Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 83 (2016) 568 – 575
Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

An efficient scheduling method based on pulse-coupled oscillator model for heterogeneous large-scale wireless sensor networks

Soichiro Yamanaka*[a], Masafumi Hashimoto[a], and Naoki Wakamiya[a]

[a] *Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan*

Abstract

Wireless sensor networks (WSNs) have been common networking technologies for data gathering applications. In order to collect necessary data effectively, such applications require large-scale WSNs many sensor nodes are deployed widely. As its solution, IEEE 802.11ah is promising. However, it operates at sub 1 GHz band that is license-free, which may result in that different service providers deploy WSNs for different purposes. This incurs serious collisions due to hidden nodes. Unfortunately, they often refuse cooperation among the others due to their service policies. Therefore, self-organized scheduling methods are needed without proactive cooperation. To this end, in this paper, we propose a self-organized scheduling method for large-scale WSNs, which is based on the pulse-coupled oscillator model. To avoid collisions effectively, the proposed method utilizes a phase response function that has attractors corresponding to time slots and a random mechanism for slot selection. Through simulation-based evaluation, we demonstrate that the proposed method can collect about 90% of data in a situation sensor nodes have different cycles of data gathering while achieving a reasonable convergence time. We also show its good flexibility for environmental changes.

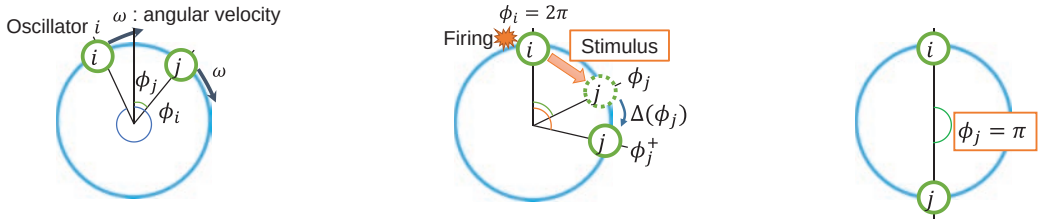
Peer-review under responsibility of the Conference Program Chairs
Keywords: biologically-inspired approach; pulse-coupled oscillators; phase response function; scheduling; wireless sensor networks

1. Introduction

Wireless sensor networks (WSNs) have been common networking technologies to realize data gathering applications including smart grid[1,2] and environmental monitoring[3,4]. One of purposes for smart grid applications is to monitor and collect real-time status of lifeline for gas, water, and electric power and so on. On the other hand, environmental monitoring applications collect environmental information and protect natural environments from negative outcomes. Such applications require large-scale WSNs many sensor nodes are deployed in wide areas in order to collect necessary information. Therefore, how to control a lot of sensor nodes and how to collect data from sensor nodes widely deployed are important issues for such applications.

* Soichiro Yamanaka. Tel.: +81-6-6879-4357; Fax: +81-6-6879-4359.
E-mail address: s-yamank@ist.osaka-u.ac.jp

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.271



(a) Phase of oscillators i, j (b) Phase change of oscillator j (c) Phase after synchronization

Fig. 1: Behavior of PCO

As an promising solution, IEEE 802.11ah[5,6] has been defined as an IEEE 802.11-based standard for wireless sensor networks at sub 1 GHz band, which aims to handle up to 6000 nodes in a single access point or a sink node. It also provides large transmission range up to 1 km. These features enable us to handle many sensor nodes in wide areas by a sink node, which leads to reduce the deployment cost. As a result, multiple WSNs are deployed at an overlapped area and nodes coexist that belong to different WSNs in a service area. As the number of nodes increases in a service area, scheduling of transmissions of nodes becomes important due to serious interference caused by transmissions.

However, sub 1 GHz band at which IEEE 802.11ah operates is license free band. As a result, different service providers may deploy WSNs for different purposes at an overlapped area. This incurs serious collisions due to hidden nodes. Unfortunately, they often refuse cooperation among the other providers due to their service policy. Therefore, in order to achieve effective scheduling, self-organized scheduling methods without proactive cooperation are preferable.

