5% and the time for 90% recovery of the packet collection ratio became shortened by 12.8% of that of the hard-state management.

Now, our concern is scalability of the management of the routing table. In IRDT, the size of the routing table is proportional to the square of the number of the nodes. We improve the routing protocol of IRDT to maintain a large-scaled network.

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, "Grant-in-Aid for Scientific Research (A) 21240004" of the Japan Society for the Promotion of Science (JSPS) in Japan, "The Okawa Foundation", and "The Sekisui Chemical Grant Program for Research on Manufacturing Based on Learning from Nature".

References

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

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

[3] 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)*, Nov. 2004, pp. 95–107.

[4] "MICA2," available at http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

[5] K. Pister and L. Doherty, "TSMP: Time Synchronized Mesh Protocol," in *Proceedings of the 20th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*, Nov. 2008.

[6] Y. Wei, H. John, 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*, vol. 3, Jun. 2002, pp. 1567–1576.

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

[8] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (Sensys)*, Nov. 2006, pp. 307–320.

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

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

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

[12] J. Kim, J. On, S. Kim, and J. Lee, "Performance Evaluation of Synchronous and Asynchronous MAC Protocols for Wireless Sensor Networks," in *Proceedings of International Conference on Sensor Technologies and Applications*, Aug. 2008, pp. 500–506.

[13] 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.

[14] D. Kominami, M. Sugano, M. Murata, T. Hatauchi, and Y. Fukuyama, "Performance Evaluation of Intermittent Receiver-driven Data Transmission on Wireless Sensor Networks," in *Proceedings of the 6th International Symposium on Wireless Communication Systems (ISWCS)*, Sep. 2009, pp. 141–145.

[15] J. C. Lui, V. Misra, and D. Rubenstein, "On the Robustness of Soft State Protocols," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP)*, Oct. 2004.

[16] N. Hashimoto, H. Taka, H. Uehara, and T. Ohira, "Attribute-based Relay Control for Reduction of Latency in Duty-cycled Wireless Sensor Networks," *IEICE technical report, USN2008-44*, pp. 33–38, Oct. 2008.

[17] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new Resource ReSerVation Protocol," *IEEE network*, vol. 7, pp. 8–18, Sep. 1993.

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

[19] S. Raman and S. McCanne, "A Model, Analysis, and Protocol Framework for Soft State-based Communication," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, Aug. 1999, pp. 15–25.