Scalable and Robust Scheme for Data Fusion in Sensor Networks

Naoki Wakamiya[1] and Masayuki Murata[2]

[1] Graduate School of Information Science and Technology, Osaka University
Toyonaka, Osaka 560-8531, Japan
wakamiya@ist.osaka-u.ac.jp
[2] Cybermedia Center, Osaka University
Toyonaka, Osaka 560-0043, Japan
murata@cmc.osaka-u.ac.jp

Abstract. In wireless sensor networks, hundreds or thousands of microsensors are deployed in an uncontrolled way to monitor and gather information of environments. Sensor nodes have limited power, computational capacities, memory, and communication capability. In this paper, we propose a novel scheme for data fusion where sensored information periodically propagates without any centralized control from the edge of a sensor network to a base station as the propagation forms a concentric circle. By observing the radio signals emitted by sensor nodes in its vicinity, a sensor node independently determines the cycle and the timing at which it emits sensored information in synchrony. 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. Through simulation experiments, we confirmed that our scheme can fuse sensor information in a fully-distributed, self-organizing, robust, adaptable, scalable, and energy-efficient manner.

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 sensored phenomena directly or indirectly to the base station, i.e., the sink of sensored 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 pervasive computing are typical examples that benefit from information gathered from circumstances and environments. A sensor node sends its sensored information by radio signals, continuously, periodically, or only when it detects an event such as a movement of the object. Since a sensor node usually has an unidirectional antenna, broadcasting is the major means of data emission.

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 sensored information should be scalable to the number of sensor nodes, robust to the failure and disruption of sensor nodes, adaptable to addition, removal, and movement of sensor 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 fusion in sensor networks, such as [2–4]. However, they 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 sensor nodes, and the residual energy of all sensor 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.

In this paper, we propose a novel and efficient scheme for data fusion in sensor networks where a large number of sensor nodes are deployed; in such networks, nodes are randomly introduced, occasionally die or get removed, and sometimes change their locations. We consider an application that periodically collects sensored information from distributed sensor nodes to a base station. Sensored information is propagated from the edge of a sensor network to the base station. We do not assume that all sensor 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 (MAC) protocol.

In periodic data fusion, 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 fusion can be attained by the following strategy on a sensor network where sensor nodes organize a tree whose root is the base station in a distributed manner. First, leaves, i.e., sensor nodes that are the most distant from the base station, simultaneously emit their sensored information to their parent nodes at a regular interval. The parent nodes, which are closer to the base station, receive information from their children. They aggregate the received information with locally sensored information to reduce the amount of data to send. Then, they emit it at a timing that is synchronized with the other sensor nodes at the same level in the tree. Likewise, sensored 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.

To accomplish the synchronized data fusion without any centralized controls, each sensor node should independently determine the cycle and the timing at which it emits a message to advertise its sensored information based on locally available information. The ideal synchronization can be attained by configuring sensor nodes prior to the deployment, provided that the clocks of sensor nodes 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.

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 [5–7]. Each oscillator O_i has a state x_i, which is determined by a monotonically increasing function $f_i : [0, 1] \rightarrow [0, 1]$ 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 $\epsilon_i(\phi_j)$ [7]. Those oscillators whose states are raised to one also fire at this time. In this way, they reach synchronization. See section 2 for further discussion. In [7], the authors applied the pulse-coupled oscillator model to clustering algorithms. They defined the stimulus function $\epsilon_i(\phi_j)$ as a monotonically increasing function of the similarity between two objects. If they resemble each other, the stimulus has a positive value. They are synchronized and constitute a cluster. On the other hand, if they are not similar, they give inhibitory stimulus to each other. As a result, they behave non-synchronously and group themselves into different clusters. In [8], the authors proposed a management policy distribution protocol based on firefly synchronization theory. The protocol is based on gossip protocols to achieve weak consistency of information among nodes. The rate of updates is synchronized in a network through pulse-coupled interactions. They verified that their protocol is scalable to the number of nodes in terms of the average update latency. Although they attempted to distribute a management policy whereas our application is designed to collect sensored information to a base station, by adapting the pulse-coupled oscillator model, we can obtain a fully distributed, self-organizing, robust, adaptable, scalable, and energy-efficient scheme for data fusion in wireless sensor networks. By observing the signals that neighboring sensor nodes emit, each sensor node independently determines the cycle and the timing at which it emits a message to achieve synchronization with those neighboring sensors and thus draw a concentric circle.

