Self-organizing Transmission Scheduling Considering Collision Avoidance for Data Gathering in Wireless Sensor Networks

Yoshiaki Taniguchi, Go Hasegawa and Hirotaka Nakano
Cybermedia Center, Osaka University,
1–32 Machikaneyama, Toyonaka, Osaka 560–0043, Japan
Email: {y-tanigu, hasegawa, nakano}@cmc.osaka-u.ac.jp

***Abstract* — In this paper, we propose self-organizing transmission and sleep scheduling mechanisms for periodically gathering sensor data at a base station in wireless sensor networks. In our proposed mechanisms, we use a conventional self-organizing communication mechanism based on phase-locking in a pulse-coupled oscillator model to propagate sensor data from the edge of the network to the base station in order of hop counts from the base station. In addition, we propose two mechanisms to avoid collisions among sensor nodes with the same hop count, a randomization-based mechanism and a desynchronization-based mechanism based on anti-phase synchronization in the pulse-coupled oscillator model. Through simulation experiments, we show that our proposed mechanism significantly improves the data gathering ratio and energy-efficiency in comparison with the conventional mechanism.**

***Index Terms* — wireless sensor networks, data gathering, phase-locking, anti-phase synchronization, pulse-coupled oscillator model**

I. Introduction

Wireless sensor networks (WSNs) have attracted attention from many researchers and developers, particularly for monitoring applications [2], [3]. In typical monitoring applications, the WSN consists of a base station (BS) and many battery-powered sensor nodes, each with a general-purpose processor with limited computational capability, a small amount of memory, and a wireless communication device. Sensor node data are gathered at the BS through wireless multi-hop networks at regular intervals. Highly energy-efficient control mechanisms are required to prolong the lifetime of battery-powered WSNs. In addition, WSN control mechanisms should be distributed and self-organizing, because many sensor nodes are randomly deployed and the network topology dynamically changes due to environmental noise, sensor node failures, and so on.

There have been many proposals for WSN data gathering mechanisms [4]–[10]. In traditional data gathering mechanisms [4]–[6], a tree topology rooted at the BS is maintained due to many-to-one features of the monitoring applications. Data packet transmission and sleep timings are dispersed to avoid radio interference and collisions. In these mechanisms, sensor nodes need to send control packets to maintain the network topology, to synchronize with neighbor sensor nodes, or to avoid interference and collisions among neighbor nodes. In the case of a network topology change, sensor nodes need to send additional control packets to restore the topology, which require time and energy.

Some papers [11]–[21] have proposed mechanisms applying *pulse-coupled oscillator (PCO) models*, which are models of biological synchronization behavior such as observed in flashing fireflies [22], [23], as a mechanism for self-organizing inter-node communication. PCO models explain several kinds of synchronous behavior among oscillators, such as *in-phase synchronization*, where oscillators completely synchronize, *phase-locking*, where oscillators synchronize with a constant offset, and *anti-phase synchronization*, where oscillators synchronize with an equal interval, as shown in Fig. 1. By applying PCO models to WSN controls, simple, distributed and self-organizing mechanisms under random and dynamic environments can be achieved.

For example, in Ref. [15], the authors proposed a time division multiple access (TDMA)-based communication mechanism, called DESYNC, implementing anti-phase synchronization. In DESYNC, sensor nodes self-organize the assignment of equal-size time slots, regardless of their initial condition and the number of sensor nodes. However, DESYNC does not consider the *hidden node problem*, and communication among sensor nodes in multi-hop WSNs can fail due to collisions when two-hop neighbor sensor nodes simultaneously transmit packets to the same sensor node.

In Ref. [19], a self-organizing communication mechanism based on phase-locking, called WAVE, was proposed. WAVE can organize two kinds of communication pattern in multi-hop WSNs, namely, diffusion and gathering. In WAVE, a sensor node self-configures its message transmission and sleep timings. For gathering, sensor nodes configure their transmission timing so that sensor nodes with the same hop count from the BS simultaneously wake up, receive messages from downstream sensor

Manuscript received May 28, 2013; revised June 26, 2013.
This work was partly supported by JSPS KAKENHI Grant No. 23700079 and 25330103. Preliminary version of this paper is presented at ICDIPC 2012, Lithuania, July 2012 [1].
Corresponding author email: y-tanigu@cmc.osaka-u.ac.jp.
doi:10.12720/jcm.8.6.389-397

©2013 Engineering and Technology Publishing

nodes, and forward the message to upstream sensor nodes. Sensor data from the edge of the WSN are therefore gathered at the BS. However, Ref. [19] mainly focuses on phase-locking applications to organize two kinds of WSN communication patterns, and detailed mechanisms for monitoring applications have not been well studied. For example, WAVE does not consider collisions due to simultaneous transmissions among sensor nodes with the same hop count, which lowers the data gathering ratio.

In this paper, we propose self-organizing transmission and sleep scheduling mechanisms for periodically gathering sensor data at a BS in WSNs. In our proposed mechanisms, sensor nodes self-organize message transmission and sleep timings. To propagate sensor data from the edge of the WSN to the BS in order of hop counts from the BS, we use a conventional self-organizing communication mechanism based on phase-locking in a PCO model, i.e. WAVE. We furthermore propose two message timing mechanisms to avoid collisions among sensor nodes with the same hop count, a simple randomization-based mechanism that does not explicitly consider the hidden node problem, and a desynchronization-based mechanism based on anti-phase synchronization in the PCO model. To the best of our knowledge, this is the first research to apply both phase-locking and anti-phase synchronization of the PCO model for data gathering in WSNs. We evaluate our proposed mechanisms in terms of the data gathering ratio and consumed energy ratio through simulation experiments.

The rest of this paper is organized as follows. In Section II, we briefly introduce related work and conventional mechanisms. In Section III, we propose self-organizing transmission and sleep scheduling mechanisms for gathering sensor data in WSNs by extending the conventional mechanism WAVE. In Section IV, we evaluate our proposed mechanisms through simulation experiments. Finally, we conclude this paper with an outlook for future work in Section V.

