An Autonomous Data Gathering Scheme Adaptive to Sensing Requirements for Industrial Environment Monitoring

Yoshiaki Taniguchi*, Naoki Wakamiya*, Masayuki Murata*, Takashi Fukushima†
*Graduate School of Information Science and Technology, Osaka University, Japan
Email: {y-tanigu,wakamiya,murata}@ist.osaka-u.ac.jp
†Production Systems Research Laboratory, Kobe Steel, Ltd., Japan
Email: fukushima.takashi@kobelco.com

***Abstract*—In wireless sensor networks (WSNs), the frequency of sensing and data gathering depends on application requirements and surrounding conditions. In this paper, we propose a data gathering scheme adaptive to sensing requirements for WSNs composed of sensor nodes with multiple sensing capabilities. To accomplish self-organizing control, we adopt the response threshold model for adaptive sensing task engagement and the pulse-coupled oscillator model for energy-efficient transmission and sleep scheduling. Through preliminary simulation experiments, we confirm that autonomous and energy-efficient data gathering can be accomplished satisfying dynamically changing sensing requirements.**

I. Introduction

The development of low-cost microsensor equipments having the capability of wireless communication has caused wireless sensor network (WSN) technology to attract the attention of many researchers and developers. It is possible to obtain information on behavior, condition, and position of elements in a local or remote region by deploying a network of battery-powered sensor nodes there. Each node in such a WSN has a general purpose processor with limited computational capability, a small memory, and a radio transceiver. Due to several restrictions including limited battery capacity, random deployment and a large number of unstable nodes, a communication mechanism should be energy-efficient, adaptive, robust, fully distributed, and self-organizing.

Sleep scheduling certainly plays an essential role in saving energy consumption for longer lifetime and guaranteeing communication delay for mission critical applications [1]. When we consider energy-efficient scheduling for periodic data gathering from all sensor nodes, for example, message transmission first begins at the edge of a WSN. At the timing when those nodes at the farthest hop distance from the base station (BS) transmit messages, nodes which are closer to the BS by one hop, i.e. next hop nodes, are scheduled to wake up to receive the messages. After a certain period, they also transmit messages, and at this time, their next hop nodes are awake and ready to receive the messages. At the same time, the farthest nodes also receive the messages and go to sleep. As a consequence of such scheduling, we see concentric traveling waves of message propagation from the edge to the BS.

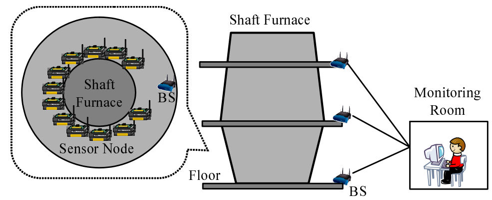


Fig. 1. Monitoring of shaft furnace of steel plant

If the interval of data gathering is fixed, such sleep scheduling to wake up nodes from the edge of a WSN to a BS is effective in saving energy consumption and reducing data gathering delay [2-4]. However, in some classes of applications, a node needs to change its sensing frequency to monitor the region or target more frequently when it detects unusual conditions and phenomena [5]. Furthermore, the number of nodes which monitor and report the phenomena should be regulated in accordance with its criticality and importance. For example, in this work we consider the deployment of temperature and CO gas sensors on the surface of the shaft furnace in a steel plant as shown in Fig. 1. Temperature changes slowly in the order of hours and once it increases, it stays high for a long period. Therefore, nodes are required to monitor temperature more frequently when changes are detected, while they can decrease the sensing frequency under stable conditions. On the other hand, CO gas suddenly may appear, move fast, and dissipate. Therefore, nodes are required to monitor CO gas more frequently than temperature when CO gas is detected.

