Energy Saving in Intermittent Receiver-driven Multi-Hop Wireless Sensor Networks

Daichi Kominami #1, Masashi Sugano *2, Masayuki Murata #3
Takaaki Hatauchi $4, Junichi Machida $5

Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita-shi, Osaka 565-0871, Japan
{1 d-kominami, 3 murata}@ist.osaka-u.ac.jp

* *School of Comprehensive Rehabilitation, Osaka Prefecture University*
3-7-30 Habikino, Habikino-shi, Osaka 583-8555, Japan
2 sugano@rehab.osakafu-u.ac.jp

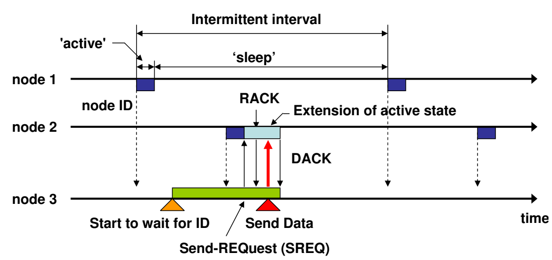
$ *Fuji Electric Systems*
1 Fujicho, Hino-shi, Tokyo 191-8502, Japan
{4 hatauchi-takaaki, 5 machida-junichi}@fesys.co.jp

***Abstract*—A major challenge in wireless sensor networks is energy saving. In the intermittent receiver-driven data transmission (IRDT) protocol, which aims to save energy, communication between nodes commences when multiple receiver nodes transmit their own IDs intermittently and a sender node receives them. Our previous research focused on the performance characteristics of IRDT when this intermittent transmission interval changes. In this paper, we analyze the probability of control packet collisions as a function of the intermittent interval and introduce a procedure for determining the proper interval that minimizes this probability. We also present that data aggregation mechanism is very suitable for IRDT and can improve the performance of IRDT. Through simulations, we show that IRDT with a proper interval and data aggregation can attain a high packet collection ratio and a large reduction in energy consumption. The proposed method achieves a packet collection ratio of more than 99% and power consumption that is 90% lower than that of the original IRDT.**

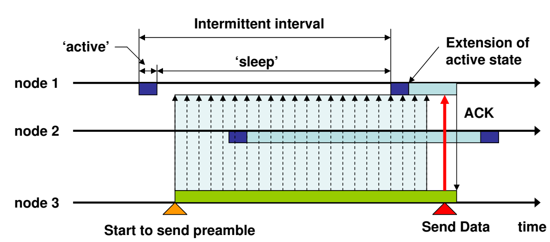
I. INTRODUCTION

Recently, due to advances in wireless and micro-electromechanical (MEMS) technologies, ad hoc networks have received considerable attention. Among ad hoc networks, sensor networks are expected to be useful in 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 [1-4].

In particular, considerable energy can be saved through intermittent operation, in which wireless nodes sleep to save power and wake up periodically to communicate with other nodes and 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 [5]. In intermittent operation, nodes must control wake-up times in order to communicate with each other. There are two types of control method for intermittent operation; synchronous [3] and asynchronous [6, 7]. For saving energy and scalability, the latter is superior in terms of the overhead for synchronization control with other nodes [8].



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



(b) Low power listening (LPL)

Fig. 1. Asynchronous intermittent transmission methods

Low power listening (LPL) protocol is a sender-driven asynchronous type of intermittent operation [4]. In LPL, receivable nodes intermittently check the channel condition. If the channel is idle, they sleep again, and if busy, they start to be ready to wait for data receptions. After receiving data packets intended for them, they return acknowledge packets.

The basic operation of LPL is shown in Fig. 1(b), where node 3 wants to send data to node 1, therefore, in order to make the channel busy, node 3 continuously sends a preamble packet for a longer time than the intermittent interval. After sending the preamble packet, node 3 sends a data packet and wait an acknowledge packet. There are many restrictions in this LPL protocol, i.e., when the intermittent interval is comparatively long to lower the duty cycle, each sender node occupies the channel by transmitting a preamble packet for a longer time than the interval, and each sender node has only a specific node with which communication is possible.