II. Conventional Mechanisms

In this section, we first explain the PCO model. Then, we briefly introduce a communication mechanism using the PCO model for WSNs.

A. PCO Model

We first explain a PCO model [23] to explain synchronous behavior of biological oscillators, as observed in flashing fireflies. Consider a set of N oscillators o_i $(1 \leq i \leq N)$. Oscillator o_i has phase $\phi_i \in [0, T]$. As time passes, phase ϕ_i shifts toward T and, after reaching it, oscillator o_i *fires* and the phase ϕ_i jumps back to zero. Oscillator o_j coupled with the firing oscillator o_i is stimulated and shifts its phase by an amount $\Delta(\phi_j)$ as follows.

$$\phi_j \leftarrow \phi_j + \Delta(\phi_j), \tag{1}$$

where $\Delta(\phi_j)$ is called a phase-response curve (PRC). Depending on the initial phase value, properties of the PRC, and coupling relationships among oscillators, a set of oscillators can show several kinds of synchronous behavior, such as in-phase synchronization, where all oscillators fire simultaneously, phase-locking, where oscillators fire with a constant offset, or anti-phase synchronization, where oscillators fire at an equal interval. Figure 1 shows an example of each kind of synchronization.

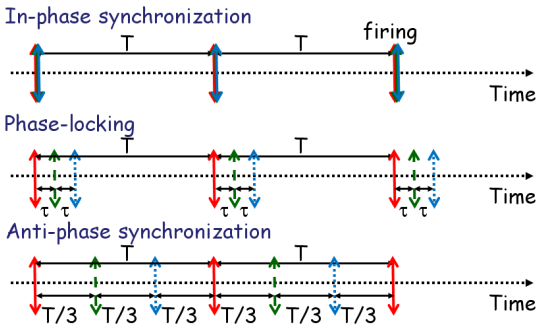


Figure. 1. Examples of in-phase synchronization, phase-locking, and anti- phase synchronization among three oscillators.

By regarding sensor nodes as oscillators and message broadcasts as firing, the PCO model can be adapted to sensor node transmission scheduling in WSNs. By applying PCO models to WSN controls, simple, distributed and self-organizing mechanisms under random and dynamic environments can be achieved.

B. WAVE

WAVE [19] is a communication mechanism using PCO phase-locking. WAVE can organize two types of communication patterns, data diffusion and data gathering, depending on application requirements. The following briefly explains the behavior of WAVE in data gathering.

Sensor node n_i $(1 \leq i \leq N)$ maintains a timer with phase ϕ_i, level l_i, and offset τ. The level indicates the number of hops from the BS, and is initialized with a large value. The offset defines the interval of message transmission between a sensor node of level l and one of $l-1$. As time passes, phase ϕ_i shifts toward T and, upon reaching it, sensor node n_i broadcasts a message and the phase jumps back to zero. A message contains sensor data and control information. When sensor node n_j receives a message from a sensor node n_i with level l_i smaller than its level l_j, it sets its level to $l_j = l_i + 1$. This stimulates the sensor node and changes its phase as follows.

$$\phi_j \leftarrow \phi_j + a \sin \frac{\pi}{\tau}\phi_j + b(\tau - \phi_j), \tag{2}$$

where a and b are parameters that determine the speed of convergence. By using the second term of Eq. (2) as the PRC function, phase-locking is accomplished through mutual interaction among sensor nodes regardless of their initial phase condition [19].

Figure 2 shows an example of the basic behavior of WAVE in the case of data gathering. The ring in Fig. 2 indicates the axis of phase, and small circle indicates

©2013 Engineering and Technology Publishing

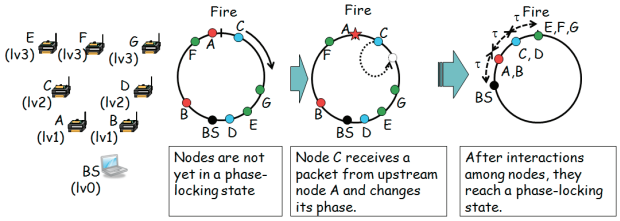


Figure. 2. Basic behavior of WAVE in the case of data gathering.

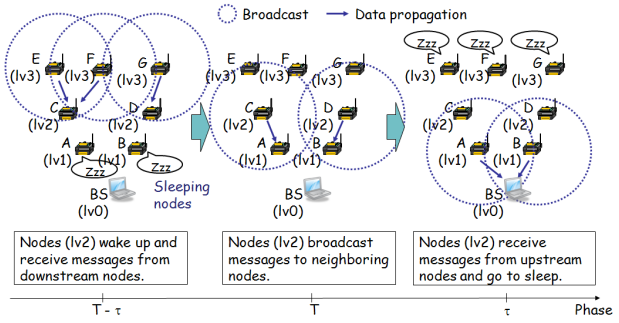


Figure. 3. Transmission and sleep timing of WAVE in the steady state.

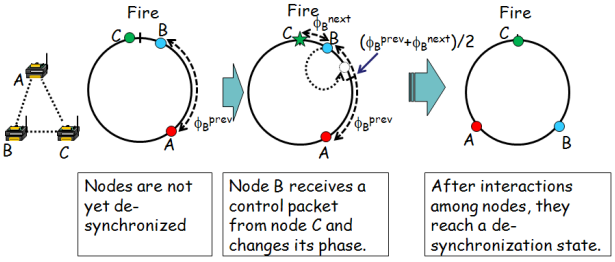


Figure. 4. Basic behavior of DESYNC in a single-hop topology.

