PAPER *Special Section on Ubiquitous Networks*

Synchronization-based Data Gathering Scheme for Sensor Networks*

Naoki WAKAMIYA[†] ***and*** **Masayuki MURATA**[†], ***Members***

SUMMARY By deploying hundreds or thousands of microsensors and organizing a network of them, one can monitor and obtain information of environments or objects for use by users, applications, or systems. Since sensor nodes are usually powered by batteries, an energy-efficient data gathering scheme is needed to prolong the lifetime of the sensor network. In this paper, we propose a novel scheme for data gathering where sensor information periodically propagates from the edge of a sensor network to a base station as the propagation forms a concentric circle. Since it is unrealistic to assume any type of centralized control in a sensor network whose nodes are deployed in an uncontrolled way, a sensor node independently determines the cycle and the timing at which it emits sensor information in synchrony by observing the radio signals emitted by sensor nodes in its vicinity. For this purpose, we adopt a pulse-coupled oscillator model based on biological mutual synchronization such as that used by flashing fireflies, chirping crickets, and pacemaker cells. We conducted simulation experiments, and verified that our scheme could gather sensor information in a fully-distributed, self-organizing, robust, adaptive, scalable, and energy-efficient manner.

key words: *sensor network, data gathering, pulse-coupled oscillator*

1. Introduction

With the development of low-cost microsensor equipment having the capability of wireless communications, sensor network technology [1] has attracted the attention of many researchers and developers. A sensor node is equipped with one or more sensors with analog/digital converters, a general purpose processor with a limited computational capacity, a small amount of memory, low-cost radio transceiver, and a battery power supply. By deploying a large number of multifunctional sensors in a monitored region and composing a sensor network of them, one can remotely obtain information on behavior, condition, and position of elements in the region. Sensor nodes monitor the circumstances and periodically or occasionally report sensed phenomena directly or indirectly to the base station, i.e., the sink of sensor information, using wireless communication channels. Sensor networks can be used in agricultural, health, environmental, and other industrial applications. More specifically, Intelligent Transportation Systems (ITS) and ubiquitous or pervasive computing are typical examples that benefit from information gathered from circumstances and environments.

Sensor nodes are distributed in a region in an uncontrolled and unorganized way to decrease the installation cost and eliminate the need for careful planning. Thus, the method used to gather sensor information should be scalable to the number of nodes, robust to the failure and disruption of nodes, adaptive to addition, removal, and movement of nodes, inexpensive in power consumption, and fully distributed and self-organizing without a centralized control mechanism. Several research works have been done in developing schemes for data gathering in sensor networks, such as [2]–[6]. However, schemes proposed in [2]–[4] require so-called global information such as the number of sensor nodes in the whole region, the optimal number of clusters, the locations of all nodes, and the residual energy of all nodes. Consequently, they needs an additional, and possibly expensive and unscalable, communication protocol to collect and share the global information. Thus, it is difficult to adapt to the dynamic addition, removal, and movement of sensor nodes. A distributed version of [2] was proposed in [5] and [6] proposed a scheme to organize sensor nodes into clusters in a distributed and self-organizing way to prolong the lifetime of a sensor network. However, they still consume much energy in cluster formation.

In this paper, we propose a novel and efficient scheme for gathering data in sensor networks where a large number of sensor nodes are deployed. We consider an application that periodically collects sensor information from distributed nodes to a base station. Since the energy consumption in data transmission has a factor proportional to the square or the 4th of the range of radio signals [2], [5], we adopt a multi-hop communication scheme. Sensor information is propagated from the edge of a sensor network to the base station by being forwarded by intermediate nodes. We do not assume that all nodes are visible to each other as in other research work. An administrator does not need to configure sensor nodes before deployment. Our scheme does not rely on any specific routing protocol, and it can be used on any medium access protocol.

Manuscript received January 1, 2003.
Manuscript revised January 1, 2003.
Final manuscript received January 1, 2003.

[†] Graduate School of Information Science and Technology, Osaka University

[*] This work was partly supported by “The 21st Century Center of Excellence Program” and a Grant-in-Aid for Scientific Research (A)(2) 16200003 from The Japanese Ministry of Education, Culture, Sports, Science and Technology.

In periodic data gathering, power consumption can be effectively saved by reducing the amount of data to send, avoiding unnecessary data emission, and turning off unused components of a sensor node between data emissions. As an example, such data gathering can be attained by the following strategy. First, sensor nodes on the edge of a sensor network, i.e., that are the most distant from the base station, simultaneously emit their sensor information within the range of radio signals. Among neighboring nodes, those closer to the base station receive information. They aggregate the received information with local sensor information to reduce the amount of data to send. Then, they emit it at a timing that is synchronized with the other nodes on the same distance from the base station. Likewise, sensor information is propagated and aggregated to the base station. As a result, we observe a concentric circular wave of information propagation centered at the base station. In this scenario, which we call the synchronized data gathering, each node only needs to periodically turn on its transceiver to receives sensor information of more distant nodes and emit an aggregated information within a limited range of radio signals.