In order to overcome these drawbacks of LPL, we proposed *intermittent receiver-driven data transmission* (IRDT) [9] that is a receiver-driven asynchronous control method. IRDT lifts the restrictions of LPL, that is, it doesn't occupy the channel when the intermittent interval is long, and can select a neighbor node to communicate with from multiple neighbors. Various receiver-driven asynchronous MAC protocols have ever been proposed, i.e., in [10, 11], but most of them assume that all nodes are active and can receive packet at any time, or use multi-channel access for transmitting control packets and data packets respectively. Ref. [10] proposed the receiver-driven media access control with single channel, but didn't use intermittent operation because this protocol had no consideration for the energy consumption. In [11], authors presented the *receiver initiated cycled receiver* (RICER) which was the asynchronous MAC protocol with intermittent operation, but used two channels and time slots to avoid control packet collisions.

Our IRDT is easily implementable because it uses single-channel access, and can reduce much energy consumption due to intermittent operation. However, single-channel access causes the control packet collision. Thus, we investigated the effect of the collision problem in our previous research. In [9], we clarified the basic performance of IRDT through a comparison with EALPL [7], one of the system based on LPL and we also show the approach to avoid control packet collisions. Note that we have developed IRDT as a protocol actually implemented in meter products [12]. Furthermore, we are proposing this technique to IEEE 802.15 Task Group 4 as a part of standard protocol for smart meter system [13].

In IRDT, each node sends its own ID to inform other nodes that they are ready to receive data packets. A sender node waits for the receivers' IDs, and when it acquires an ID from an appropriate receiver, it establishes a link with the receiver and sends a data 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 select a receiver from one or more communication candidates, which can save considerable energy by shortening the sender's active time waiting for an ID packet. We show an example of intermittent operation of IRDT in Fig. 1(a). Node 3 is the sender and checks the ID from node 2 and accepts node 2 as an appropriate receiver. By comparing Fig. 1(a) and 1(b), we show that node 3 of IRDT can reduce more time in an active state than that of LPL.

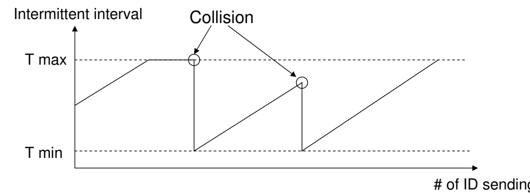


Fig. 2. Dynamic control of intermittent interval proposed in [9].

SREQ collisions are a critical problem in IRDT as discussed in our previous research [9]. As shown in Fig. 1(a), the sender node responds with the SREQ packet when an ID from an appropriate node arrives. If the appropriate node for more than one sender node sends IDs, the sender nodes simultaneously receive the IDs and return SREQs. Therefore, multiple SREQs collide with each other. Previous results suggest two approaches to resolve this problem. The first is setting the intermittent interval to a proper value, such that SREQ collisions hardly occur. The second is decreasing the frequency of data sending. We focused on the first approach and used the simple and effective approach in [9], in which the intermittent interval is set dynamically to handle the communication load according to SREQ collisions. In the dynamic setting of the intermittent interval, nodes set their own intermittent intervals to T_{min} when they detect a packet loss after waiting for an SREQ. If SREQ collisions are not detected, the nodes gradually increase their intervals to T_{max} (Fig. 2). However, this reactive method starts only after the occurrence of SREQ collisions and causes the recurrence of SREQ collisions and increase in the ID-waiting time as nodes return their intermittent intervals to T_{max}. In addition, since this packet loss can be induced by not only congestion which causes SREQ collisions but also channel errors, the intermittent interval may change regardless of the node's communication load.

In this paper, we make use of the first and the second approaches. In case nodes have information on the network topology and can estimate their own load, a proper value of the intermittent interval is expected to avoid SREQ collisions. As for the latter approach, data aggregation enables the number of sent data packets to be decreased [14]. This clearly reduces the number of SREQ sent to nodes' receivers and has a great effect on reduction of the SREQ collisions as described in Section III. We clarify the impact of a proper intermittent interval and data aggregation in IRDT respectively by simulation. In addition, we compare the performance of IRDT with that of LPL, when both method use a proper interval and data aggregation.

The rest of this paper is organized as follows. In Section II , we discuss the procedure for determining a proper intermittent interval in IRDT. In Section III, we describe data aggregation in IRDT. We present the simulation results in Section IV and our conclusions in Section V.