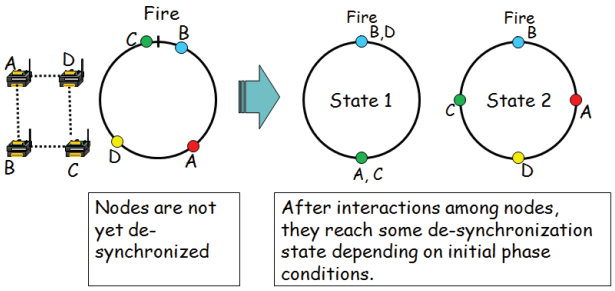


Figure. 5. Basic behavior of DESYNC in a multi-hop topology.

phase of sensor node. The circle moves clockwise on the ring with period T according to the phase of sensor node. When the phase of sensor node is T, the circle reaches the top of the ring. As shown in Fig. 2, sensor nodes reach a phase-locking state after interactions among sensor nodes regardless of their initial phase condition. Figure 3 shows transmission and sleep timing of WAVE from the viewpoint of sensor nodes two hops away from the BS in steady state. In steady state, a sensor node wakes up to receive messages with sensor data from downstream sensor nodes when its phase ϕ_i is $T-\tau$. When the phase reaches T, the sensor node broadcasts a message with sensor data. After that, it remains awake for duration τ to receive messages from upstream sensor nodes, then returns to sleep. Sensor nodes are therefore awake for duration 2τ in one communication cycle T.

In WAVE, simultaneous message broadcasts among sensor nodes with the same hop count leads to radio signal collisions and a decreased data gathering ratio. For example, when sensor nodes E and F in Fig. 3 simultaneously broadcast messages, upstream sensor node C cannot receive either message due to collision. Therefore, an additional mechanism is required to apply WAVE to data gathering in WSNs. We propose two mechanisms to disperse the message transmission timing among sensor nodes with the same hop count in Section III.

C. DESYNC

DESYNC [15] is a TDMA-based communication mechanism using PCO anti-phase synchronization. In DESYNC, sensor node n_i $(1 \leq i \leq N)$ maintains a timer with phase ϕ_i. Sensor node n_i broadcasts a control packet when its phase ϕ_i reaches T, then the phase resets to zero. When sensor node n_j receives a control packet from a neighbor sensor node, it records the phase difference as

$$\phi_j^{prev} = T - \phi_j. \tag{3}$$

If the control packet reception is the first from a broadcast of sensor node n_j, it records the phase as

$$\phi_j^{next} = \phi_j, \tag{4}$$

and changes its phase as follows.

$$\phi_j \leftarrow (1-\alpha)\phi_j + \alpha\phi_j^{mid}, \tag{5}$$

where

$$\phi_j^{mid} = \frac{\phi_j^{prev} + \phi_j^{next}}{2}, \tag{6}$$

and α is a parameter that determine the speed of convergence. When all sensor nodes are able to communicate with each other, anti-phase synchronization is accomplished through mutual interaction among sensor nodes, regardless of their initial phase condition.

Figure 4 shows an example of anti-phase synchronization convergence in a single-hop WSN composed of three sensor nodes by using DESYNC. As shown in Fig. 4, sensor nodes reach a desynchronization state after interactions among sensor nodes. Sensor node n_i uses the phase duration from $T - \phi_i^{prev}/2$ to $\phi_i^{next}/2$ as its time slot.

DESYNC does not consider the hidden node problem, and depending on the initial phase conditions, communication among sensor nodes can fail in multi-hop WSNs due to collision among two-hop neighbor sensor nodes [16]. For example, if four sensor nodes are connected as shown in Fig. 5, the steady state becomes either state 1 or state 2 depending on the initial phase conditions. In state 1, a sensor node and its hidden node (sensor nodes A and C or sensor nodes B and D) are synchronized and

©2013 Engineering and Technology Publishing

fire at same timings, whereas all sensor nodes fire with an equal interval in state 2. In state 1, sensor nodes cannot communicate with each other. For example, sensor node C cannot receive packets from either sensor node B or sensor node D due to collision. In this paper, we propose a mechanism to consider the hidden node problem in

III. Proposed Mechanisms

In this section, we propose two WAVE-based self-organizing transmission and sleep scheduling mechanisms for data gathering in WSNs. In our proposed mechanisms, sensor nodes periodically broadcast messages that include sensor data and control information. Here, the sensor node transmission range is assumed as a constant radius circle among sensor nodes. Sensor node transmission and sleep timings are self-configured through mutual interaction among sensor nodes without topology maintenance. We should note here that our mechanisms do not need additional control packets to maintain topology or to synchronize among sensor nodes, although control packets are often required in traditional data gathering mechanisms [4]–[6].

In our proposed mechanisms, phase-locking is used to propagate sensor data from the edge of the WSN to the BS in order of hop count, as in WAVE. In addition, to avoid collisions among sensor nodes with the same hop count, we propose a desynchronization-based mechanism and a randomization-based mechanism. The following describes details of our proposed mechanisms.

A. Desynchronization-based Mechanism

1) Overview: We first propose a desynchronization-based mechanism. The mechanism avoids collisions among sensor nodes with the same hop count by using the DESYNC algorithm to adjust sensor node offsets. Figure 6 shows an example of message transmission timing and offset τ_i among sensor nodes in a steady state when the desynchronization-based mechanism is used. In Fig. 6, sensor nodes A, B, and C have the same hop count from the BS. Their message transmission timings are dispersed since their offset τ are dispersed. Here, sensor nodes A and C are two-hop neighbor sensor nodes (hidden nodes), and cannot directly communicate with each other. In the desynchronization-based mechanism, to avoid collisions among two-hop neighbor sensor nodes, information on the message transmission timing of two-hop neighbor sensor nodes is obtained from single-hop upstream neighbor sensor nodes, i.e. sensor node D in Fig. 6. The following describes details of the desynchronization-based mechanism.

2) Control information: Sensor node n_i has a timer t_i, and maintains phase $\phi_i \in [0, T]$, level l_i, offset τ_i $(0 < \tau_i \leq \tau^{max})$, message transmission timing table $\mathcal{E}_i$, and sensor data $\mathcal{D}_i$. T is the data gathering interval. Level l_i indicates the number of hops from a BS, and is initially set to a large value. A BS has a level of zero. Offset τ_i determines the message transmission interval between

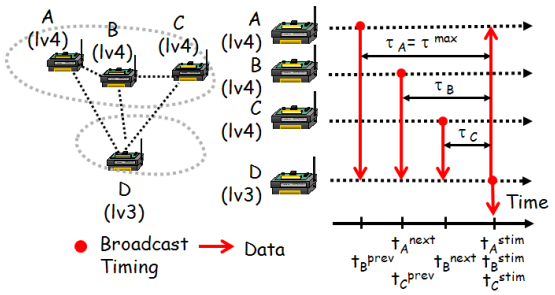


Figure. 6. Example of message transmission timing and offset in the desynchronization-based mechanism.