To tackle the above-mentioned problem, we propose in this paper a data gathering scheme adaptive to dynamically changing sensing requirements. We adopt biologically inspired models to accomplish self-organizing control. In our mechanism, nodes operate on traveling wave-based periodic data gathering at regular intervals. We organize traveling waves in a self-organizing fashion by adopting a pulse-coupled oscillator model, which explains the biological synchronization observed in a group of flashing fireflies [6]. We assume that the regular sensing frequency is the same among sensors of different types. When conditions of the sensing target, such

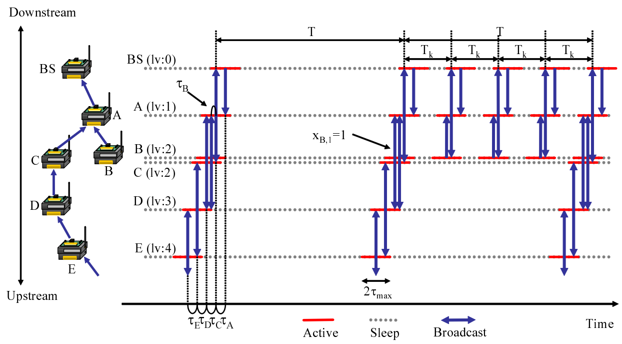


Fig. 2. Broadcast timing of proposed scheme

as temperature and gas concentration, change at a certain point, surrounding nodes decide whether to monitor the target more frequently or not depending on the need for sensing. To regulate the number of nodes engaged in frequent monitoring of the target in a self-organizing way, we adopt a response threshold model [7], i.e. a mathematical model of division of labor and task allocation in social insects. Once a node considers to monitor the target more frequently, it increases its sensing frequency since the obtained sensor data are forwarded to the BS at a higher frequency, nodes on the path to the BS also adapt to the new frequency. As a result, one or more traveling waves emerge in the WSN. In this paper, through preliminary simulation experiments, we confirm that autonomous and energy-efficient data gathering can be accomplished satisfying dynamically changing sensing requirements.

The rest of this paper is organized as follows. We propose a data gathering scheme adaptive to sensing requirements in Section II. Then, we show simulation results in Section III. Finally, we conclude this paper and describe future research work in Section IV.

II. A Data Gathering Scheme Adaptive to Sensing Requirements

In this section, we propose a data gathering scheme adaptive to sensing requirements. An example of the behavior of our scheme is illustrated in Fig. 2. In our scheme, all nodes have k_{max} sensing devices, for example, temperature, light, and gas concentration, and monitors condition of k_{max} sensing targets. For example, a sensing target of a temperature sensor is the temperature in the vicinity of the sensor, so we use sensor and sensing target interchangeably hereafter. Node i has two sensing states, *normal state* and *frequent state* for each sensor k ($1 \leq k \leq k_{max}$), and they are denoted as $x_{i,k} = 0$ and $x_{i,k} = 1$, respectively. Usually, node i is in *normal state* for all sensors, monitors sensing targets, and broadcasts sensor data at a regular interval of T seconds. When node i moves to *frequent state* on sensor k, it begins to operate at a new interval of $T_k = T/2^{c_k}$ seconds. To relay sensor data from node i to a BS, nodes on the path from node i to the BS also must change their operation interval to T_k. Here, $c_k \in \{0, 1, 2, \cdots, \lfloor \log_2(T/2\tau_{max}) \rfloor\}$ is an integer value predetermined for sensing target k.

A. Adaptive Sensing using Response Threshold Model

To decide the sensing state for sensing target k at node i in a fully distributed and self-organizing manner, we adopt the response threshold model of division of labor and task allocation in social insects [7]. There are only little research work on the application of the response threshold model to WSN control [8, 9]. For example, [8] considers task allocation for mobile sensor network coverage, and [9] proposes an architecture for division of labor in sensor/actuator networks.

Now, we consider demand $s_{i,k}$ of sensing target k at node i. Following the response threshold model, the demand $s_{i,k}$ changes as

$$\frac{ds_{i,k}}{dt} = \delta_{i,k}(t) - \alpha \frac{m_{i,k}(t)}{n_{i,k}(t)}, \quad (1)$$

where $\delta_{i,k}(t)$ corresponds to the rate of increase and α determines the work efficiency. $m_{i,k}(t)$ and $n_{i,k}(t)$ represent the number of neighbor nodes performing frequent sensing on sensor k at time t and the total number of neighbor nodes of sensor k, respectively. In the equation, when the increase of demand is equivalent to the work output, the demand does not change. If the number of workers is insufficient, the demand increases.

The definition of $\delta_{i,k}$ depends on application requirements. For example, in a steel plant, temperature changes slowly in the order of hours and once it increases, it stays high for a long period of time. Therefore, nodes are required to monitor temperature more frequently when temperature is changing, while the sensing frequency can be decreased under stable condition independently of the absolute value of temperature. It means that the rate of temperature change is appropriate as $\delta_{i,k}$ for temperature monitoring. On the other hand, CO gas suddenly appears and moves fast. The existence of CO gas is harmful to workers near the shaft. Therefore, the absolute value is more important for CO gas sensors and $\delta_{i,k}$ should be defined based on the value itself.