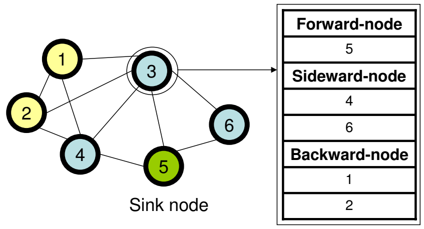


Fig. 3. Classification of neighbor nodes at node 3

II. Avoidance of control packet collisions

In this section, we describe how the intermittent interval affects the probability of packet collisions and the procedure for determining a proper intermittent interval that minimizes this probability.

The change of intermittent interval affects the following three respects:

1) Probability of SREQ collisions
This is the probability that when a node sends an ID, multiple nodes return SREQ packets simultaneously. A longer intermittent interval increases this probability. If SREQ collisions occur, the energy consumption of the sender nodes increases because of retransmissions. Furthermore, such SREQ collisions can occur repeatedly.
2) Probability of ID collisions
This probability corresponds to the likelihood that the ID packets periodically sent by all neighbor nodes collide against other packets. A shorter intermittent interval increases this probability. As in the case of SREQ collisions, retransmissions increase energy consumption.
3) Wating time of sender nodes for ID
In IRDT, most energy is consumed when the node is waiting to receive an ID packet. A shorter intermittent interval of the node decreases the waiting time of sender nodes, but increases the node's duty cycle. Conversely, a longer intermittent interval increases the waiting time of sender nodes and decreases the node's duty cycle.

Next, we obtain the optimal intermittent interval analytically, which minimizes the sum of the SREQ collision and ID collision probabilities. We refer to this intermittent interval as the "proper interval (denoted by T^*)".

A. Analytical derivation of control packet collision probability

In the analysis of the control packet collision probability, we introduce the following assumptions.

- All nodes have complete information on network topology and a static routing table based on this information. Then, all nodes classify their neighbors into "forward-node", "sideward-node", and "backward-node". A "forward-node" ("backward-node") is a neighbor node that has the smaller (larger) number of hops from the sink node ; "sideward-node" has the same number of hops. For example, in Fig 3, node 3's neighbors are classified.
- Each sensor node generates a data packet according to Poisson process with intensity λ and sends the data to the sink node. In addition, when they forward the data, they always select forward-nodes and all forward-nodes are equally likely to be chosen as the receiver.
- Each node sends ID packet at the regular intermittent interval denoted by T. Moreover, all nodes use CSMA/CA when sending any kind of packet. When packet collisions occur, the packets are always discarded.

From the above assumptions, we can calculate $G(R)$ which is the approximate average number of data packets that node R receives in one second. $G(R)$ depends on the number of node R's backward-nodes and the backward-nodes' traffic load. Here, we define $N_b(R)$ as the set of backward-nodes of node R and $|N_f(n)|$ as the number of forward-nodes of node n. The probability that one node (denoted by n) of them select node R for their receiver is $|N_f(n)|$, therefore, $G(R)$ is expressed as following:

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

SREQ collisions occur when two or more neighbor nodes send SREQ packets simultaneously. We assume that all nodes use CSMA/CA mechanism, which can reduce SREQ collisions, but SREQ collisions may occur unless there are no hidden nodes because SREQ packets can be returned just at once. In CSMA/CA mechanism with exponential backoff, the number of time slots that each node chooses randomly is 2^{BE} where BE is the moderate integer value. Here, we assume that the number of data packets that node R receives from each of its backward-nodes is equal. Therefore, the probability that each node returns an SREQ when receiving an appropriate ID can be expressed as $1 - e^{-G_b(R)T}$ where $G_b(R)$ is $\frac{G(R)}{|N_b(R)|}$ and doesn't return can also be expressed as $e^{-G_b(R)T}$. P_{SREQ}, the probability that SREQ collisions occur, is also the probability that at least one node of node R have a data packet and more than one neighbor of node R has data packet but CSMA/CA mechanism can avoid an SREQ collision when node R send an ID packet. Thus, P_{SREQ} can be calculated as following:

$$P_{SREQ} = 1 - \sum_{k=0}^{|N_b(R)|} C(R,k) e^{-(|N_b(R)|-k)G_b(R)T} (1 - e^{-G_b(R)T})^k \tag{2}$$

where $C(R,k)$ means the number of combinations of the different k nodes out of $N_b(R)$ which considers the hidden-terminal problem under CSMA/CA.