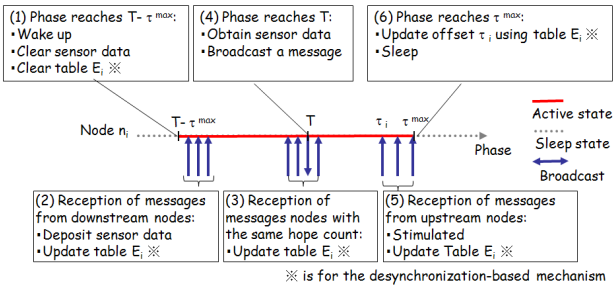


Figure. 7. Sensor node behavior in the proposed mechanisms.

sensor node n_i and a sensor node that stimulates sensor node n_i, and is initially set to τ^{max}. The maximum offset τ^{max} is determined according to sensor node density. Message transmission timing table $\mathcal{E}_i$ is used to maintain information on message transmission timing of sensor node n_j within two hops from sensor node n_i. Each entry of the message transmission timing table $e_{i,j} \in \mathcal{E}_i$ consists of sensor node identifier j, level l_j, and estimated message transmission time $t_{i,j}^{trans}$ of sensor node n_j, i.e. $e_{i,j} = (j, l_j, t_{i,j}^{trans})$. Sensor data $\mathcal{D}_i$ are generated from its sensor and data received from downstream sensor nodes during the wake-up period.

3) Message structure: A message transmitted from sensor node n_i contains its identifier i, level l_i, message transmission timing information $\mathcal{F}_i$, and sensor data $\mathcal{D}_i$. Message transmission timing information $\mathcal{F}_i$ is information on message transmission timing of downstream sensor nodes. The information is generated from message transmission timing table $\mathcal{E}_i$, and is defined as:

$$\mathcal{F}_i = \{f_{i,j} | l_j = l_i + 1\}, \tag{7}$$

where $f_{i,j} = t_i^{trans} - t_{i,j}^{trans}$, and t_i^{trans} is the message transmission time of sensor node n_i.

4) Sensor node behavior: In the following, we describe sensor node behavior after a certain number of cycles have elapsed and sensor nodes have reached the steady state. Figure 7 shows the steady-state sensor node behavior. Sensor node n_i behaves in accordance with its phase ϕ_i and received control information from neighbor sensor nodes as follows:

(1) Wake-up:

When phase ϕ_i becomes $T - \tau^{max}$, sensor node n_i wakes up and clears message transmission timing table $\mathcal{E}_i$

and sensor data $\mathcal{D}_i$. Then, it waits for message reception from neighbor sensor nodes.

(2) Message reception from downstream nodes:

When sensor node n_i receives a message from downstream sensor node n_j with $l_j = l_i+1$ at time t, it adds the received sensor data $\mathcal{D}_j$ to its sensor data $\mathcal{D}_i$. In addition, sensor node n_i adds a new entry $e_{i,j} = (j, l_j, t)$ to its message transmission timing table $\mathcal{E}_i$.

(3) Message reception from same-hop nodes:

When sensor node n_i receives a message from sensor node n_j with $l_j = l_i$ at time t, it adds a new entry $e_{i,j} = (j, l_j, t)$ to its message transmission timing table $\mathcal{E}_i$.

(4) Message transmission:

When phase ϕ_i reaches T, sensor node n_i broadcasts a message, which is received by all awake sensor nodes in the radio transmission range. After reaching T, phase ϕ_i goes back to zero. In addition, sensor node n_i records time t as message transmission time t_i^{trans}. Note that a CSMA/CA mechanism is also used for message transmission, as described in Section III-A5.

(5) Message reception from upstream nodes:

After the phase returns to zero, sensor node n_i stays awake for τ^{max}. When sensor node n_i receives a message from upstream sensor node n_j with $l_j < l_i$ at time t, it sets its level $l_i = l_j + 1$. Then it is stimulated and shifts its phase as follows.

$$\phi_i \leftarrow \phi_i + a \sin \frac{\pi}{\tau_i}\phi_i + b(\tau_i - \phi_i). \tag{8}$$

The phase shift is done only once during the τ^{max} awake period. Sensor node n_i records the time t as stimulated time t_i^{stim}.

Furthermore, if the received message satisfies $l_j = l_i - 1$, sensor node n_i updates its message transmission timing table $\mathcal{E}_i$ as follows. First, it uses received message transmission timing information $\mathcal{F}_j$ to calculate the estimated message transmission time $t_{j,k}^{trans} = t - f_{j,k}$ of sensor node n_k, which is a single- or two-hop neighbor sensor node from sensor node n_i. If there is no entry for sensor node n_k in its message transmission timing table $\mathcal{E}_i$, sensor node n_i adds a new entry $e_{i,k} = (k, l_i, t_{j,k}^{trans})$ to the table. If an entry for sensor node n_k exists in its message transmission timing table $\mathcal{E}_i$ and $t_{i,k}^{trans} > t_{j,k}^{trans}$ is satisfied, sensor node n_i overwrites the entry as $e_{i,k} = (k, l_i, t_{j,k}^{trans})$. Figure 8 shows the flowchart for updating message transmission table $\mathcal{E}_i$ of sensor node n_i according to message reception from upstream sensor node n_j.

(6) Updating offset and sleep:

When phase ϕ_i reaches τ^{max} and sensor node n_i is stimulated, it adjusts its offset τ_i as follows; otherwise, it waits for stimulation by receiving a message from an upstream sensor node. First, sensor node n_i obtains t_i^{prev} and t_i^{next} as follows,

$$t_i^{prev} = \max_{t \in \mathcal{T}_i, t < t_i^{trans}} t, \tag{9}$$

$$t_i^{next} = \min_{t \in \mathcal{T}_i, t > t_i^{trans}} t, \tag{10}$$

where $\mathcal{T}_i$ is a set of estimated message transmission times in $\mathcal{E}_i$. If t_i^{prev} or t_i^{next} are not obtained, they are set to zero. The bottom of Fig. 6 shows examples for t_i^{prev} and t_i^{next}. Next, sensor node n_i adjusts offset τ_i using the following function.