To accomplish the synchronized data gathering without any centralized controls, each sensor node should independently determine the cycle and the timing at which it emits a message to advertise its sensor information based on locally available information. The ideal synchronization can be attained by configuring sensor nodes prior to the deployment, provided that the clocks are completely synchronized, sensor nodes are placed at the appropriate locations, and they maintain their clocks through their lifetime. However, we cannot realistically expect such an ideal condition. A clock synchronization method [7] is helpful to some extent, but it consumes energy for clocks with much skew of widely deployed sensor nodes to keep synchronized.

Self-organized and fully-distributed synchronization can be found in nature. For example, fireflies flash independently, at their own interval, when they are apart from each other. However, when a firefly meets a group, it adjusts an internal timer to flash at the same rate as its neighbors by being stimulated by their flashes. Consequently, fireflies in a group flash in synchrony. Mutual synchronization in a biological system is modeled as pulse-coupled oscillators [8]–[11]. Each oscillator O_i has a state x_i, which is determined by a monotonically increasing function $f_i : [0,1] \to [0,1]$ of a phase ϕ_i. The phase cyclically shifts as time passes. When the state reaches one, an oscillator fires a pulse and goes back to the initial state $x_i = 0$. The pulse stimulates other oscillators within a range of pulse propagation and raises their state x_j by some amount of $\epsilon_i(\phi_j)$ [10]. Those oscillators whose states are raised to one also fire at this time. They are regarded as synchronized. In this way, all oscillators reach synchronization through mutual interactions. When we adopt a model of pulse-coupled oscillators to sensor networks, each sensor node can independently determine the cycle and the timing at which it emits a message to achieve synchronization with those neighboring nodes by observing the signals that neighboring nodes emit. Each sensor node only emits its sensor information at its own timing. There is no need for any additional communications that consume invaluable battery power as a synchronization method does. Thus, we can obtain a fully distributed, self-organizing, robust, adaptive, scalable, and energy-efficient scheme for data gathering in wireless sensor networks.

The rest of the paper is organized as follows. First, in Section 2, we briefly introduce our assumptions on sensor networks and propose our synchronization-based data gathering scheme. Then, we show some simulation results in Section 3. Section 4 discusses some additional considerations of our scheme. Finally, we conclude the paper in Section 5.

2. Synchronization-based Data Gathering Scheme

In this section, we first identify some assumptions of our scheme and then give detailed descriptions of our proposed data gathering scheme.

2.1 Sensor Networks

Sensor networks that our scheme assumes have the following characteristics. Components of a sensor network are hundreds or thousands of sensor nodes and a base station. A base station is a node to which all sensor information is gathered and from which users, administrators, applications, and systems can obtain sensor information. The base station is placed at a preferable location within the range of a radio signal from one or more sensor nodes. Sensor nodes are deployed in an uncontrolled way. Sensor nodes are dynamically introduced to monitor the region more densely or to replace dead sensor nodes. A sensor node stops operating when its battery is depleted. A sensor node might be moved to another place. A sensor node does not have any means of identifying its geographical location. A sensor node is prone to failure.

A sensor node monitors its surroundings and obtains sensor information. A sensor node can hear radio signals from other nodes. A sensor node aggregates its local sensor information and the information received from other sensor nodes [2], [12]. A sensor node has a timer. Its phase shifts as time passes, but the timer can be adjusted to an arbitrary point. When a timer expires, it is initialized to zero. According to the phase of a timer, or the state defined by its phase, a sensor node emits its sensor information, possibly aggregated with that of other nodes, without waiting for the reception of sensor information from other sensor nodes.

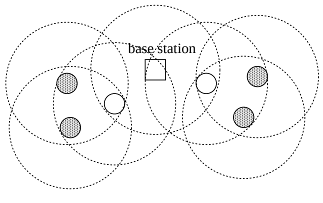

Fig. 1 An example of a sensor network

A sensor node has an unidirectional antenna. Thus, broadcasting is the major means of data emission. The propagation range of radio signals is limited. Information emitted by a sensor node can be received by other sensor nodes within the range of a radio signal.

We do not assume any specific MAC protocol. We can adapt CSMA/CA, FDMA, and CDMA, but we prefer CSMA/CA in this paper for its simplicity. Our scheme does not rely on any specific routing protocol. We do not assume any specific routing protocol but apply the most suitable, whether it be a flat or hierarchical, multi-hop, tree- or star-based routing protocol. The routing protocol determines a single sensor node or a set of sensor nodes that a sensor node can communicate with. In this paper, a message is emitted without specifying receiver. Instead, a message emitted by a sensor node is received by all sensor nodes in its vicinity, i.e., the range of a radio signal.

2.2 Scalable and Robust Data Gathering

First, we give a brief explanation of the basic behavior of data gathering in our scheme. Consider the network of one base station and six sensor nodes as in Fig. 1. Dashed circles stand for the ranges of radio signals.