Here we consider only for the case where the value of k is smaller than three because the term $e^{-(|N_b(R)|-k)G_b(R)T}(1 - e^{-G_b(R)T})^k$ is so small that we can ignore it with large k. $C(R,k)$ is defined as below:

$$C(R,k) = \begin{cases} 1 & (k=0) \\ |N_b(R)| & (k=1) \\ h(R)\frac{2^{BE}-1}{2^{BE}} & (k=2) \end{cases} \tag{3}$$

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

Next, we target in collisions of ID packets at node R. An ID packet collision occurs when ID packets are sent by the neighbors of node R while node R is receiving an SREQ or data packet. Note that we don't need to pay attension to the backoff timeslot of CSMA/CA as discussed in P_{SREQ} because ID packets are hardly transmitted coincidentally. Here, the average number of hidden nodes of node R, $H(R)$, when node R is receiving the SREQ or data is defined as follows.

$$H(R) = \frac{1}{|N_a(R)|} \sum_{n \in N_a(R)} h(R, n) \tag{4}$$

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

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

$$P_{ID} = \frac{T_r H(R)}{T} \tag{5}$$

B. Procedure for determining a proper interval

To determine the proper interval, we modify Eq. (2). Equation (2) shows the probability that SREQ collision occurs when one ID packet is sent by node R, and Eq. (5) shows the probability of ID collision when node R receives one SREQ or data packet. Therefore, we introduce P'_{SREQ}, the product of P_{SREQ} and $(G(R)T)^{-1}$, which corresponds with the SREQ collision probability for receiving one SREQ or data packet (Eq. 6).

$$P'_{SREQ} = \frac{1 - \sum_{k=0}^{2} C(R,k) e^{-(2-k)G_b(R)T} (1 - e^{-G_b(R)T})^k}{G(R)T} \tag{6}$$

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

$$P_{CTRL} = P'_{SREQ} + P_{ID} \tag{7}$$

Unfortunately, an explicit expression of T^* which minimizes Eq. (7) cannot be represented, but instead, we can compute the approximate value of T^* by calculating the minimum value of the sum and then T^* every 10 ms in the semiopen interval (0.0 s, 2.0 s]. However, if we cannot determine the minimum value, we use the intermittent interval that minimizes the sum of the energy consumption for the ID-waiting time of the backward-nodes and for own intermittent operation (denoted by T_m). The total energy consumption of the neighbor nodes of node R per second can be represented as $\sum_{n \in N_b(R)} \frac{E_w T}{|N_f(n)|+1} G(n)$ where E_s and E_w are the transmission power and the reception-standby power, respectively.

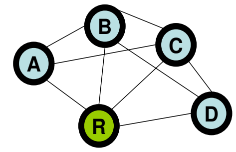


Fig. 4. Network model

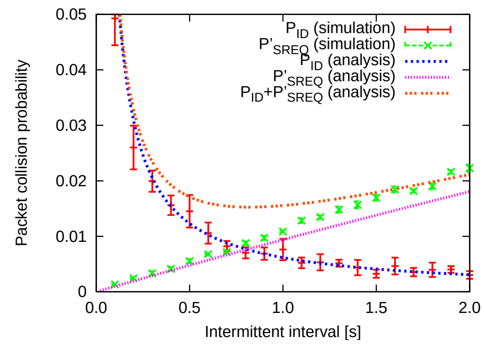


Fig. 5. Probability of control packet collisions

On the other hand, the energy consumption of the intermittent operation of node R is $\frac{E_s T_p + E_w T_s}{T}$, where T_p is transmission time for an ID packet and SREQ waiting time after sending an ID. T_m, which minimizes the sum of these, is expressed as follows.

$$T_m = \sqrt{\frac{(E_s T_p + E_w T_s)}{E_w \sum_{n \in N_b(R)} \frac{G(n)}{(N_f(n)+1)}}} \tag{8}$$