The probabilities that node i begins to work, i.e. monitor the sensing target more frequently and that node i stops frequent sensing are given by the following equations, respectively.

$$P(x_{i,k} = 0 \rightarrow x_{i,k} = 1) = \frac{s_{i,k}^2(t)}{s_{i,k}^2(t) + \theta_{i,k}^2(t)} \quad (2)$$

$$P(x_{i,k} = 1 \rightarrow x_{i,k} = 0) = p_{i,k}(t), \quad (3)$$

where $p_{i,k}(t) \in [0, 1]$ is a parameter. For example, by using an inverse of changing rate of temperature as $p_{i,k}(t)$, we can expect that nodes immediately resume the normal sleep schedule once temperature becomes stable. The readiness of node to get engaged in a frequent sensing task depends on the threshold $\theta_{i,k}$. Threshold is adjusted depending on whether a node performs frequent sensing or not.

$$\frac{d\theta_{i,k}}{dt} = \begin{cases} -\xi, & \text{if } i \text{ performs frequent sensing} \\ \varphi, & \text{otherwise,} \end{cases} \quad (4)$$

where ξ and φ are parameters. The threshold adjustment makes specialists having a small threshold and keep sensing the target frequently. In addition, the threshold adjustment makes the response threshold model insensitive to parameter settings, such as the range of demand $s_{i,k}$ and α.

B. Data Gathering with Adaptive Intervals

Now, we explain how to accomplish adaptive data gathering on traveling wave-based communication. To autonomously generate and maintain traveling wave message propagation, we adopt a pulse-coupled oscillator model, which explains biological synchronization observed in a group of flashing fireflies [6]. A firefly is considered to have an internal timer. If alone, a firefly flashes when a timer expires. In a group, a firefly shifts its timer by a small amount, when it observes a flash of other fireflies while it is not flashing. By mutual interaction through phase shifting, a group of fireflies eventually either reaches global synchronization where timers of all fireflies are synchronized or a traveling wave where fireflies flash alternately keeping a fixed phase difference.

1) Control Parameters: In our scheme, node i maintains the phase $\phi_i \in [0, T]$ ($d\phi_i/dt = 1$), level value l_i, PRC (phase-response curve) function $\Delta_i(\phi_i)$, offset τ_i ($0 < \tau_{min} \leq \tau_i \leq \tau_{max} < 0.5T$), demand vector $\boldsymbol{S_i} = \{s_{i,k} | 1 \leq k \leq k_{max}\}$, threshold vector $\boldsymbol{\Theta_i} = \{\theta_{i,k} | 1 \leq k \leq k_{max}\}$, sensing state vector $\boldsymbol{X_i} = \{x_{i,k} | 1 \leq k \leq k_{max}\}$, relay flag vector $\boldsymbol{F_i} = \{f_{i,k} | 1 \leq k \leq k_{max}\}$, sensing state table $\boldsymbol{Y_i} = \{\forall j \boldsymbol{X_j}\}$ containing sensing state vectors of all neighbor nodes, and sensor data $\boldsymbol{D_i} = \{\boldsymbol{D_{i,k}} | 1 \leq k \leq k_{max}\}$. Entries of vectors are updated based on control information embedded in a received message from a neighbor node, which is emitted at the interval of sensing. It implies that there is no additional transmission of control messages.

A level value l_i indicates the number of hops from a BS. Initially, the level value is set at infinity. A BS uses a level value of zero. The offset τ_i defines the interval of message emission between a node of level $l-1$ and that of level l. Offset τ_i is chosen randomly to avoid synchronous message emissions among nodes of the same level (see nodes B and C in Fig. 2). The maximum offset τ_{max} is determined taking into account the density of nodes. The PRC function determines the amount of phase shift on receiving a broadcast message. To generate concentric traveling waves of message propagation regardless of the initial phase condition, we use the following PRC function for all nodes [2].

$$\Delta_i(\phi_i) = a \sin \frac{\pi}{\tau_i} \phi_i + b(\tau_i - \phi_i), \tag{5}$$

where a and b are parameters which determine the speed of convergence. A relay flag $f_{i,k}$ is used to notify downstream nodes of the existence of upstream nodes in *frequent state*. When sensor node i relays data to downstream nodes from upstream nodes in *frequent state*, the relay flag $f_{i,k} = 1$, otherwise, $f_{i,k} = 0$. Sensor data $\boldsymbol{D_{i,k}}$ contains all sensor data, together with originator's id, generated or received during the wake-up period for sensing target k.