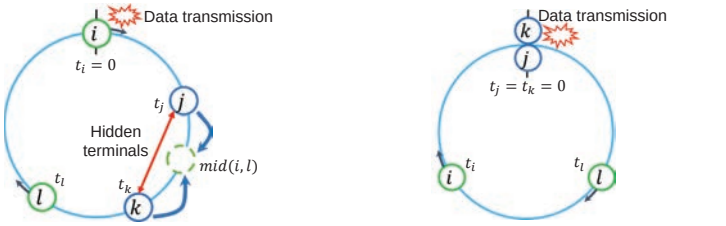
One of approaches to achieve self-organized scheduling is biologically-inspired scheduling based on the pulse coupled oscillator (PCO) model[7]. The PCO model is a mathematical model that describes synchronization behavior such as firefly flashing, frog calling, and pacemaker cells in a heart. In the PCO model, the phase of the oscillator is shifted by stimulus from the other oscillators. The strength of stimulus is determined by the phase response function as a function of own current phase. Scheduling algorithms based on the PCO model have been proved that they achieve effective and adaptive scheduling for environmental changes such as traffic fluctuations and topology changes[8,9,10,11]. For instance, Degesys *et al.*[8] proposed DESYNC that can adjust automatically participating nodes based on behavior of the PCO model. However, DESYNC including in other methods in[9,10,11] cannot be applied to overlapped WSNs because they assume ideal situations all nodes can communicate with each other. Consequently, they significantly increase collisions, especially, in a situation the hidden node problem[12] is non-negligible, which is serious in the situation where multiple WSNs are deployed at an overlapped area and there are many nodes in a service area. Therefore, it is need to be alleviated for large-scale single-hop WSNs.

To this end, in this paper, we propose an effective scheduling algorithm for large-scale WSNs, which is based on the PCO model that introduces new phase response function. In order to address the hidden node problem, the phase response function is designed to have attractors corresponding to time slots in WSNs. The proposed method further improves the data gathering rate by stochastic slot selection mechanism after detecting data collisions. Through simulation-based evaluations, we demonstrate that the proposed method can achieve high rate of data gathering with a reasonable convergence time and it is adaptive for environmental changes in both homogeneous and heterogeneous situations.

2. Pulse-coupled oscillator model

2.1. Overview

The PCO model is a mathematical model that describes the synchronization behavior in biological systems. In the PCO model, the behavior of an oscillator is modeled by two variables: phase and stimulus. Figure 1 shows the behavior of two coupled oscillators. Let $\phi_i \in [0, 2\pi]$ denote a phase of oscillator i as a function of angular velocity ω. Phase ϕ_i grows according to ω as time passes (Fig. 1(a)). When ϕ_i reaches 2π, oscillator i fires, which is a stimulus



(a) Updates of transmission timings of j and k after detecting the transmission from i

(b) Simultaneous transmission of j and k due to convergence on the same attractor

Fig. 2: Behavior of DESYNC with hidden nodes

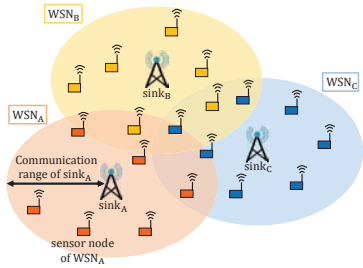


Fig. 3: Overlapped wireless sensor networks

for oscillator j that is coupled with oscillator i, as shown in Fig. 1(b). After firing, ϕ_i immediately becomes zero. On receiving the stimulus, oscillator j shifts its phase ϕ_j depending on the strength of the stimulus. The strength is determined by the current phase when receiving the stimulus, which is defined by phase response function $\Delta(\phi)$ where ϕ is a phase when receiving a stimulus. Let ϕ_j^+ be the phase of oscillator j after receiving a stimulus. The above behavior of oscillator j is modeled by

$$\phi_j^+ = \phi_j + \Delta(\phi_j). \tag{1}$$

$\Delta(\phi)$ determines whether the behavior of oscillators is synchronization or anti-phase synchronization. If we use $\Delta(\phi) = \sin\phi$, we obtain anti-phase synchronization; oscillator j finally converges on phase $\phi_j = \phi_i + \pi$, which is called *attractor*. Anti-phase synchronization can be applied to scheduling problems in WSNs. Unless otherwise noted, anti-phase synchronization is called just synchronization.