We define the *level* of each sensor node as the number of hops from the base station. Two sensor nodes that can receive a radio signal of the base station are regarded on level 1 (open circle). Four nodes that can directly communicate with nodes on level 1 are on level 2 (filled circle). Information propagates from sensor nodes on the highest level to the base station. When we consider periodic data gathering, it is efficient in terms of power consumption that sensor nodes on the same level synchronously inform their parents of their sensor information. In addition, since each node emits its sensor information at its own timing without waiting for the reception of sensor information from other nodes, the nodes must emit their information at a time slightly before their parents emit information. For example, if the base station needs information about the region at time t, sensor nodes on level 1 simultaneously emit their information at $t-\delta$. Since emission is synchronized among sensor nodes on the same level, δ should be appropriately chosen so that all sensor nodes on the same level can successfully emit their information despite the existence of collisions on the medium access layer. For sensor nodes on level 1 to reflect information gathered on the higher level, all four nodes on level 2 should emit their information at $t-2\delta$ in synchrony with each other. Consequently, if there are level 3 nodes, they emit their information at $t-3\delta$. If such synchronized data gathering is attained, the radio component of a sensor node needs to be turned on only for δ out of the data gathering interval in this example.

Sensor nodes belonging to the same level have to be synchronized, even if they are geographically apart. In the above example, synchronization is needed for two sets of sensor nodes, i.e., two open-circle nodes and four filled-circle nodes. In addition, a set of synchronized nodes has to synchronize with another set that is closer to the base station but with a gap of δ.

To attain such inter- and intra-level synchronizations, we adapt the pulse-coupled oscillator model. The base station emits a beacon signal at a regular interval to make sensor nodes within the range of its radio signal synchronize with each other. We denote a set of N sensor nodes as $\mathcal{S}=\{S_1,\cdots,S_N\}$. Sensor node S_i belongs to level l_i. Initially, level l_i is set to infinity or a reasonably large value. It has a timer and a state x_i. A state is given by a monotonically increasing function $f_i:[0,1]\rightarrow[0,1]$ of a phase ϕ_i of the timer.

$$x_i = f_i(\phi_i) \tag{1}$$

For example, we used the following f_i in this paper.

$$\forall i,\quad f_i(\phi_i)=\frac{1}{b}\ln[1+(e^b-1)\phi_i] \tag{2}$$

This formula is taken from [8], [10]. $b>0$ is one of parameters that dominate the rate of synchronization [8]. As the dissipation b increases, f_i raises more rapidly and, as a result, synchrony emerges more rapidly. To take into account the offset δ_i, we consider a regulated phase ϕ'_i, which is given by the following equation.

$$\begin{aligned}\phi'_i &= p(\phi_i,\delta_i)\\ &= \begin{cases}\phi_i+\delta_i, & \text{if } \phi_i+\delta_i\le 1\\ \phi_i+\delta_i-1, & \text{otherwise}\end{cases}\end{aligned} \tag{3}$$

From ϕ'_i, we obtain a regulated state x'_i by $f_i(\phi'_i)$. Sensor node S_i emits a message when its regulated state x'_i becomes one. Thus, it fires δ_i earlier than state x_i reaches one.

At time t, sensor node S_j receives a message from sensor node S_i, which is specified as S_j's next node to the base station by a routing protocol or whose level l_i is smaller than S_j's level l_j. It is stimulated and its state changes as

$$x_j(t^+) = B(x_j(t)+\epsilon). \tag{4}$$

Function B is defined as,

$$B(x)=\begin{cases}x, & \text{if } 0\le x\le 1\\ 0, & \text{if } x<0\\ 1, & \text{if } x>1\end{cases}\quad. \tag{5}$$

The regulated state x'_j of stimulated node S_j is given as $x'_j = f_j(p(g_j(x_j(t^+)), l_j\delta_j))$ where $g_j = f_j^{-1}$. When node S_j's regulated state x'_j becomes one, it also emits a message in synchrony with node S_i. Since collisions occur on the medium access layer, sensor node S_i ignores messages from t to $t+\delta_i$ when it has already been stimulated at t to avoid being stimulated by deferred signals, as fireflies do.

A message that sensor node S_i emits to advertise its information contains the level l_i, a stimulus ϵ with which it stimulates nodes around it, δ for its children to use an identical offset, and its sensor information possibly aggregated with other sensor information. The number of bits needed for the level identifier is as many as several bits. If the number of levels exceeds the bits assigned to the level identifier, we can use those bits in a cyclic way. The stimulus ϵ and the offset δ take decimal fractions between zero and one. If we use four bits for the level identifier, three bits as the exponent, and nine bits for the fraction, a total of twenty-eight bits are needed. References [2], [3], [5] used 2000-bit messages and [4] used 1000-bit messages. Consequently, our protocol is 1.2% and 2.4% more expensive, respectively, than those protocols in power consumption for message exchange.

The level that sensor node S_i belongs to is given as the smallest level, say l_j, among messages that sensor node S_i can receive plus one, i.e., $l_i = l_j + 1$. A beacon signal from the base station advertises the level zero. When a new sensor node occasionally receives a message from a faraway sensor node, it first wrongly determines its level. As time passes, however, it receives another signal from a sensor node that is closer to the base station. At this point, it finally identifies its level correctly. We show an example of such a transition of level identification later. Since a sensor node ignores a message from a sensor node whose level is the same or higher for synchronization, there is no direct interaction among sensor nodes on the same level. Therefore, intra-level synchronization is attained through inter-level stimulus.