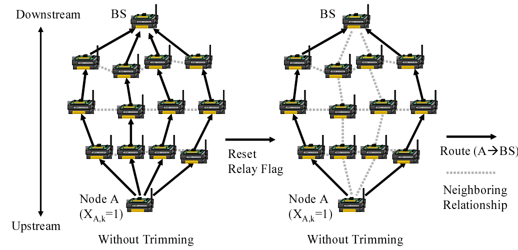


Fig. 3. An example of message reduction

2) Node Behavior: Node i behaves in accordance with its phase ϕ_i and relay flag vector $\boldsymbol{F_i}$. Node i wakes up when $\phi_i \mod \min_k T_k$ becomes $\min_k T_k - \tau_{max}$, where $\min_k T_k = T/\max_k(f_{i,k}2^{c_k}, 1)$. In Fig. 2, node A with sensor $k = 1$ additionally wakes up three times between the regular sensing of interval of T, for having $\min_k T_k = T/4$ with $f_{A,1} = 1$, for example. Upstream nodes setting the same relay flag (node B in Fig. 2) are scheduled to broadcast a message during $\min_k T_k - \tau_{max}$ and $\min_k T_k$ from the node i's viewpoint. After waking up, node i initializes its relay flag vector $\boldsymbol{F_i}$ to 0, clears sensor data $\boldsymbol{D_i}$, and waits for message reception.

When node i receives a message from upstream node j whose level value must be $l_j = l_i + 1$, node i deposits the received sensor data $\boldsymbol{D_j}$. The broadcast message also contains relay flag vector $\boldsymbol{F_j}$, sensing state vector $\boldsymbol{X_j}$. If any relay flag $f_{j,k} \in \boldsymbol{F_j}$ is set at 1 in the received message, node i sets the corresponding relay flag $f_{i,k} = 1$. In addition, node i replaces or adds the entry of node j in its sensing state table $\boldsymbol{Y_i}$ by the received sensing state vector $\boldsymbol{X_j}$.

When node i receives a message from node j with $l_j = l_i$, it checks whether the received sensor data $\boldsymbol{D_j}$ covers its own sensor data $\boldsymbol{D_i}$. Those sensor data contained in the broadcast message of node j are removed from $\boldsymbol{D_i}$. If $\boldsymbol{D_{i,k}}$ becomes empty, node i sets the relay flag $f_{i,k} = 0$. As a result, the amount of sensor data in a message can be reduced and the number of sensor nodes involved in frequent relaying could be further reduced as shown in Fig. 3.

Then, when $\phi_i \mod \min_k T_k$ reaches $\min_k T_k - \epsilon$, node i determines whether it moves to *frequent state* for sensing target $\forall k \in \{k | \phi_i \mod T_k = T_k - \epsilon\}$ or not. The parameter ϵ ($0 < \epsilon < \tau_{min}$) corresponds to the sensing delay. First, node i calculates $n_{i,k}$ and $m_{i,k}$ from sensing state table $\boldsymbol{Y_i}$. Next, node i derives a demand vector $\boldsymbol{S}_i$ using Eq. (1). Then, node i determines its sensing state vector $\boldsymbol{X_i}$ using Eq. (2) and Eq. (3). If any sensing state $x_{i,k}$ is 1, node i monitors sensing target k, deposits sensor data, and sets its relay flag $f_{i,k} = 1$. If ϕ_i is $T - \epsilon$, i.e. timing for regular sensing, node i monitors all sensing targets and deposits the sensor data. After that, node i adjusts its threshold vector $\boldsymbol{\Theta}_i$ using Eq. (4).