2.2. Scheduling based on PCO model and its problems

Anti-phase synchronization among of oscillators can be regarded as transmission schedules for data gathering applications that send data periodically in wireless networks when we make phases in the PCO model correspond to transmission timers sensor nodes have. Note that the transmission timer has inverse relationship with the phase; as time passes, the former decreases to zero whereas the latter increases to 2π. By doing so, DESYNC[8] successfully reduces data collisions and achieves high data gathering rate in single-hop WSNs. In DESYNC, each node records two transmission timings just before and just after it fires. After that, it updates the transmission timer so as to approach the middle of the recorded timings. Consequently, this mechanism achieves autonomous distributed scheduling realizing collision-free.

However, DESYNC may incur serious performance degradation in wireless networks with hidden node problem since it assumes the ideal situation in which all nodes can communicate with each other. For example, we consider a simple network where there are four nodes i, j, k, and l, which is depicted in Fig. 2. Nodes i and l can communicate with all nodes, whereas node j and k cannot communicate with each other, which implies that they are hidden nodes each other. We now focus on a case in Fig. 2(a) that when node i sends data, nodes j, k and l will receive it. On receiving it, nodes j and k update their timers based on the same transmission timings of nodes i and l since they are hidden nodes each other. As a result, nodes j and k converge on the same transmission timing, or attractor (Fig. 2(b)). Unfortunately, once different nodes converge on the same attractor, they cannot exit from the attractor on the PCO model, which results in data collisions. These problems can also occur in the other PCO-based methods in[9,10,11].

3. Assumptions

We assume that one or more WSNs are deployed in an area for data gathering purpose, as shown in Fig. 3. In each WSN, a sink node is associated with many sensor nodes. All nodes including sink and sensor nodes have the same communication range, i.e., all sensor nodes can communicate with the sink node to which they belong by one hop. They also send data to the sink node periodically at a certain cycle, which is denoted by C_i where i is an identifier of

a sensor node. Sensor node i sends data based on a transmission timer t_i that decreases as time passes; sensor node i sends data when $t_i = 0$. We also assume an automatic repeat request (ARQ), i.e., a sink node sends an acknowledgment (ACK) after receiving data correctly. All time periods from starting to send data until receiving the corresponded ACK are within a time slot, which is denoted by T. Sensor nodes do not retransmit data even if they fail to transmit it.

4. Proposed scheduling method

There are two problems described in Section 2.2; 1) hidden nodes converge on the same transmission timing when they receive data from the same two sensor nodes, and 2) sensor nodes cannot change transmission timings by interactions with other sensor nodes once they converge on the same timing. These problems are caused by two reasons; 1) the number of attractors is different among sensor nodes because they receive data from different sensor nodes, and 2) total amount of stimulus among sensor nodes is the same once they converge on the same timing.

Therefore, we propose a new scheduling method that addresses the two problems. In the proposed method, in order to alleviate the first problem, sensor nodes update their transmission timer based on a new phase response function that allows oscillators to converge on different attractors by making sensor nodes easy to converge on a nearby attractor corresponding to a time slot. The proposed method also exploits a random mechanism for slot selection to alleviate the second problem when detecting collisions, which is based on transmission information observed by each sensor node just before it starts sending data.