The rest of the paper is organized as follows. First, in Section 2, we briefly introduce sensor networks and the outline of our data fusion scheme. Next, we apply a pulse-coupled oscillator model to our data fusion in Section 3. Then, we show some experimental results in Section 4. Regarding the results, Section 5 discusses some additional considerations of our scheme. Finally, we conclude the paper in Section 6.

2 Sensor Networks and Proposed Data Fusion Scheme

To collect sensored information for use by users, applications, or systems, a base station is placed at a location from which one can draw information of the region. Thus, sensor nodes must organize a network over which information sensored at all sensor nodes in the region can be successfully gathered to the base station. Since sensor nodes are usually deployed in an uncontrolled manner, they are prone to failure, they often move, and they die due to battery depletion. Thus, a scalable, robust, adaptable, low-power-consuming, fully distributed scheme is needed to organize a sensor network and collect sensored information.

The 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. 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 monitors its surroundings and obtains sensored information. A sensor node can hear radio signals from other nodes. A sensor node aggregates its locally sensored information and the information received from other sensor nodes [2–4]. 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, a sensor node emits its sensored information, possibly aggregated with that of other nodes, without waiting for the reception of sensored information from other sensor nodes. 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 protocol 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.

3 Pulse-Coupled Oscillator and Its Application to Scalable and Robust Data Fusion

In this section, we first introduce the pulse-coupled oscillator model and then give a detailed description of our proposed data fusion scheme. By synchronizing the message emissions of sensor nodes, sensored information propagates from the edge of a sensor network toward the base station at a preferred frequency.

3.1 Pulse-Coupled Oscillator

The pulse-coupled oscillator model is developed to explain the synchronous behaviors of biological oscillators such as pacemaker cells, fireflies, crickets [5]. In

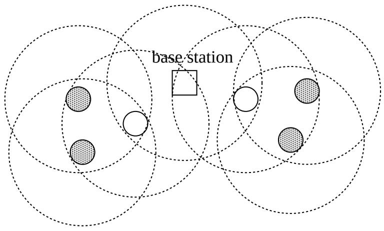

Fig. 1. An example of a sensor network

this section, mainly following the model proposed in [7], we give a brief explanation of the pulse-coupled oscillator model.

Consider a set of N oscillators, $\mathcal{O} = \{O_1, \cdots, O_N\}$. Each oscillator has phases $\phi_i \in [0, 1]$ and $x_i \in [0, 1]$, whose relation is expressed by a phase-state function f_i:

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

$f_i : [0, 1] \to [0, 1]$ is smooth and monotonically increasing. $f_i(0) = 0$ and $f_i(1) = 1$ hold.

As time passes, ϕ_i shifts toward one and, after reaching it, jumps back to zero. The periodic cycle is given as T_i, where $\frac{d\phi_i}{dt} = \frac{1}{T_i}$. When x_i reaches one, the oscillator fires and x_i is initialized to zero. Oscillators coupled with the firing oscillator are stimulated. Their state x_j is raised by an amount $\epsilon_i(\phi_j)$.

$$x_j(t^+) = B(x_j(t) + \epsilon_i(\phi_j)). \tag{2}$$

Function B is defined as,

$$B(x) = \begin{cases} x, & \text{if } \ 0 \leq x \leq 1 \\ 0, & \text{if } \ x < 0 \\ 1, & \text{if } \ x < 1 \end{cases} . \tag{3}$$

Their phase ϕ_j is given as $\phi_j = f_j^{-1}(x_j)$. When x_j reaches one, oscillator j also fires. Oscillators i and j are then regarded as synchronized.

It has been proven that oscillators with different phase-state functions and different frequencies attain synchronization from arbitrary initial conditions.

3.2 Scalable and Robust Data Fusion

First, we give a brief explanation of the basic behavior of data fusion in our scheme. Consider the network of one base station and six sensor nodes as illustrated 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 sensor 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 fusion, it is efficient in terms of power consumption that sensor nodes on the same level synchronously inform their parents of their sensored information. In addition, since each sensor node emits its sensored information at its own timing without waiting for the reception of sensored 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 $t-\delta$. For them to reflect information gathered on the higher level, all four sensor 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 fusion is attained, the radio component of a sensor node needs to be turned on only for δ out of the data fusion interval in this example. 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. 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 sensor 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 explained in Section 3.1. 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 $f_i : [0,1] \to [0,1]$ of a phase ϕ_i of the timer. For example, we used the following f_i as an example in the simulation experiments.

$$\forall i, \quad f_i(\phi_i) = \frac{1}{b}\ln[1+(e^b-1)\phi_i], \tag{4}$$

This formula is taken from [5, 7]. $b > 0$ is one of parameters that dominate the rate of synchronization [5]. 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.