Figure 5 shows the results of the analysis and simulation of control packet collisions for the network topology shown in Fig. 4, where $\lambda = 0.024, BE = 3$ and the error bar corresponds to the 95% confidence interval. From the results shown in Fig. 5, it can be concluded that both the analysis and the simulation of P_{ID} and P_{SREQ} roughly correspond, so our analysis roughly seems to be good. But for P_{SREQ}, as the intermittent interval become longer, the simulation result surpasses the analytical result due to the assumption that CSMA/CA can always prevent packet collisions regardless of whether or not there are hidden nodes. In fact, CSMA/CA cannot completely avoid packet collisions, and SREQ collisions tend to occur as more backward-nodes have data packets. Therefore, when the packet generation rate is high, SREQ collisions occur more frequently. In an actual multi-hop network, a node sends data packets not only to the forward-node but also to the sideward-node and the backward-node. This is because P_{SREQ} in an actual network is difficult to estimate. Moreover, the actual average number of data packets received in one second increases due to retransmissions. Then, P_{ID} decreases slightly since the node with data packet does not send ID packets and P_{SREQ} can increase.

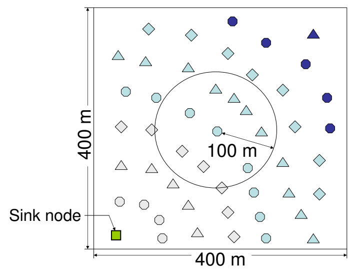


Fig. 6. Network model

TABLE I
PARAMETER SETTINGS

Parameter	Value
Sending current	20 mA
Waiting current	25 mA
Sleeping current	0 mA
ID packet size	40 byte
SREQ, RACK, DACK packet size	26 byte
Data packet size	128 byte
Transmission speed	100 kbps

III. DATA AGGREGATION IN IRDT

Data aggregation can reduce the number of data packet transmissions of each node. We assume that when a node aggregate m data packets, the size of the data packet increases m times. Therefore, a larger m effectively decreases $G(R)$ in Eq. (2) then P_{SREQ} also decreases, but unfortunately, increases T_r in Eq. (5) and also P_{ID}. Then, we show this tradeoff in the next section.

Here, we show the great effect of data aggregation with sideward-nodes. In [9], we presented a method in which each node gives priority to forward-nodes as appropriate receivers. When using data aggregation, however, relay with sideward-nodes is more effective. SREQ collisions occur when two or more neighboring nodes that have the same hop count own data packets. If data aggregation is done among these nodes well, only one node has the aggregated data packet and then no SREQ collision occurs. Moreover, data aggregation can resolve repeated SREQ collisions which occur when there is only one forward node such as a sink node. We show this repeated SREQ collisions causes much more energy consumption in our previous research. If IRDT doesn't use data aggregation, repeated SREQ collisions continue until an sending timer expires. Specifically, when data aggregation is possible, we extend the priority of the forward-nodes to the sideward-nodes that have data packets. Whether or not the sideward-nodes have data packets can be obtained by including this information in the ID packets.

We limit the size of aggregated data packets for the reasons noted above, namely, a large value of m increases P_{ID} and the channel occupation time. We include the number m in the ID packets to inform the receiver nodes, which can also be used if the sideward-nodes have data packets. Using this information prevents the data packet size of the nodes from exceeding m times the original data size as a result of aggregation.

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

1) Continuing intermittent operation for a fixed time: Sender nodes immediately change into the ID-waiting state in IRDT when they receive or generate a data packet. At that time, the data aggregation can be achieved by continuing their intermittent operation to receive data packets until the fixed time passes without changing into the ID-waiting state. The node changes to the ID-waiting state when the size of the aggregated data packet reaches a certain predetermined size or a certain period of time passes.
2) Transmitting an ID packet in ID-waiting state: In the current IRDT, the node that has a data packet waits for forward-nodes' ID packets, but does not send an ID packet. It becomes possible to receive data packets by changing its radio to the transmission mode and transmitting an ID packet until receiving an appropriate ID.

The first method decreases the data transmission frequency with aggressive data aggregation and the second method aggregates data without increasing the delay time. In this paper, we focus on the first method to achieve greater energy savings.

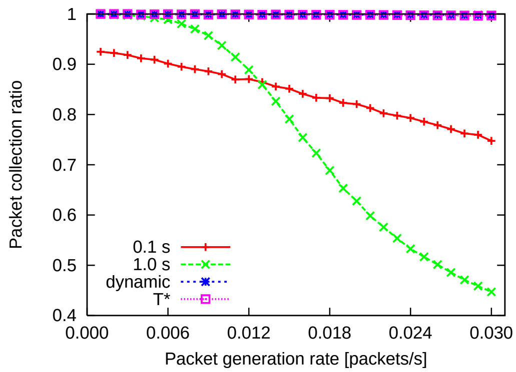
IV. SIMULATION RESULTS

In this section, we clarify the performance characteristics of IRDT with (a) the proper interval setting or (b) the introduction of data aggregation. In addition, we compare the performance characteristics of IRDT with those of LPL when both (a) and (b) are introduced.