4.1. Overview of behavior of each node

Figure 4 shows the state transition diagram of the proposed method, which consists of five states: *Idle*, *Tx*, *Rx*, *Sleep*, and *WaitACK* states. All sensor nodes initially starts at Idle state. When detecting busy due to data transmissions by the other sensor nodes, sensor node i changes from Idle state to Rx state. On changing to Rx state, it updates t_i based on

$$t_i^+ = t_i + \Delta(t_i) \tag{2}$$

where $\Delta(t)$ is a phase response function that is described later in Section 4.2. After that, sensor node i backs to Idle state.

When t_i reaches zero, sensor node i starts a data transmission while its state is transited from Idle to Tx states. After finishing the data transmission, it waits for ACK sent from the sink node in WaitAck state. When receiving the ACK, it changes the state to Idle state and resets $t_i = C_i$. On the other hand, when the ACK timeout expires due to collisions, sensor node i resets t_i based on the random selection mechanism, which is described in Section 4.3. To save energy consumption, sensor nodes sleep when $t_i > n_i \cdot T$ where T is a time slot and n_i is the number of the remaining time slots to start sending data, i.e., when it has plenty of time for the data transmission. When $t_i \leq n_i \cdot T$, sensor node i wakes up and listens to data sent from the other nodes in Idle state.

4.2. Timer updates when detecting busy

In DESYNC, hidden nodes converge to the same transmission timing when they receive stimulus from the same nodes, which causes data collisions among the nodes. Unfortunately, the collisions occur every data transmission cycle. This is because few attractors are formed to the number of sensor nodes, e.g., only three attractors are formed to four sensor nodes in Fig. 2(b). In order to solve the problem, we need to form more attractors than total number of sensor nodes in the data transmission cycle. In addition, to avoid collisions, phase response functions should become zero if and only phases converge on attractors.

To this end, we propose a new phase response function $\Delta(t_i)$ that realizes the above, which is obtained by

$$\Delta(t_i) = \alpha \frac{T}{\pi} G(t_i) \sin\left(\frac{\pi}{T} t_i\right) \tag{3}$$

where α is a parameter for convergence speed and $G(t_i)$ is given by

$$G(t_i) = \begin{cases} 1 & \text{if } -\cos\left(\frac{\pi}{T} t_i\right) > 0 \\ -1 & \text{otherwise.} \end{cases} \tag{4}$$

The range of α is $0 < \alpha \leq 1$ because the timer needs to be updated so as to approach an attractor from one direction to avoid collisions.

However, time slots are discrete values in scheduling whereas phases are continuous values in the PCO model. As a result, attractors are needed to be formed at intervals of a time slot size T. In order to address this issue, sensor nodes pull back the transmission timers to the range of $(T, 2T]$ in the case that they detect busy when $t_i < T$. Therefore, $\Delta(t_i)$ when $t_i < T$ is determined by $\Delta(t_i) = -\beta t_i + T$ where β is a parameter for convergence speed and its range is $\frac{1}{2} \leq \beta \leq 1$ because of the same reason for α.

Assuming that sensor nodes sleep for power saving when $t_i > n_i \cdot T$, $\Delta(t_i)$ is finally calculated by

$$\Delta(t_i) = \begin{cases} -\beta t_i + T & \text{if } t_i < T \\ \alpha \dfrac{T}{\pi} G(t_i) \sin\left(\dfrac{\pi}{T} t_i\right) & \text{if } T \leq t_i < n_i \cdot T \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

4.3. Timer updates when detecting collisions

In the PCO model, oscillators continue to fire at the same timing once they converge to the same attractor, which results in data collisions in WSNs. In the case of convergence on the same attractor, we need how to distribute them to different attractors.