To summarize, the basic behavior of our sensor network can be explained as follows. We first consider the initial stage of deployment where all sensor nodes are introduced to a region. The base station begins to emit the beacon signal at the regular interval of data gathering. All sensor nodes initialize their levels to infinity or a reasonably large value. They also initialize their timers. Each sensor node begins to sensor its surroundings and stores sensor information into its memory. When the regulated state reaches one, it emits a message to advertise its sensor information, level, function ϵ, and offset δ. When it receives a message from another sensor node, it first compare its level with the level in the message. If the former is smaller than the latter by more than two, it ignores the message. If the former is smaller than the latter by one, it aggregates received sensor information with its locally stored information. Finally, if the former is larger than the latter, the sensor node is stimulated. It adjusts its level and raises its state x. A stimulated sensor node begins to emit a message that carries sensor information stored in its memory when the regulated state x' reaches one. If the state x reaches one by being stimulated, those two sensor nodes are synchronized at this time. Once synchronization is attained, a sensor node switches to a battery-saving mode.

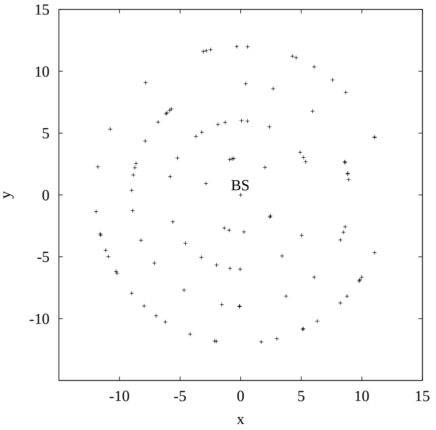

Fig. 2 An example of concentric circular sensor networks

Next, we consider the case where a new sensor node is introduced in a sensor network in operation. Initially, a new sensor node does not synchronize with any other sensor nodes. It monitors its surroundings, emits sensor information, and receives messages from sensor nodes in its vicinity, as in the above case. Being stimulated several times, its level becomes correctly identified, and its timer synchronizes with that of a sensor node whose level is smaller by one. When a sensor node disappears due to battery depletion or movement, a sensor node that is synchronized with the vanished node will be stimulated by another that is audible. If there is no other node with smaller level in its vicinity, the sensor node first becomes isolated. Since it does not receive stimuli any more, it can recognize the isolation and then it initializes its own level so that it can synchronize with other neighboring sensor nodes. When a sensor node moves, it first initializes its level to a large value in order to identify its new level while avoiding disturbing other nodes. Then, it behaves as a new node and attain the synchronization with new neighboring node.

3. Simulation Results

We employ a concentric circular sensor network for an easier understanding in the following experiments. We have confirmed that our protocol can successfully achieve desirable results on any sensor network with an arbitrary distribution of sensor nodes. The base station

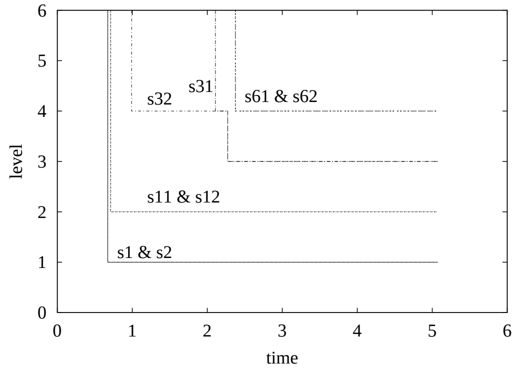

Fig. 3 Transition of levels

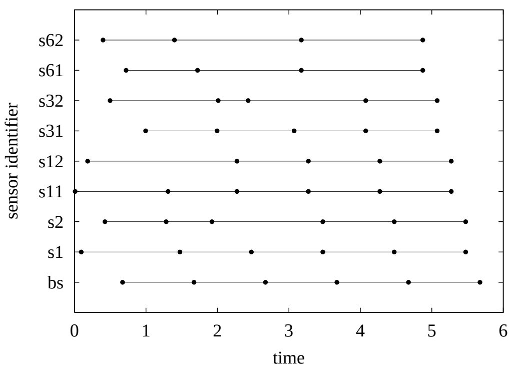

Fig. 4 Timing of message emission

is assumed to be located at the center of the region. The range of radio signals is identical among sensor nodes, and the radius is fixed at five units of length. Sensors are randomly placed on circumferences of a concentric circle whose center is the base station. The n-th circle has a radius of $3n$ units of length. For example, the second circle has a radius of six units of length. Sensors are placed from the innermost circle. When the number of sensor nodes on a circumference of the n-th circle reaches $10n$, then the subsequent sensor nodes are placed on the circumference of the $(n+1)$-th circle. An example of a simulated network is illustrated in Fig. 2 for 100 sensor nodes. Thus, when sensor nodes are numbered from the first node placed, the correct level of a sensor node can be calculated from its identifier. This allows easier investigation but does not restrict the applicability of our scheme.