When $\phi_i \mod \min_k T_k$ becomes zero in $0 < \phi_i < T$, or ϕ_i reaches T, node i broadcasts a message, which is received by any awake node in the range of radio communication. A message emitted by node i contains node id i, level value l_i, sensing state vector $\boldsymbol{X_i}$, relay flag vector $\boldsymbol{F_i}$, synchronization

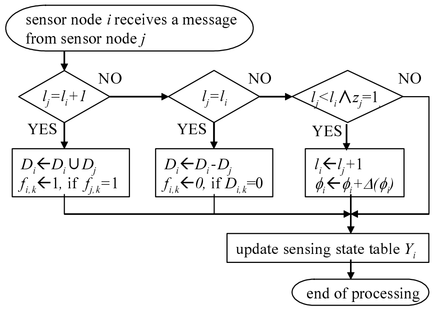


Fig. 4. Node behavior on message reception

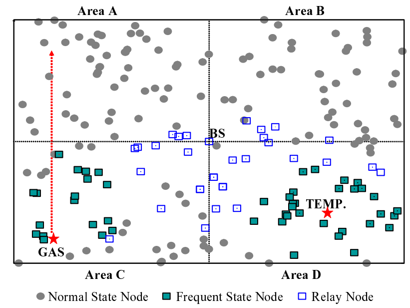


Fig. 5. Node distribution (snapshot at 1200 seconds)

flag z_i, and sensor data $\boldsymbol{D_i}$. z_i is set to 1 only if the phase ϕ_i is T on broadcasting the message. After broadcasting, the phase reaching T goes back to zero.

After the broadcasting, the node stays awake for τ_{max}. When node i receives a message having synchronization flag $z_j = 1$ from node j with $l_j < l_i$, it sets its level value $l_i = l_j + 1$ and shifts its phase by an amount $\Delta_i(\phi_i)$ in Eq. (5). The phase shift is done only once during the τ_{max} awake period. When node i receives a message from node j with $l_j = l_i$, it only updates its sensing state table $\boldsymbol{Y_i}$. After broadcasting, node i waits for τ_{max} before going to sleep. Therefore, a node is awake for the duration of $2\tau_{max}$ in one data gathering interval $\min_k T_k$, i.e. the duty cycle becomes $2\tau_{max}/\min_k T_k$. The algorithm on message reception is illustrated in Fig. 4

C. Overhead of the Mechanism

Now, let us consider protocol overhead and complexity. Each sensor node operates on the phase of timer and wakes up at the frequency of sensing. A message containing sensor data and control information is broadcast once per awake period.

Information that node i has to maintain are, level value l_i, demand vector $\boldsymbol{S_i}$, threshold vector $\boldsymbol{\Theta_i}$, sensing state vector $\boldsymbol{X_i}$, relay flag vector $\boldsymbol{F_i}$, and sensing state table $\boldsymbol{Y_i}$. Then, we can formulate the total amount as $|l_i| + |\boldsymbol{S_i}| + |\boldsymbol{\Theta_i}| + |\boldsymbol{X_i}| + |\boldsymbol{F_i}| + |\boldsymbol{Y_i}| + |\boldsymbol{D_i}|$. When we assume that a level value needs 4 bits (up to 15 hops), the demand and threshold values are both expressed by 32 bits, the relay flag and sensing state need 1 bit each. Thus, for a number of n neighbors, the total amount becomes $4 + (66 + n)k_{max}$ bits. In our scenario of a steel plant, $k_{max} = 2$, i.e. temperature and CO gas concentration, and the number of neighbors is a few dozen. Therefore, the memory size required for control information is less than 200 bits. Most off-the-shelf sensor nodes can afford this amount, e.g. MICAz MOTE [10] has 32 kbits of available memory.

Control information in a message includes node id i, level value l_i, sensing state vector $\boldsymbol{X_i}$, relay flag vector $\boldsymbol{F_i}$, and synchronization flag z_i. Then, we can formulate the total amount as $|i| + |l_i| + |\boldsymbol{X_i}| + |\boldsymbol{F_i}| + |z_i|$. When we assume a node id requires 10 bits (1023 nodes) and a synchronization flag requires 1 bit, the total amount becomes $15 + 2k_{max}$ bits. If $k_{max} = 2$, only 19 bits are needed for control information in a message.