To this end, we utilize a random mechanism for slot selection, which is described as follows. Let C_i be the data transmission cycle of sensor node i. Let S_i be a set of empty time slots which are observed during $t_i \leq n_i \cdot T$, which is initialized to $S_i = \phi$ after the data transmission. On detecting collisions by the ACK timeout, sensor node i first tries to choose an empty slot by using information for the past empty time slots observed before the data transmission. More specifically, it chooses a slot s_i uniformly at random from S_i when $S_i \neq \phi$. Based on it, t_i is set to time corresponding to the chosen slots. In contrast, when $S_i = \phi$, sensor node i chooses a slot uniformly at random from the range of $(0, C_i - n_i \cdot T]$. Assuming that sensor nodes send successfully at the previous transmission. When they detect collisions, the collisions are likely to be caused due to the above mechanism. In such a situation, they are desirable to stay the time slots to avoid unnecessary collisions. However, if collisions caused by sensor nodes that have different cycles of transmissions, staying the same time slot is not suitable because transmissions do not always collide every transmission due to their different cycles. Therefore, when sensor nodes that succeeded the previous transmission detect collisions, they stay the same time slots with probability γ. Otherwise, they choose time slots based on the above random mechanism.

Summarizing the above, when detecting collisions, sensor node i sets its transmission timer t_i^+ for the next transmission as follows:

$$t_i^+ = \begin{cases} C_i - T & \text{with probability } \gamma \text{ if previous transmission succeeded} \\ F(S_i) - T & \text{otherwise} \end{cases} \tag{6}$$

$$F(S_i) = \begin{cases} C_i - \bar{s}_i & \text{if } |S_i| > 0 \\ random(0, C_i - n_i \cdot T) & \text{otherwise} \end{cases} \tag{7}$$

where $\bar{s}_i$ is a time slot chosen uniformly at random from S_i and $random(0, m)$ is a function that selects an integer uniformly at random from the range of $(0, m]$ $(m \in \mathbb{Z})$.

In order to find an empty slot, n_i is updated as whether the previous data transmission succeeds or not. When the previous data transmission fails, a sensor node observes further time slots and find empty slots, which improves the convergence time. Therefore, sensor node i increases n_i in such a situation. On the other hand, when the previous data transmission succeeds, a sensor node does not need to detect empty time slots. In this case, sensor node i decreases n_i to save energy. Therefore, n_i^+, which is n_i in the next cycle, is determined by

$$n_i^+ = \begin{cases} \min(2n_i, n_{\max}) & \text{if data transmission fails} \\ \max(n_i/2, n_{\min}) & \text{otherwise} \end{cases} \tag{8}$$

where $n_{\max}$ and $n_{\min}$ are maximum and minimum values of n_i, respectively.

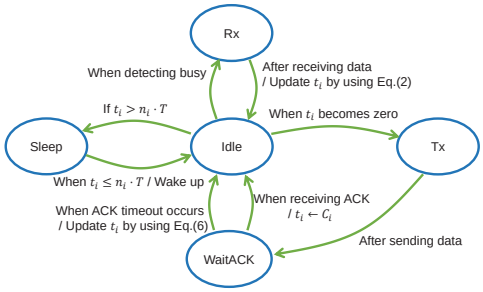


Fig. 4: State transition diagram of the proposed method

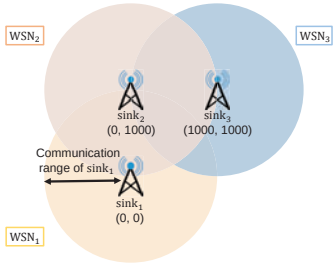


Fig. 5: Placement of WSNs

5. Evaluation

5.1. Simulation setting and method

To assess the performance of the proposed method, we run simulations with ns-3 [13]. Figure 5 shows simulation environments where three sink nodes are placed at coordinate (0, 0), (0, 1000), and (1000, 1000), respectively. WSNs to which these belong are denoted by WSN_1, WSN_2, and WSN_3, respectively. In each WSN, sensor nodes are placed uniformly at random around the sink node. All nodes including sink nodes have the same communication range, which is set to 1 km. All sensor nodes that belong to the same WSN send data by the same cycle. In addition, the time slot size is set to 5 ms.