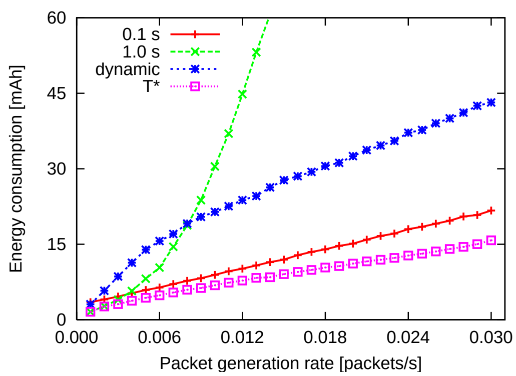
We use the network model shown in Fig. 6, in which one sink node and 49 sensor nodes are deployed over 400 m-square. In this figure, the sink node is represented by a square and other shapes denote sensor nodes. The communication range of nodes is 100 m and the sensor nodes shown in the figure with the same shape and color have the same number of hops from the sink node. The main parameters are shown in Table I and other parameters are the same as those used in [9]. Sensor nodes other than the sink node on the network generate data packets according to Poisson process and transmit the data to the sink node by multi-hop relay. We investigate the packet collection ratio, that is, the number of packets received at the sink node divided by the number of all generated packets. We also investigate the energy consumption of the highest loaded nodes, denoted by the maximum energy consumption, and the average energy consumption for all nodes.

A. Performance using a proper intermittent interval

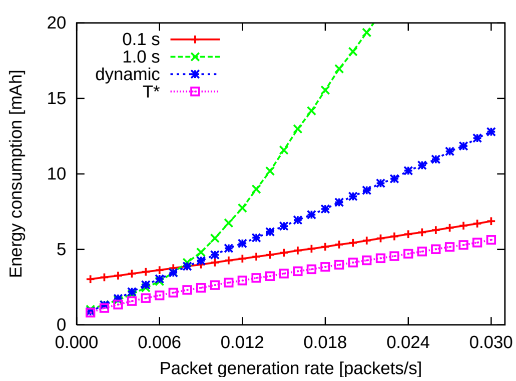
The collection ratio and the power consumption when each node sets own intermittent interval to 0.1 s, 1.0 s, and T^* obtained in Section II-B are shown in Fig. 7. We also show the results when each node sets its intermittent interval dynamically as discussed in [9]. As shown in Fig. 7(a), when the intermittent interval is 0.1 s, IRDT cannot achieve even a



(a) Packet collection ratio



(b) Maximum energy consumption



(c) Average energy consumption

Fig. 7. Performance using T^*

95% collection ratio even when the packet generation rate is 0.001, but can always achieve a collection ratio of more than 70%. By contrast, at an intermittent interval of 1.0 s, IRDT can attain an approximately 100% collection ratio when the packet generation rate is low, although it cannot attain a collection ratio of even 45% at relatively high packet generation rates.

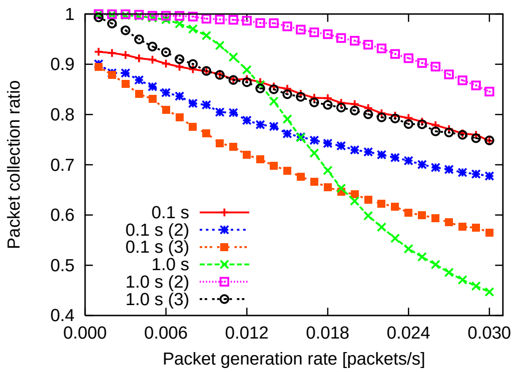
These results can be explained with Eq. (2), (5). From the Eq. (2) and Fig. 5, P_{SREQ} and the repeated SREQ collisions mentioned in section II-B increase as the intermittent interval becomes longer, and the collection ratio for the high packet generation rate at 1.0 s results in the lower value. Meanwhile, from Eq. (5), P_{ID} is a constant value when the intermittent interval is fixed. This causes a decrease in the collection ratio according to a constant probability, which occurs notably when the intermittent interval is short. Remarkably, these results can be improved by both setting the intermittent interval to T^* and the dynamic setting of the intermittent interval. As a result, the collection ratio is always near 100% over the entire range of the packet generation ratio in Fig. 7(a).