The phase-state functions f_i are identical among sensor nodes and defined by (2). In the following experiments, we used $b = 3.0$ [10] and $\epsilon = 0.3$ [8]. Offset values δ_i are also identical as $\delta_i = 0.2$. This means that sensor nodes on the n-th circle emit their messages faster than the beacon by $0.2n$ units of time. We call this condition "the sensor network reaching global synchronization by our scheme." In the experiments, we ignore the propagation delay of a radio signal and the collision of radio signals on the medium access layer. In an actual situation, δ must be large enough when there are many sensor nodes to take into account collisions. However, since sensor nodes on the different levels have different phases in our scheme, the possibility of collision is reduced. In addition, no routing protocol is employed in simulation experiments. With our proposed scheme, sensor information propagates to the center of the circle without a help of routing protocols.

3.1 Basic Behavior

First, we show simulation results for the case where the sensor network has 100 sensor nodes. Initial states of the sensor nodes take random values from 0.0 to 1.0 that follow a uniform distribution. A simulation experiment stops when a sensor network reaches global synchronization. In this section, we assume that timers of sensor nodes have the same timer period. Thus, timers expire at the same frequency. When there exist timers with different frequencies, the fastest timer would dominate the synchronization as stated in [8]. Thus the frequency of data gathering, which is controlled by the interval of beacon signals from the base station, should be the smallest in the sensor network.

Figure 3 illustrates the transitions of levels of sensor nodes s1 and s2 on the first, s11 and s12 on the second, s31 and s32 on the third, and s61 and s62 on the fourth circle. Initially, their levels are set to reasonably large values. When a sensor node receives a radio signal from a node whose level has already been determined, it can identify its level. In the figure, nodes s1 and s2, which are on the innermost circle, received a beacon signal at time 0.673 and found that their levels were one. Then, nodes s11 and s12 received radio signals from sensor nodes on the first circle at 0.712 and set their levels to two. Nodes s31 and s32 occasionally first received a radio signal from a sensor node on the same circle, i.e., the third one. As a result, they wrongly identified their levels as four at 2.11 and 0.990, respectively. However, at 2.27, they received a radio signal from a sensor node on the second circle and changed their levels to three. In this example, global synchronization was accomplished at 5.07.

Figure 4 shows how the sensor network reaches the global synchronization. Dots on lines stand for instants when sensor nodes emit messages. It can be seen that each sensor node first flashed independently of the others based on its local timer. However, as time passed, sensor nodes on the same circle became synchronized by being stimulated by radio signals that sensor nodes on the inner circle emitted. They began to flash in synchrony with other nodes on the same circle and earlier than nodes on the inner circle by the offset, $\delta = 0.2$. Finally, global synchronization was accomplished at 5.07. Observing the rightmost dots on all nine nodes, it can be seen that sensor nodes emit messages in synchrony

Table 1 Summary of transition of timing of message emission

	level1	level2	level3	level4
1	0.120960	0.152091	0.258217	0.286049
2	0.103793	0.081072	0.215027	0.251488
3	0.045114	0.043577	0.164458	0.167781
4	0.000000	0.023744	0.057024	0.053984
5	0.000000	0.000000	0.000000	0.047049
6	0.000000	0.000000	0.000000	0.000000

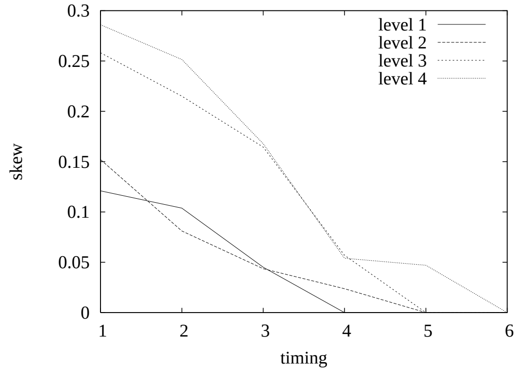

Fig. 5 Transition of timing of message emission

at exactly 0.2 units of time ahead of the data emission by sensor nodes on the inner circle.

Table 1 summarizes the transition of timing of message emission. For example, value 0.120960 in the table means that sensor nodes on the innermost circle emitted their first message at the timing earlier or later than the appropriate instant by 0.120960 on average. The table also shows that sensor nodes eventually attained the synchronization as time passed. In addition, it can be seen that sensor nodes on the outer circle took longer time to become synchronized. Figure 5 illustrates the transition. Once the global synchronization was accomplished, it was never lost.

3.2 Dynamic Deployment and Removal of Sensor Nodes

The following figures were obtained from simulation experiments where sensor nodes were deployed in the region and stopped working at random one by one. The time that a sensor node was deployed follows a uniform distribution from 0.0 to 10.0 units of time. The time to stopping a sensor node follows a uniform distribution from 15.0 to 25.0 units of time.