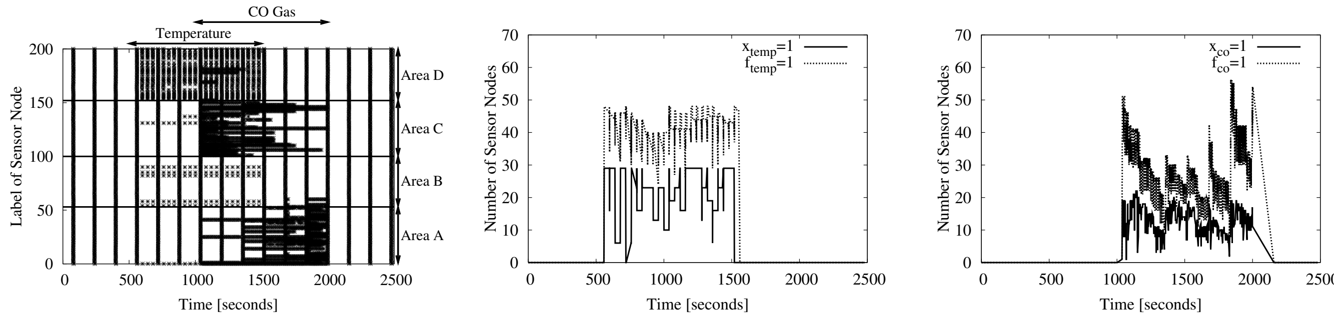
III. Simulation Experiments

In this section, we show results of preliminary simulation experiments. We consider a WSN of 200 nodes randomly distributed in a 100 m $\times$ 100 m region as shown in Fig. 5. A BS is located in the center of the region. Both sensing and communication ranges are fixed at 20 m. We consider two sensing targets, i.e. temperature and CO gas concentration. The normal data gathering interval is $T = 160$ seconds. For unusual conditions, we set $c_{temp} = 2$ and $c_{gas} = 4$, and therefore, $T_{temp} = 40$ seconds and $T_{gas} = 10$ seconds, respectively.

We use $\delta_{i,temp} = \beta|dv_{i,temp}/dt|$ and $\delta_{i,gas} = v_{i,gas}$ in Eq. (1), and $p_{i,temp} = 1 - \beta|dv_{i,temp}/dt|$ and $p_{i,gas} = 1 - v_{i,gas}$ in Eq. (3) for all nodes. $v_{i,temp} \in [0, 1]$ and $v_{i,gas} \in [0, 1]$ are temperature and CO gas concentration normalized by its maximum value, respectively. We use $\beta = 1000$. A node consumes 52.2 mW for transmission, 59.1 mW for reception, 60 μW in idle mode, and 3 μW in sleep mode [10]. τ_{max} and τ_{min} is set at 1 second and 0.5 second, respectively. We assume that control information in a message amounts to 24 bits and one sensor data amounts to 16 bits. We use $a = 0.01$ and $b = 0.5$ for PRC in Eq. (5) and the parameters of response threshold model are set to $\alpha = 1$, $\xi = 0.01$, and $\varphi = 0.001$.

Until 500 seconds, nothing happens where temperature v_{temp} is 0.1 and no CO gas is observed. From 500 seconds, temperature begins to linearly increase at a randomly chosen location in the area D. At 1500 seconds, temperature at the location reaches 0.9 and stays high until the end of simulation. During the temperature increase, CO gas of concentration $v_{gas} = 0.8$ leaks out from a randomly chosen location in the area C at 1000 seconds. The gas moves to the area A at the speed of 0.08 m/s. At 2000 seconds, the gas disappears.

Figure 6(a) shows the time instants when a node, whose id is shown on the left y-axis, broadcasts a message. Nodes are numbered based on the location as shown on the right y-axis. Figures 6(b) and 6(c) show the number of nodes in *frequent state* for each sensing target. At first, all nodes are in the *normal state*. At the end of the awake period at 500 seconds, some nodes in the area D decide to monitor and report temperature more often to respond to the increased sensing demand caused by the temperature change. From the next timing, at most 160 seconds later, they begin frequent



(a) Timing of message broadcasting

(b) Nodes in *frequent state* for temperature

(c) Nodes in *frequent state* for CO gas

Fig. 6. Simulation experiments to evaluate basic behavior of proposed scheme

sensing of temperature as shown in Fig. 6(b). Since they set the relay flag $f_{i,temp}$ in broadcast messages, nodes on the path to the BS also set their relay flag and begin to operate at the shorter intervals of $T_{temp} = 40$ seconds. At 1500 seconds, nodes find that the temperature becomes stable again, and they return to *normal state*.