An increase in the power consumption can be suppressed by using an interval of T^* as shown in Fig. 7(b), 7(c) due to prevention of control packet collisions. Additionally, preventing repeated SREQ collisions and shortening the ID-waiting time can decrease energy consumption. An intermittent interval of T^* results in a 41% reduction in the maximum energy consumption compared with dynamic setting of intermittent interval at a packet generation rate of 0.001. A reduction in maximum energy consumption of 64% is also achieved at a packet generation rate of 0.030. Reductions in the average energy consumption of 10% and 44% are attained when the packet generation rates are 0.001 and 0.030, respectively.

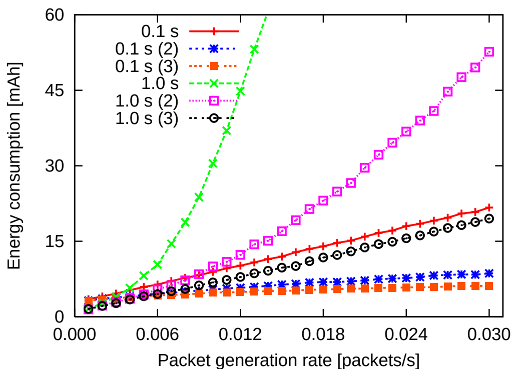
B. Effect of data aggregation

The performance of IRDT when the data aggregation function is introduced is shown in Fig. 8; the number in the label denotes how many data packets can be included in one aggregated data packet. Immediately after the reception or generation of data, each node waits for 5.0 s for aggregation without forwarding. When the intermittent interval is 1.0 s, the packet collection ratio improves with data aggregation up to two data packets, but becomes worse with aggregation of three data packets. At an intermittent interval of 0.1 s, data aggregation always decreases the collection ratio since the large data packets are likely to collide with the ID packets. Moreover, the loss of the aggregated data packet greatly decreases the collection ratio. As mentioned above, when the intermittent interval is comparatively long, data aggregation up to two is effective in terms of the avoidance of SREQ collision, although aggregation of three or more is disadvantageous. However, at a short intermittent interval, the data aggregation is ineffective due to the increase of P_{ID}.

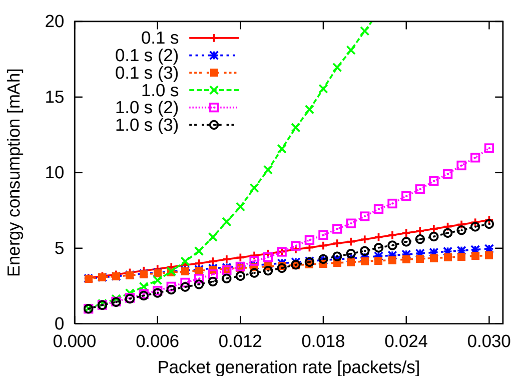
The maximum and average energy consumption decreases as the number of aggregated data increases. However, when the packet generation rate is low, data aggregation seldom occurs during the waiting time of 5.0 s and the energy consumption does not improve considerably. Note that when the intermittent interval is 0.1 s, the slight difference between 0.1 s (2) and 0.1 s (3) in Fig. 8(b), 8(c) indicates that the increase in retransmissions due to ID collisions increases the energy consumption. For aggregation up to three data packets, when the packet generation rate is 0.030, an 83% reduction in maximum energy consumption and a 77% reduction in