Figure 6 illustrates how newly introduced sensor nodes identify their levels. The level is initially set to a large value, but is gradually adapted as it encounters another sensor node through reception of radio signals, as described in Section 2.2. Since we cannot give a detailed explanation of the figure due to space limitation, we focus on sensor node s1 on the innermost circle, whose trajectory is depicted with a solid line. Sensor node s1 was deployed at 9.05. It first received a radio signal from a sensor node on the second circle and wrongly considered it to be on the third circle. Then, it observed a radio signal from a sensor node on the first circle at 9.25, and changed its level to two. Finally, at 9.45, it received a beacon, i.e., a radio signal that the base station emits. Then sensor node s1 identified its level as one. We can expect similar transition to the global synchronization during the movement of a sensor node if it initializes its own level while moving.

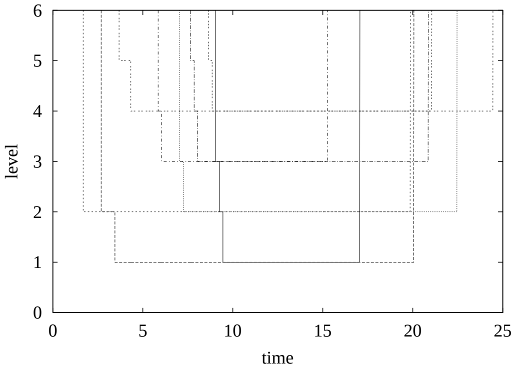

Fig. 6 Transition of levels (dynamic deployment)

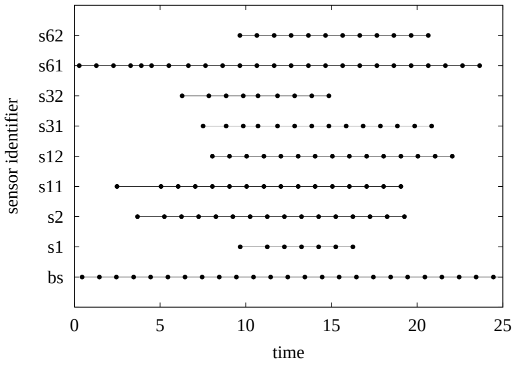

Fig. 7 Timing of message emission (dynamic deployment)

Figure 7 shows a series of message emissions of sensor nodes s1, s2, s11, s12, s31, s32, s61, and s62, as in Fig. 4. In this experiment, global synchronization was attained at 13.7. It is obvious from the figure that sensor nodes do not lose synchronization once they are fully synchronized even if sensor nodes disappear. Figure 8 illustrates the transition of timing of message emission. Fluctuations in the figure indicate that sensor nodes that had already attained the synchronization lost their appropriate timing by being stimulated by newly deployed sensor nodes. As a result, it took longer to attain the global synchronization than in the case where sensor nodes were deployed all at once.

From the above observations, we can conclude that our scheme can adapt to the dynamic changes in sensor networks, including the addition, removal, and movement of sensor nodes. A sensor network reaches syn-

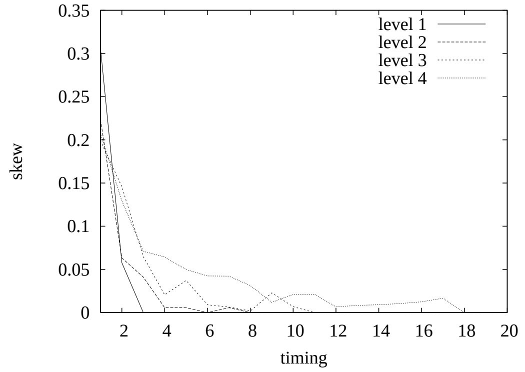

Fig. 8 Transition of timing of message emission (dynamic deployment)

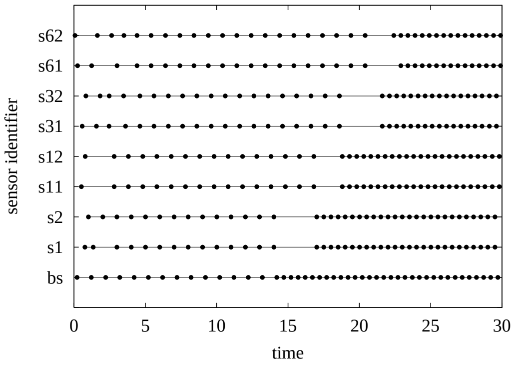

Fig. 9 Timing of message emission (changing frequency)

chronization even if new sensor nodes are deployed or sensor nodes move. A sensor network does not lose synchronization once it is attained even if sensor nodes stop due to battery depletion.

3.3 Frequency of Data Gathering

The frequency and the timing of data gathering can be controlled through adjusting the emission of beacons. The beacon dominates the synchronization. In Fig. 9, we show the course of synchronization when the base station changes the frequency of beacon emission. At 6.41, global synchronization was accomplished. At 14.6, the interval of beacon signals was reduced to half. The change propagated the sensor network to the edge and, finally, the sensor network reached global synchronization at 22.9 with the reduced frequency.