In order to evaluate effectiveness of the proposed method, we consider with two different situations: homogeneous and heterogeneous cases. In a homogeneous case, all sensor nodes have same data transmission cycles. On the other hand, in a heterogeneous case, sensor nodes that belong to different WSNs have different data transmission cycles.

In a homogeneous case, data transmission cycles of sensor nodes in WSN_1, WSN_2, and WSN_3 are set to 60 s. In contrast, they are set to 20 s, 30 s, and 50 s, respectively, in a heterogeneous case. In all simulations, sensor nodes start sending data at random in the range of [0, their cycle]. Ten simulation experiments are conducted for statistical evaluation purposes. Therefore, results presented in this paper are average of ten simulations with the 95% confidence interval. We set total simulation time to 4000 s in all simulations. In the proposed method, we set α, β, and γ to 0.8, 1.0, and 0.9, respectively, which are better values obtained empirically. $n_{\min}$ and $n_{\max}$ are set to 2 and 50, respectively, in homogeneous WSNs. In heterogeneous WSNs, $n_{\max}$ increases to 100 to obtain better performance. For comparison purposes, we use DESYNC with parameter $\alpha_d = 0.95$, which is the best value in [8].

We compare the performance of the two methods in terms of two metrics: the data gathering rate and the convergence time. The data gathering rate is defined as the rate of the number of successful transmissions to total number of transmissions. On the other hand, the convergence time is defined as time when variation of data gathering rate stays within the 5 % band until end of the simulation. We assess the proposed method in the following scenarios.

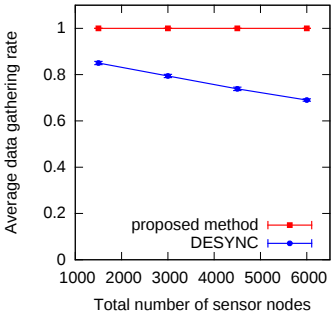
Scenario 1 All sensor nodes in WSN_1, WSN_2, and WSN_3 start sending data at the same time as the simulation starts.

Scenario 2 Sensor nodes in WSN_1 first start sending data at the same time as the simulation starts. At 1500 s, sensor nodes in WSN_2 and WSN_3 start sending data simultaneously. Finally, sensor nodes in only WSN_3 stop sending data at 3000 s.

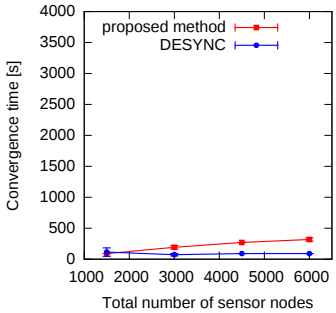
5.2. Simulation results

5.2.1. Homogeneous WSNs

Figure 6 shows simulation results in Scenario 1 while varying the total number of sensor nodes in WSNs, which consists of the average data gathering rate (Fig. 6(a)) and the corresponding convergence time (Fig. 6(b)). The average data gathering rate means that averaged value of data gathering rate obtained after 1000 s. In each simulation, three WSNs have the same number of sensor nodes, e.g., each WSN has 1000 sensor nodes when the total number of sensor nodes equals 3000 in Fig. 6.



(a) Data gathering rate



(b) Convergence time

Fig. 6: Simulation results in Scenario 1: Homogeneous case

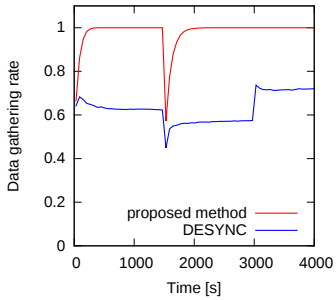
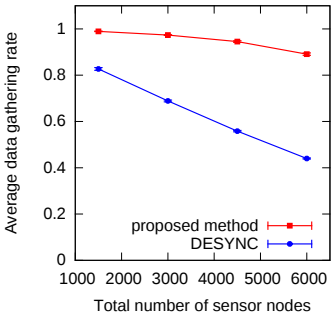
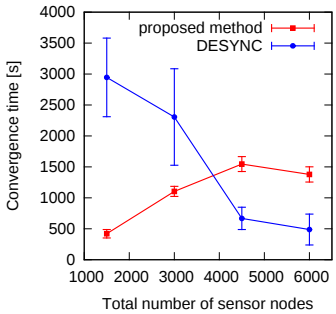