Figure 5 depicts a snapshot of the WSN at 1200 seconds. Each filled circle, filled square, and open square correspond to a node in *normal state*, a node in *frequent state*, and a relay node, respectively. Some nodes between the BS and nodes in the *frequent state* stay in the normal state and keep operating at the normal interval. It contributes to a reduction of energy consumption and probability of collisions among nodes.

In a similar manner, at 1000 seconds, some nodes in the area C begin to operate at the shorter intervals of $T_{gas} = 10$ seconds for CO gas leakage. As CO gas moves to the area A, nodes in the area C eventually go back to *normal state* while nodes in the area A move to the *frequent state* as shown in Fig. 6(a). Nodes in the frequent state evaluate Eqs. (2) and (3) often, whereas nodes moving from the *normal state* to the *frequent state* begin frequent sensing at the next wake-up. Therefore, we observe the gradual decrease on short time scale and the sudden increase over a long time scale in Fig. 6(c).

During the simulation experiments, the total energy consumption of our proposal is 57.6 mJ per node and duty cycle is 0.03 per node. On the other hand, when we adopt 10 seconds as the regular data gathering interval to prepare for CO gas leakage, the per-node energy consumption is 632 mJ and duty cycle is 0.2. Therefore, we can conclude that our scheme accomplishes autonomous and energy-efficient data gathering satisfying dynamically changing sensing requirements.

IV. Conclusion

In this paper, we proposed an autonomous data gathering scheme adaptive to sensing requirements in WSNs composed of nodes with multiple sensing capabilities. To accomplish self-organizing control, we adopted the response threshold model for adaptive sensing task engagement and the pulse-coupled oscillator model for energy-efficient transmission and sleep scheduling. Through preliminary simulation experiments, we confirmed that autonomous and energy-efficient data gathering can be accomplished satisfying dynamically changing sensing requirements.

As future research work, we need in-depth evaluation of the sensing adaptability of our proposal from application viewpoints. We also need to evaluate our mechanism in comparison with other data gathering schemes. Furthermore, we plan to implement our scheme using off-the-shelf nodes to verify the practicability of our scheme.

Acknowledgments

The authors would like to thank Kenji Leibnitz and the anonymous reviewers for their helpful comments. This work was partly supported by "Global COE (Centers of Excellence) Program" and Grant-in-Aid for Scientific Research on Priority Areas 18049050 of the Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] L. Wang and Y. Xiao, "A survey of energy-efficient scheduling mechanisms in sensor networks," *Mobile Networks and Applications*, vol. 11, no. 5, pp. 723–740, Oct. 2006.

[2] Y. Taniguchi, N. Wakamiya, and M. Murata, "A traveling wave based communication mechanism for wireless sensor networks," *Journal of Networks*, vol. 2, no. 5, pp. 24–32, Sep. 2007.

[3] D. Messina, M. Ortolani, and G. L. Re, "Reliable data gathering in tree-based IEEE 802.15.4 wireless sensor networks," in *Proc. of MobiQuitous 2007*, Aug. 2007, pp. 1–7.

[4] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adapive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks," *Wireless Communications & Mobile Computing*, vol. 7, no. 7, pp. 863–875, Sep. 2007.

[5] C. Alippi, G. Anastasi, C. Galperti, F. Mancini, and M. Roveri, "Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications," in *Proc. of MASS 2007*, Oct. 2007, pp. 1–6.

[6] P. Goel and B. Ermentrout, "Synchrony, stability, and firing patterns in pulse-coupled oscillators," *Physica D*, pp. 191–216, Mar. 2002.

[7] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J.-L. Deneubourg, "Adaptive task allocation inspired by a model of division of labor in social insects," in *Proc. of Biocomputing and Emergent Computation*, 1997, pp. 36–45.

[8] K. H. Low, W. K. Leow, and J. Marcelo H. Ang, "Task allocation via self-organizing swarm coalitions in distributed mobile sensor network," in *Proc. of AAAI-04*, Jul. 2004, pp. 28–33.

[9] T. H. Labella and F. Dressler, "A bio-inspired architecture for division of labour in SANETs," in *Proc. of BIONETICS 2006*, Dec. 2006.

[10] "MICAz," http://www.xbow.com/.