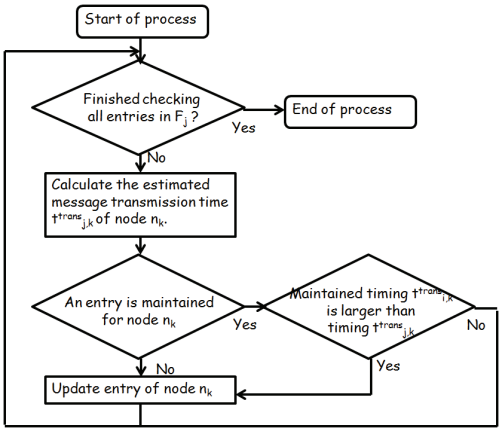


Figure. 8. Flowchart for updating message transmission table of sensor node n_i according to message reception from upstream sensor node n_j.

$$\tau_i \leftarrow (1-\alpha)\tau_i + \alpha\tau_i^{mid}, \tag{11}$$

where

$$\tau_i^{mid} = \begin{cases} \dfrac{\tau_i^{prev} + \tau_i^{next}}{2} & \text{if } t_i^{next} > t_i^{prev} > 0 \\ \dfrac{\tau_i^{prev}}{2} & \text{if } t_i^{next} = 0,\ t_i^{prev} > 0 \\ \tau^{max} & \text{otherwise,} \end{cases}$$

$$\tau_i^{prev} = t_i^{stim} - t_i^{prev},$$

$$\tau_i^{next} = t_i^{stim} - t_i^{next}.$$

After adjusting offset τ_i, sensor node n_i goes to sleep.

5) Message transmission using CSMA/CA: In the desynchronization-based mechanism, although transmission timing is dispersed among two-hop neighbor sensor nodes, message loss cannot be perfectly prevented due to initial phase conditions, unsuspected lower-layer delays, or environmental noise. When there is message loss, the message transmission timing table is not updated and a sensor node transmits a message at the same timing in succeeding cycles, which leads to message loss in those cycles, too. To prevent message loss in succeeding cycles, the desynchronization-based mechanism uses a CSMA/CA mechanism [24]. In CSMA/CA, a sensor node first waits for a random duration, then detects the channel condition. When the channel is idle, the sensor node transmits a message. Otherwise, it waits for a random duration. If channel is still busy after a certain number of detection trials, the sensor node fails to transmit the message.

B. Randomization-based mechanism

In the randomization-based mechanism, offset τ_i is randomly selected between 0 and τ^{max} in step 6 in Section III-A4 at each cycle, instead of using the DESYNC algorithm as in the desynchronization-based mechanism.

©2013 Engineering and Technology Publishing

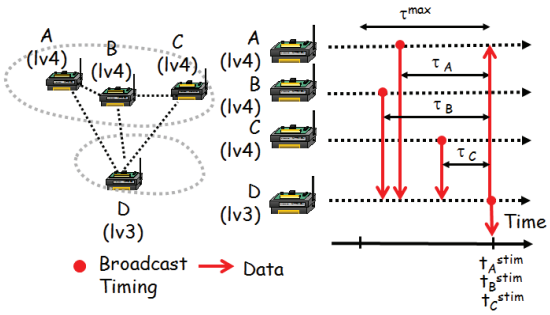


Figure. 9. Example of message transmission timing and offset in the randomization-based mechanism.

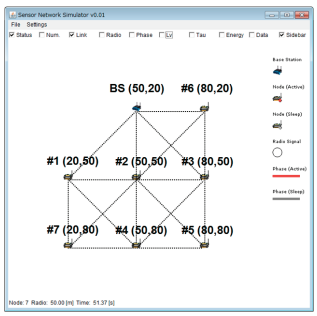


Figure. 10. Sensor node deployment for evaluating basic behavior of our proposed mechanisms.

Therefore, unlike in the desynchronization-based mechanism, in the randomization-based mechanism sensor node n_i does not maintain message transmission table $\mathcal{E}_i$ and does not include message transmission timing information $\mathcal{F}_i$ in messages. While the randomization-based mechanism is simpler and requires less control overhead than the desynchronization-based mechanism, it does not explicitly consider the hidden node problem. Figure 9 shows an example of message transmission timing and offset τ_i among sensor nodes in a steady state when the randomization-based mechanism is used.

IV. Simulation Evaluation

This section evaluates our proposed mechanisms through simulation experiments.

A. Simulation Settings

In the simulation, the size of monitoring region is fixed at 100 m×100 m. Data gathering cycle T is set to 1 s and the maximum offset τ^{max} is set to 0.1 s. We use $a = 1$, $b = 0.5$ for Eq. (8), and $\alpha = 0.5$ for Eq. (11). Sensor node initial phase values are randomly set. The message header size is set to 2 B, the size of a single sensor datum is set to 2 B, and the size of each entry for message transmission timing information $f_{i,j} \in \mathcal{F}_i$ is set to 1 B. The wireless communication bandwidth is set to 250 kbps. The sensor node transmission range is considered as a 50 m radius circle. When a sensor node simultaneously receives messages from two or more sensor nodes, it fails to receive those messages due to collision. For CSMA/CA parameters, back-off slot duration is set to 1 ms and the maximum back-off is set to 4. Energy consumption of the transmit, receive, idle, and sleep states are set to 52.2 mW, 59.1 mW, 60 μW, and 3 μW, respectively [25].

We evaluate our proposed data gathering mechanism in terms of the *data gathering ratio* and the *consumed energy ratio*. The data gathering ratio is the ratio of the amount of sensor data gathered at the BS per cycle divided to the number of sensor nodes. The consumed energy ratio is the ratio of total consumed energy by radio transmission devices on the WSN, divided by the amount of sensor data gathered at the BS per cycle.