Fig. 7: Data gathering rate over time in Scenario 2: Homogeneous case



(a) Data gathering rate



(b) Convergence time

Fig. 8: Simulation results in Scenario 1: Heterogeneous case

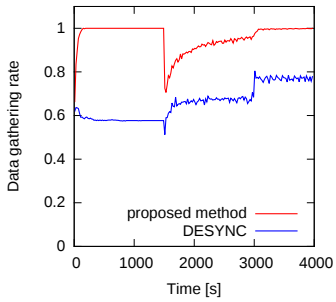


Fig. 9: Data gathering rate over time in Scenario 2: Heterogeneous case

From Fig. 6(a), we can see that the proposed method achieves 100% data gathering regardless of the number of sensor nodes, whereas DESYNC degrades the data gathering rate as the number of sensor nodes increases. The number of sensor nodes in hidden node relation increases as the number of sensor nodes increases. Therefore, the number of sensor nodes that converge to the same time slot increases. For this reason, the data gathering rate in DESYNC decreases as the number of sensor nodes increases. On the other hand, the proposed method can avoid collisions among sensor nodes in hidden node relation by the phase response function in Eq. (3) and the random mechanism in Eq. (6). In terms of the convergence time, we can see that the proposed method can converge rapidly although it takes further time than DESYNC. Even when the total number of sensor nodes increases, it takes about 300 s, which corresponds to about five cycles of the data gathering. Use of observed information of empty slots contributes to reduce convergence time.

We then evaluate adaptability of the proposed method for environmental changes in Scenario 2. The result is shown in Fig. 7 where 3000 sensor nodes exist in each WSN, i.e., total of 9000 nodes. On the start of the simulation, the proposed method rapidly converges on 100% data gathering rate. In contrast, DESYNC also converges speedily but its data gathering rate achieves only about 0.6. When sensor nodes start sending data in WSN_2 and WSN_3 at 1500 s, both the proposed method and DESYNC degrade the data gathering rate temporarily. However, the proposed method improves to 100% of the data gathering rate at few cycles whereas DESYNC still degrades the data gathering rate. At 3000 s, finishing data transmissions of nodes in WSN_3 incurs no effects for the proposed method. The reason for this is that the number of attractors is fixed in the proposed method regardless of environmental changes whereas in DESYNC it changes depending on interaction among neighbor nodes. From the above results, we conclude that the proposed method is adaptive for environmental changes in the homogeneous situation.

5.2.2. Heterogeneous WSNs

Figure 8 depicts results for the heterogeneous situation corresponding to Fig. 6. Comparing Figs. 8(a) and 6(a), when the total number of sensor nodes is 6000, data gathering rates of the proposed method is slightly degraded whereas that of DESYNC is reduced by about 30% compared with the homogeneous case. In terms of the convergence time, Fig. 8(b) shows that the convergence time in DESYNC is large when the total number of sensor nodes is small. This results from that sensor nodes change their data transmission timing largely because α_d, a parameter for convergence speed in DESYNC, is large. On the other hand, Fig. 8(b) shows that the convergence time in the proposed method increases largely compared with the homogeneous case. This is because coexistence of sensor nodes that have different cycles of data gathering causes a significant decrease in time slots not to collide transmissions among the other sensor nodes. However, utilization is about 0.75 when the total number of sensor nodes is 4500, which means that the proposed method can be used in high load situations while achieving a reasonable convergence time.