(a) Packet collection ratio



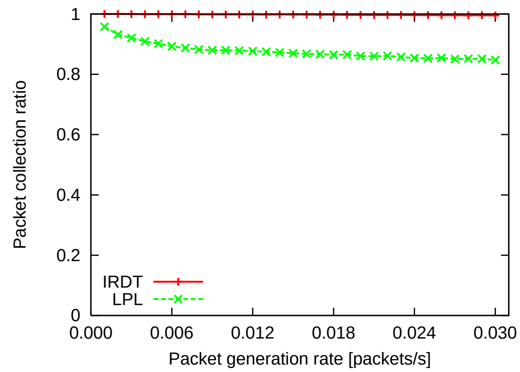
(b) Maximum energy consumption



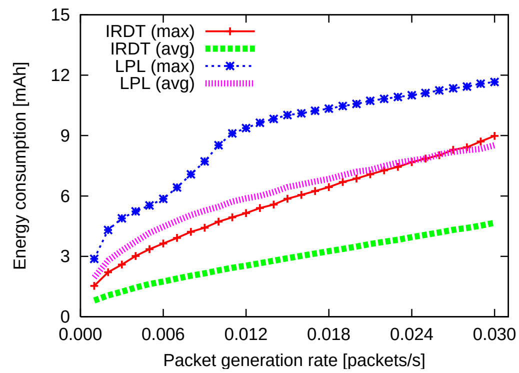
(c) Average energy consumption

Fig. 8. Performance with data aggregation

average energy consumption can be attained at an intermittent interval of 1.0 s. Moreover, maximum and average energy consumption is reduced by 70% and 47% at an interval of 0.1 s respectively. These improvements are achieved in particular due to forwarding data to sideward-nodes, which effectively suppresses SREQ collisions in the sink-neighbor nodes.



(a) Packet collection ratio



(b) Energy consumption

Fig. 9. Performance using data aggregation and T^*

C. Comparison with LPL

We compare the performance of IRDT with the proper interval and data aggregation with that of LPL (Fig. 9), where the data aggregation is up to two data packets to prevent the packet collection ratio from decreasing. To conduct a fair comparison, LPL also uses data aggregation and an appropriate intermittent interval that minimizes the energy consumption such as Eq. (8). However, due to LPL's MAC layer protocol, the intermittent interval are limited to 8 values (10, 20, 50, 100, 200, 400, 800, 1600 ms) [4]; therefore, LPL uses the closest value to the appropriate interval from these eight values.

The results show that IRDT attains a higher collection ratio than LPL. In addition, IRDT has lower maximum and average power consumption at any time in Fig. 9(b). Maximum energy consumption can be reduced to 33% to 52% and average energy consumption can be reduced to 38% to 54%. Moreover, a 92% reduction of the maximum energy consumption and an 84% reduction of the average energy consumption are achieved compared with the original IRDT at an intermittent interval of 1.0 s. In particular, it is important to lower the maximum energy consumption for the long term operation of the network; therefore, the avoidance of the control packet collisions is very effective.

V. Conclusion

In this paper, we investigated the relation between the probability of control packet collisions and the intermittent interval in a receiver-driven asynchronous IRDT system. We presented a procedure for obtaining the proper interval that minimizes the probability of control packet collisions. Moreover, we showed that data aggregation can further reduce this probability. We examined the efficacy of the proper interval and data aggregation through a comparison with the original IRDT and LPL, which is a sender-driven asynchronous system, by using computer simulation. As a result, a reduction in the maximum and average energy consumptions of more than 30% compared with LPL could be obtained. Furthermore, compared with the original IRDT, the maximum energy consumption was reduced by 92% and the average energy consumption was reduced by 84%. Load balancing to reduce maximum energy consumption and a more detailed simulation considering node failure, energy depletion, and wireless channel conditions is our future works.

References

[1] X. Du, Y. Xiao, and F. Dai, "Increasing Network Lifetime by Balancing Node Energy Consumption in Heterogeous 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 Transactions on Networking*, vol. 15, no. 1, pp. 93–104, Feb 2007.

[3] 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*, Nov 2008.

[4] 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)*, pp. 95–107, Nov 2004.

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

[6] 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 '06)*, pp. 307–320, Nov 2006.

[7] 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.

[8] 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*, pp. 500–506, 2008.

[9] 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'09)*, pp. 141–145, Sep 2009.

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

[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*, pp. 3769–3776, vol. 7, Jun 2004.

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

[13] F. Kojima and et al, "Low energy MAC for non-beacon enabled PAN," https://mentor.ieee.org/802.15/dcn/09/15-09-0594-01-004e-low-energy-mac-for-non-beacon-enabled-pan.pdf.

[14] I. Chalermek, E. Deborah, G. Ramesh, and H. John, "The Impact of Data Aggregation in Wireless Sensor Networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pp. 457–458, Jul 2002.