In this example, we slightly modified the scheme. Consider the case where sensor node S_i is synchronized with sensor node S_j, whose level l_j is $l_i - 1$. When sensor node S_j emits a message, ϕ_i is one when the sensor nodes are synchronized. Now, the frequency of sensor node S_j is doubled. When sensor node S_j fires, ϕ_i is only 0.5, but sensor node S_i is stimulated and ϕ_i is raised from 0.5 to 1.0. Thus, if δ_i is smaller than 0.5, sensor node S_i does not have a chance to emit a message. If δ_i is larger than 0.5, sensor node S_i emits a message later than the appropriate timing by 0.5. To overcome this problem, a sensor node adjusts its offset. When sensor node S_i becomes synchronized with sensor node S_j and maintains synchronization for several times, it changes the offset δ_i to $\delta_i \rightarrow 1.0 - \phi_i + \delta_i$. In the above example, δ_i becomes $0.5 + \delta_i$ and sensor node S_i emits its sensor information earlier than emission of sensor node S_j by δ_i as expected.

4. Further Discussions

We additionally give further considerations on our scheme from viewpoints of scalability, robustness, and energy-efficiency.

Scalability: Our scheme can be applied to sensor networks whose region is large and/or where a large number of sensor nodes are deployed, since there is no centralized control and it is highly ad hoc and self-organizing. However, as the number of sensor nodes increases, the time needed to reach global synchronization increases. Although it has been proved that "the time taken to synchronize is inversely proportional to the product ϵb" [8], we need further detailed investigation into the influence of those parameters, the number of sensor nodes, and the range of stimulus.

Robustness: By robustness, we mean that sensor information can be continuously gathered from sensor nodes at the desired rate even during the failure of some sensor nodes or unstable conditions.

When a radio transmitter fails, a sensor node cannot emit its sensor information. Before global synchronization, the broken node cannot contribute toward synchronization because it cannot stimulate other sensor nodes and thus it is not harmful to the others. After global synchronization, the failure of a radio transmitter has no influence on synchronized data gathering.

If a radio receiver fails before global synchronization, its sensor node does not become synchronized with the other sensor nodes. As a result, it continues to emit its sensor information at its own interval, independently of the others. If it has wrongly identified its level, neighboring sensor nodes that receive radio signals only from the failed sensor node are influenced and become isolated from the sensor network. Other sensor nodes can correctly determine their levels and attain global synchronization among themselves.

On the other hand, if the failed sensor node has correctly identified its level before the failure, its message emission disturbs global synchronization. A sensor node on the next level receives radio signals from both normal and failed sensor nodes at different phases. Being stimulated by those non-synchronized signals, the state and phase of the sensor node does not converge, and thus they never become synchronized. However, it is easy to solve this problem. When the failed node stops message emission or sets its level at a large enough value, it never stimulates other sensor nodes and there is no disturbance.

In some cases, a timer gains or loses, being affected by, for example, geomagnetism. A sensor node with a wrong timer regains synchronization through reception of radio signals from sensor nodes on the lower level. Sensor nodes that are stimulated by the failed sensor node vary from the global synchronization. However, since they are stimulated by other correct sensor nodes, they again reach synchronization. We should note here again that the global synchronization is kept and reestablished through mutual interactions among neighboring sensor nodes which only emit their sensor information at their own intervals as fireflies do. Our scheme is simple and energy efficient.

In an actual situation, radio communications are not stable and asymmetric. The radio signal from a sensor node does not always arrive at another node even if that node is within a range of the radio signal due to interference among radio signals and reflections and disturbances of obstacles such as walls, floors, ceilings, and human beings.

First, consider the case that a radio signal from sensor node S_i reaches sensor node S_j, where node S_i is closer to the base station than node S_j, but a signal does not reach from node S_j to node S_i. A timer of node S_j becomes synchronized with data emissions of node S_i and node S_j emits its sensor information at appropriate instants, i.e., slightly before the emission of node S_i. However, sensor information that node S_i emits does not contains the information of node S_j. Therefore, by investigating the information received from node S_i, node S_j can notice that its signal does not reach node S_i. Then, node S_j increases the transmission power so that its signal reaches node S_i.

On the other hand, in the case that node S_i can hear node S_j but node S_j does not receive any signals from node S_i, node S_j does not know the existence of node S_i. Consequently, node S_j does not establish the synchronization with node S_i. If node S_j finds another node closer to the base station, it is stimulated and attains the synchronization. Otherwise, it becomes isolated.

If a sensor node does not fall within radio range of any other sensor node, it is isolated. However, the sensor node can join the sensor network when it is moved closer to one of the other nodes. If new sensor nodes are deployed between the isolated node and the network, it can join the network through the mediation of the new nodes. Another solution is to introduce a scheme to control the transmission power. When a sensor node does not receive any radio signals for several cycles of the timer, it advertises “I cannot hear anyone” by using the stronger transmission power. Receiving the advertisement, neighboring nodes increase their transmission power. Consequently, the isolated node can be synchronized with one of neighboring nodes and join the network. We need further considerations on the effective and efficient power control scheme.