We finally present results of adaptability for environmental changes in a heterogeneous case in Fig. 9, corresponding to Fig. 7. Figure 9 shows the almost same tendencies for the results in Fig. 7. Note that the proposed method takes longer time to converge when joining many sensor nodes at 3000 s. In such a situation, increasing $n_{\max}$ is effective for improving the convergence time. However, increasing $n_{\max}$ means that sensor nodes stay awake for long time, which consumes further energy. Therefore, we need to set $n_{\max}$ while considering its trade-off relationship.

6. Conclusion

In this paper, we proposed the self-organized scheduling method based on the PCO model for large-scale single-hop WSNs. The proposed method uses the improved phase response function and the random mechanism for slot selection in order to improve the degradation of the data gathering rate due to the hidden node problem. From the simulation results, we conclude that the proposed method can achieve high rate of data gathering with the reasonable convergence time in both homogeneous and heterogeneous situations. Furthermore, the proposed method is also adaptive for environmental changes such as an increase and decrease in sensor nodes. As future work, we plan to improve the convergence time of the proposed method in heterogeneous situations.

References

1. Sankar, L., Rajagopalan, S.R., Mohajer, S., Poor, H.V.. Smart meter privacy: A theoretical framework. *smart grid, IEEE transactions on* 2013;**4**(2):837–846.
2. Makonin, S., Popowich, F., Gill, B.. The cognitive power meter: Looking beyond the smart meter. In: *Electrical and Computer Engineering (CCECE), 2013 26th Annual IEEE Canadian Conference on*. IEEE; 2013, p. 1–5.
3. Chang, H., Zhou, N., Zhao, X., Cao, Q., Tan, M., Zhang, Y.. A new agriculture monitoring system based on WSNs. In: *Proceedings of 12th International Conference on Signal Processing (ICSP), 2014*. IEEE; 2014, p. 1755–1760.
4. Liu, X., Cao, J., Song, W.Z., Tang, S.. Distributed sensing for high quality structural health monitoring using wireless sensor networks. In: *Proceedings of IEEE 33rd Real-Time Systems Symposium (RTSS), 2012*. IEEE; 2012, p. 75–84.
5. Aust, S., Prasad, R.V., Niemegeers, I.G.M.M.. IEEE 802.11ah: Advantages in standards and further challenges for sub 1 GHz Wi-Fi. In: *Proceedings of IEEE International Conference on Communications*. 2012, p. 6885–6889.
6. Sun, W., Choi, M., Choi, S.. IEEE 802.11 ah: A long range 802.11 WLAN at sub 1 GHz. *Journal of ICT Standardization* 2013;**1**(1):83–108.
7. Goel, P., Ermentrout, B.. Synchrony, stability, and firing patterns in pulse-coupled oscillators. *Physica D: Nonlinear Phenomena* 2002; **163**(3):191–216.
8. Degesys, J., Rose, I., Patel, A., Nagpal, R.. DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks. In: *Proceedings of International Symposium on Information Processing in Sensor Networks*; 1. IEEE. ISBN 978-1-59593-638-7; 2007, p. 11–20.
9. Mutazono, A., Sugano, M., Murata, M.. Energy efficient self-organizing control for wireless sensor networks inspired by calling behavior of frogs. *Computer Communications* 2012;**35**(6):661–669.
10. Hernández, H., Blum, C.. FrogSim: distributed graph coloring in wireless ad hoc networks. *Telecommunication Systems* 2013;**55**(2):211–223.
11. Choochaisri, S., Apicharttrisorn, K., Korprasertthaworn, K., Taechalertpaisarn, P., Intanagonwiwat, C.. Desynchronization with an artificial force field for wireless networks. *ACM SIGCOMM Computer Communication Review* 2012;**42**(2):7–15.
12. Cano, C., Bellalta, B., Cisneros, A., Oliver, M.. Quantitative analysis of the hidden terminal problem in preamble sampling WSNs. *Ad Hoc Networks* 2012;**10**(1):19–36.
13. *ns-3*. available at https://www.nsnam.org/.