B. Evaluation of basic Behavior in the Transient State

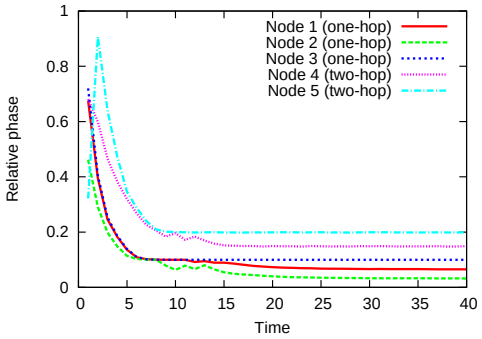
We first evaluate fundamental performance of our proposed mechanisms in the transient state. For evaluation, we use a WSN as shown in Fig. 10. In the WSN, sensor nodes 1, 2, 3, 6 are one-hop away from the BS and sensor nodes 4, 5, 7 are two-hop away from the BS. In the simulation, the WSN is started with the BS and five sensor nodes 1–5 at $t = 0$. Later, sensor nodes 6 and 7 are added at $t = 40$. At $t = 70$, sensor nodes 1 and 4 are removed from the WSN. Sensor node initial phase values are randomly set.

Figure 11(a) shows how the WSN reached the steady state in this experiment when the desynchronization-based mechanism is used. The y-axis is *relative phase* which is the phase of sensor node when the phase of the BS is zero. As shown in Fig. 11(a), the relative phase of sensor nodes are random at first. After interactions among sensor nodes, sensor nodes reach a steady state where the order of relative phase is according to hop counts from the BS and all relative phases of two-hop neighbor sensor nodes are dispersed. Therefore, collisions among sensor nodes with the same hop count can be prevented by using the desynchronization-based mechanism. However, as shown in Fig. 11(a), it takes roughly 20 cycles to reach the steady state in the small WSN composed of five sensor nodes. The reduction of cycles in the transient state is considered as future work.

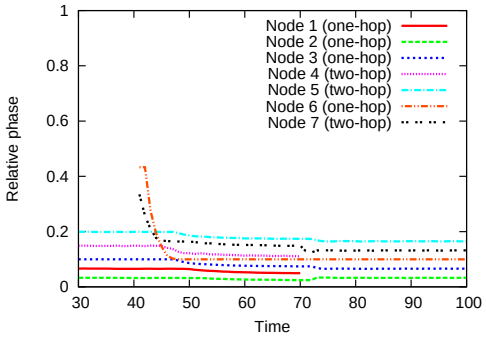
Figure 11(b) shows the relative phase against addition and removal of sensor nodes. At $t = 40$, sensor nodes 6 and 7 with random initial phase are added to the WSN. After some interactions among sensor nodes, sensor node transmission timings are self-configured and they again reach a steady state as shown in Fig. 11(b). After that, sensor nodes 1 and 4 are removed at $t = 70$. As well as addition of sensor nodes, transmission timings of remained sensor nodes are self-configured and they again reach a steady state after some interactions as shown in Fig. 11(b). Here, we should note that these configuration is accomplished in a self-organizing manner without sending control packets.

Figure 12 shows the relative phase when the randomization-based mechanism is used. Since offset τ is randomly set for each sensor node in the randomization-based mechanism, the relative phase fluctuates as shown

©2013 Engineering and Technology Publishing

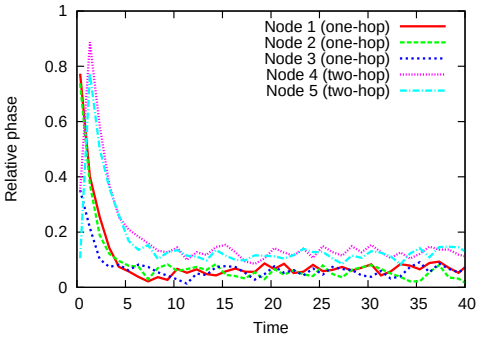


(a) Relative phase of sensor nodes in the transient state. Initial phase values of sensor nodes are ramdomly set.

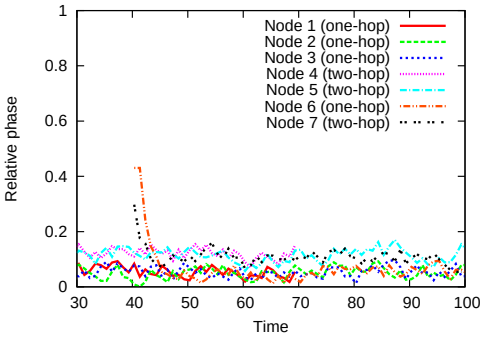


(b) Relative phase of sensor nodes against addition/removal of sensor nodes. Sensor nodes 6 and 7 are added at $t = 40$, and sensor nodes 1 and 4 are removed at $t = 70$.

Figure.11. Evaluation of basic behavior of the desynchronization-based mechanism.



(a) Relative phase of sensor nodes in the transient state. Initial phase values of sensor nodes are ramdomly set.



(b) Relative phase of sensor nodes against addition/removal of sensor nodes. Sensor nodes 6 and 7 are added at $t = 40$, and sensor nodes 1 and 4 are removed at $t = 70$.

Figure. 12. Evaluation of basic behavior of the randomization-based mechanism.

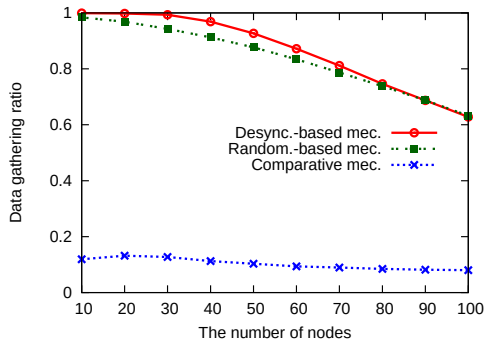
in Fig. 12. However, the order of relative phase is according to hop counts from the BS. When we compare the desynchronization-based mechanism and the randomization-based mechanism, the number of cycles required for the transient state in the randomization-based mechanism is smaller than that in the desynchronization-based mechanism. Since the randomization-based mechanism does not require configuration of phases among sensor nodes with the same hop count from the BS, the number of cycles in the transient state becomes smaller.