Energy-efficiency: Since our scheme can attain a global synchronization that effectively schedules the emission of sensor information, we can save power consumption by turning off unused components of a sensor node between periodic message emissions. Before global synchronization, a sensor node should keep awake to listen for radio signals of other sensor nodes and to emit a message as stimulus for others. However, after global synchronization is attained, a sensor node can move to a power-saving mode. It turns off unused components including a radio transceiver from $\phi_i = 0.0$ to $1.0 - \delta_i$. At $\phi_i = 1.0 - \delta_i$, a sensor node turns on a radio transceiver to emit a message. Then, at $\phi_i = 1.0$, it receives radio signals from sensor nodes, which it can use to confirm that it is well synchronized. Then, its phase ϕ_i returns to zero and the sensor node goes to sleep again. As a result, battery consumption can be reduced to δ_i compared to fully active operation.

However, a sensor node itself cannot detect global synchronization because it can perceive only the sensor nodes around it. Thus, we propose to start the power-saving mode when a sensor node considers it is synchronized with one or more sensor nodes whose level is smaller than its own level by one. When a sensor network has not yet reached global synchronization, the timers of the sensor nodes that a sleeping sensor node relies upon might either gain or lose. When they gain, the sensor node receives radio signals at the phase $\phi_i < 1.0$. Since it is awake, it is stimulated. When they lose, radio signals reach the sensor node while it is sleeping. Thus, it cannot accomplish synchronization. To attain synchronization again, a sensor node stops the power-saving mode when it does not receive any valid radio signals while it is awake.

5. Conclusion

In this paper, inspired by biological systems, we proposed a novel scheme for data gathering in sensor networks that is fully-distributed, self-organizing, robust, adaptive, scalable, and energy efficient. Through simulation experiments, we confirmed that our scheme could accomplish the synchronized data gathering.

We are now considering ways to make even more efficient data gathering. For example, when sensor nodes are deployed densely, there are areas that are monitored by two or more sensor nodes. To avoid a waste of energy to collect duplicated information, it is effective to organize a cluster of the sensor nodes that monitor the same area and then make one of members in the cluster report the sensor information. A phase-lock condition in a pulse-coupled oscillator model, where oscillators fire with a constant phase difference, can be adopted for this purpose. We need to consider how sensor nodes are clustered and how the stimulus should be determined in a distributed way.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Computer Networks (Elsevier) Journal, vol.38, no.4, pp.393–422, March 2002.

[2] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," Proceedings of the Hawaii International Conference on System Sciences, pp.3005–3014, January 2000.

[3] S. Lindsey, C. Raghavendra, and K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric," Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS-01), pp.2001–2008, April 2001.

[4] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering–based heuristic for data gathering and aggregation in sensor networks," Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), pp.16–20, March 2003.

[5] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communications, pp.660–670, October 2002.

[6] O. Younis and S. Fahmy, "Distributed clustering in adhoc sensor networks: A hybrid, energy-efficient approach," Proeedings of IEEE INFOCOM 2004, March 2004.

[7] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," IEEE Network, vol.18, pp.45–50, July/August 2004.

[8] R.E. Mirollo and S.H. Strogatz, "Synchronization of pulse-coupled biological oscillators," Society for Industrial and Applied Mathematics Journal on Applied Mathematics, vol.50, no.6, pp.1645–1662, December 1990.

[9] X. Guardiola, A. Díaz-Guilera, M. Llas, and C. Pérez, "Synchronization, diversity, and topology of networks of integrate and fire oscillators," The America Physical Society Physical Review E, vol.62, no.4, pp.5565–5569, October 2000.

[10] M.B.H. Rhouma and H. Frigui, "Self-organization of pulse-coupled oscillators with application to clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.23, no.2, pp.180–195, February 2001.

[11] I. Wokoma, I. Liabotis, O. Prnjat, L. Sacks, and I. Marshall, "A weakly coupled adaptive gossip protocol for application level active networks," Proceedings of EEE 3rd International Workshop on Policies for Distributed Systems and Networks (Policy 2002), pp.244–247, June 2002.

[12] D.L. Hall and J. Llinas, eds., Handbook of Multisensor Data Fusion, CRC Press, 2001.

Naoki Wakamiya received M.E. and Ph.D. from Osaka University, Japan, in 1994 and 1996, respectively. He was a Research Associate of Graduate School of Engineering Science, Osaka University from April 1996 to March 1997, a Research Associate of Educational Center for Information Processing from April 1997 to March 1999, an Assistant Professor of Graduate School of Engineering Science from April 1999 to March 2002. Since April 2002, He is an Associate Professor of Graduate School of Information Science and Technology, Osaka University. His research interests include QoS architecture for distributed multimedia communications and wireless sensor networks. He is a member of IEICE, IPSJ, ACM, and IEEE.

Masayuki MURATA received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Osaka University, and is now with Graduate School of Information Science and Technology, Osaka University since April 2004. He has more than three hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM, The Internet Society, IEICE and IPSJ.