C. Comparative Evaluation with Conventional WAVE in the Steady State

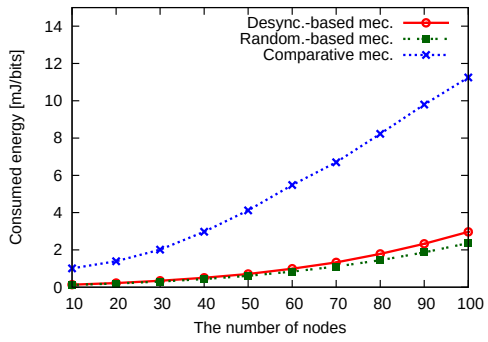
We next evaluate the performance of mechanisms in the steady state. In this evaluation, to evaluate the performance under the situations where a number of hidden nodes are deployed, a BS is located at the center of the monitoring region, and N sensor nodes are randomly distributed in the monitoring region. All results are from the 100th to 200th cycle after the start of simulations and averaged over 100 simulation experiments. Here, we should note that 100 cycles are enough duration for reaching the steady state in this simulation setting. For comparison, we also conduct WAVE data gathering simulation experiments that employ only the CSMA/CA collision avoidance mechanism among sensor nodes with the same hop count (referred as the *comparative mechanism*). Since previous studies have evaluated PCO-based mechanisms with non-PCO-based mechanisms in the viewpoint of data gathering delay, packet loss probability and so on [15], [19], [21], we evaluate the fundamental performance of our proposed mechanisms compared with the conventional PCO-based mechanism in this paper.

Figures 13(a) and 13(b) show the data gathering ratio and consumed energy ratio by number of sensor nodes, respectively. As shown in Fig. 13(a), the data gathering ratio in the desynchronization-based mechanism and the randomization-based mechanism is greatly improved compared to the comparative mechanism. Because the comparative mechanism synchronizes the phase of sensor nodes with the same hop count, collisions among two-hop neighbor sensor nodes are common, despite CSMA/CA implementation. On the other hand, because the phase of sensor nodes with the same hop count is dispersed in our proposed mechanisms, the number of collisions among

©2013 Engineering and Technology Publishing

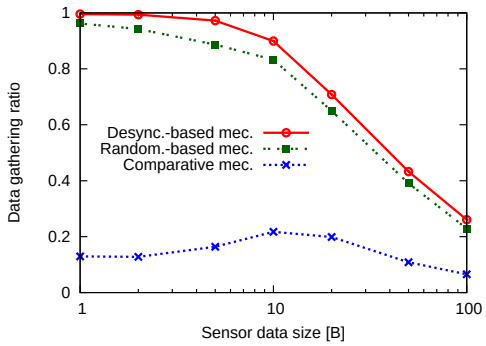


(a) Data gathering ratio

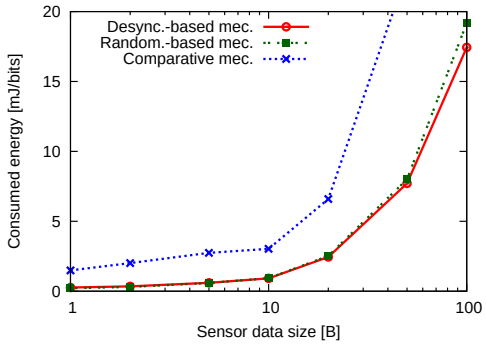


(b) Consumed energy ratio

Figure.13. Evaluation of our proposed mechanisms against number of sensor nodes.



(a) Data gathering ratio



(b) Consumed energy ratio

Figure.14. Evaluation of our proposed mechanisms against size of sensor data ($N = 30$).

two-hop neighbor sensor nodes can be reduced.

As shown in Fig. 13(a), in all mechanisms the data gathering ratio decreases as the number of sensor nodes increases. In the simulation, the maximum offset τ^{max} is set to a fixed value, 0.1 s. When the number of sensor nodes N in the monitoring region increases, the number of sensor nodes transmitting messages at a given time increases, increasing the number of collisions. In the simulation settings, the data gathering ratio of the desynchronization-based mechanism decreases when the number of sensor nodes exceeds 30, and approximately equals that of the randomization-based mechanism when the number of sensor nodes exceeds 70.

When we compare the desynchronization-based mechanism and the randomization-based mechanism, the data gathering ratio of the desynchronization-based mechanism is up to 6% higher. In addition, the data gathering ratio of the desynchronization-based mechanism is almost 100% when the number of sensor nodes is less than 30. This is because the desynchronization-based mechanism considers the hidden node problem. On the other hand, the message size in the desynchronization-based mechanism is larger than that of the randomization-based mechanism, because message transmission timing information is included. As a result, although the data gathering ratio of both mechanisms are similar when the number of sensor nodes exceeds 70, as shown in Fig. 13(b) the consumed energy ratio of the desynchronization-based mechanism is higher than that of the randomization-based mechanism.

Figures 14(a) and 14(b) show the data gathering ratio and consumed energy ratio by single sensor datum size, respectively, when $N = 30$. Similar to previous results, the data gathering ratio in both proposed mechanisms is greatly improved over the comparative mechanism, regardless of the sensor data size. When we compare our proposed mechanisms, the data gathering ratio of the desynchronization-based mechanism is up to 11% higher than that of the randomization-based mechanism, although the consumed energy ratios are similar.

From above results and discussions, we can conclude that the desynchronization-based mechanism is suitable for WSN applications which require high data gathering ratio. However, it requires additional control overhead to avoid collisions among hidden nodes. On the other hand, the randomization-based mechanism is suitable for WSN applications which require energy-efficiency and simplicity of mechanism rather than data gathering ratio. Although the randomization-based mechanism does not explicitly consider the hidden node problem, it is simpler and requires less control overhead.

©2013 Engineering and Technology Publishing

V. Conclusion and Future Work

In this paper, we proposed self-organizing transmission and sleep scheduling mechanisms for periodically gathering sensor data at a BS. The proposed mechanism applied phase-locking in a PCO model to schedule transmission timing of sensor nodes in order of hop counts from the BS. In addition, we proposed the randomization-based mechanism and the desynchronization-based mechanism using PCO anti-phase synchronization to avoid collisions among sensor nodes with the same hop count. Simulation experiments confirmed the effectiveness of the proposed mechanisms. In future work, we should confirm and improve the performance of our mechanisms in actual environments through experimental evaluations.

Acknowledgment

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

References

[1] Y. Taniguchi, G. Hasegawa, and H. Nakano, "Self-organizing transmission scheduling mechanisms using a pulse-coupled oscillator model for wireless sensor networks," in *Proc. ICDIPC 2012*, Jul. 2012, pp. 85–90.
[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communication Magazine*, vol. 40, pp. 102–114, Aug. 2002.
[3] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, Oct. 2004.
[4] F. Wang and J. Liu, "Networked wireless sensor data collection: Issues, challenges, and approaches," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 673–687, 2011.
[5] O. D. Incel, A. Ghosh, and B. Krishnamachari, "Scheduling algorithms for tree-based data collection in wireless sensor networks," in *Theoretical aspects of distributed computing in sensor networks*. Springer, 2011, pp. 439–477.
[6] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks," *Wireless Communications and Mobile Computing*, vol. 7, pp. 863–875, May 2007.
[7] M. R. Ahmad, E. Dutkiewicz, and X. Huang, "A survey of low duty cycle MAC protocols in wireless sensor networks," in *Emerging communications for wireless sensor networks*. InTech, 2011, pp. 69–90.
[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. IEEE HICSS 2000*, Jan. 2010, pp. 3005–3014.
[9] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, Oct. 2004.
[10] S. Lindsay and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Proc. IEEE Aerospace Conference 2002*, 2002, pp. 3–1125–3–1130.
[11] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proc. ACM SenSys 2005*, Nov. 2005, pp. 142–153.
[12] Y. Kubo and K. Sekiyama, "Communication timing control with interference detection for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, pp. 1–10, 2007.
[13] N. Wakamiya and M. Murata, "Synchronization-based data gathering scheme for sensor networks," *IEICE Transactions on Communications*, vol. E88-B, no. 3, pp. 873–881, Mar. 2005.
[14] H. Nakano, A. Utani, A. Miyauchi, and H. Yamamoto, "Data gathering scheme using chaotic pulse-coupled neural networks for wireles sensor networks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92-A, no. 2, pp. 459–466, Feb. 2009.
[15] J. Degesys, I. Rose, A. Patel, and R. Nagpal, "DESYNC: self-organizing desynchronization and TDMA on wireless sensor networks," in *Proc. ACM/IEEE ISPN 2007*, Apr. 2007, pp. 11–20.
[16] J. Degesys and R. Nagpal, "Towards desynchronization of multi-hop topologies," in *Proc. IEEE SASO 2008*, Oct. 2008, pp. 129–138.
[17] A. Mutazono, M. Sugano, and M. Murata, "Self-organizng anti-phase synchronization scheme for sensor networks inspired by frogs' calling behavior," *ISAST Transactions on Compututer and Intelligent Systtems*, vol. 1, no. 2, pp. 86–93, Dec. 2009.
[18] H. Hernández and C. Blum, "Implementing a model of Japanese tree frogs' calling behavior in sensor networks: A study of possible improvements," in *Proc. GECCO 2011*, Jul. 2011, pp. 615–622.
[19] 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.
[20] R. Pagliari, Y.-W. P. Hong, and A. Scaglione, "Bio-inspired algorithms for decenteralized round-robin and proportional fair scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 4, pp. 564–575, May 2010.
[21] Y. Taniguchi, A. Kanzaki, N. Wakamiya, and T. Hara, "An energy-efficient data gathering mechanism using traveling wave and spatial interpolation for wireless sensor networks," *Journal of Information Processing*, vol. 20, no. 1, pp. 167–176, Jan. 2012.
[22] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *Society for Industrial and Applied Mathematics Journal*, vol. 50, no. 3, pp. 1645–1662, Dec. 1990.
[23] P. Goel and B. Ermentrout, "Synchrony, stability, and firing patterns in pulse-coupled oscillators," *Physica D*, vol. 163, no. 3–4, pp. 191–216, Mar. 2002.
[24] "IEEE standard 802.15.4-2003," Oct. 2003.
[25] "MICAz Data Sheet," http://www.xbow.com/.

Yoshiaki Taniguchi received his B.E., M.E. and Ph.D. degrees from Osaka University, Japan, in 2002, 2004 and 2008, respectively. Since 2008, he has been an Assistant Professor at the Cybermedia Center, Osaka University. His research interests include wireless sensor networks, wireless mesh networks and object tracking systems. He is a member of IEEE, IEICE, IPSJ and IIEEJ.

Go Hasegawa received his M.E. and Ph.D. degrees from Osaka University, Japan, in 1997 and 2000, respectively. From 1997 to 2000 he was a Research Assistant at the Graduate School of Economics, Osaka University. He is currently an Associate Professor at the Cybermedia Center, Osaka University. His research is in the area of transport architecture for future high-speed networks. He is a member of IEEE and IEICE.

Hirotaka Nakano received his B.E., M.E. and D.E. degrees from the University of Tokyo, Japan, in 1972, 1974 and 1977, respectively. He joined NTT Laboratories in 1977 and has been engaged in research and development of videotex systems and multimedia-on-demand systems. He was an executive manager of the Multimedia Systems Laboratory at the NTT Human Interface Laboratories from 1995 to 1999. He was the head scientist of the Multimedia Laboratory at NTT DOCOMO until 2004. From 2004 to 2013, he was a Professor at the Cybermedia Center, Osaka University. His research is in the area of ubiquitous networks. He is a member of IEEE, IEICE, IIEEJ and ITE.

©2013 Engineering and Technology Publishing