$$\phi'_i = p(\phi_i, l_i\delta_i) = \begin{cases} \phi_i + l_i\delta_i, & \text{if } \phi_i + l_i\delta_i \le 1 \\ \phi_i + l_i\delta_i - 1, & \text{otherwise} \end{cases} \tag{5}$$

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

```
 0                                                             31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L L L L|E E E|F F F F F F F F F|E E E|F F F F F F F F F|       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+     +
|               Sensored Information                            |
+                                                               +
```

Fig. 2. Message header format

level l_i is smaller than S_j's level l_j. It is stimulated and its state changes as

$$x_j(t^+) = B(x_j(t) + \epsilon). \tag{6}$$

The regulated state x'_j of stimulated sensor 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 sensor S_j's regulated state x'_j becomes one, it also emits a message in synchrony with sensor 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 a sensor node S_i emits to advertise its information contains the level l_i, a stimulus ϵ with which it stimulates sensor nodes around it, δ for its children to use an identical offset, and its sensored information possibly aggregated with other sensored information. Figure 2 illustrates a message format. 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. The offset is in the order of milliseconds to seconds. If the interval of data fusion is one hour, the offset of one second is expressed as $1/(60 \times 60) = 2.78e - 4$. If it is once a day, the offset becomes $1/(24 \times 60 \times 60) = 1.16e - 5$. The single IEEE standard floating-point representation requires 32 bits, but we can use a smaller number of bits in our scheme. Of course, it also useful to employ an absolute value to express the offset. 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] 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 fusion. All sensor nodes initialize their levels to inifinity or a reasonably large value. They also initialize their timers. Each sensor node begins to sensor its surroundings and stores sensored information into its memory. When the timer expires, it emits a message to advertise its sensored 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 sensored 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 sensored 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, which is discussed in Section 5.2.

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 sensors its surroundings, emits sensored 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 sensor node will be stimulated by another that is audible. If there is no other sensor node in its vicinity, the sensor node becomes isolated.

4 Simulation Results

In this section, we show some results of simulation experiments to investigate the basic behavior and characteristics of the proposed scheme.

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 is assumed to be located at the center of the region. The range of the radio signal 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

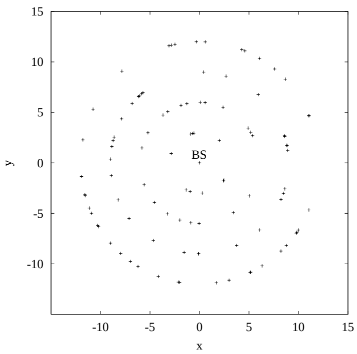

Fig. 3. An example of concentric circular sensor networks

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. 3 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 (4). In the following experiments, we used $b = 3.0$ [7]. The functions of stimulus ϵ_i are identical and defined as

$$\forall i \forall j, \quad \epsilon_i(\phi_j) = \epsilon. \tag{7}$$

We used $\epsilon = 0.3$ [5]. Finally, offset values δ_i are also identical and given as

$$\forall i, \quad \delta_i = \delta. \tag{8}$$

We used $\delta = 0.2$. 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. We should note here that no routing protocol is employed in simulation experiments. With our proposed scheme, sensored information propagates to the center of the circle without a help of routing protocols.

4.1 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

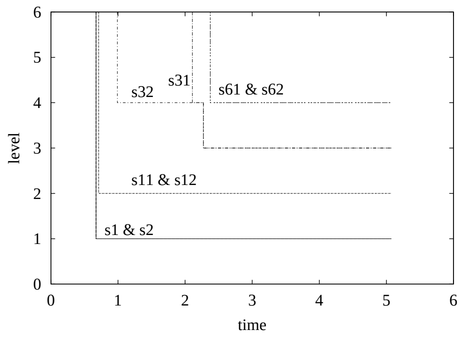

Fig. 4. Transition of level

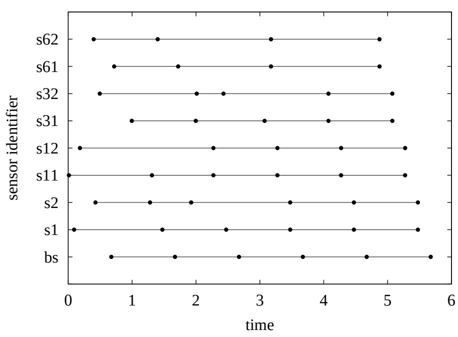

Fig. 5. Timing of message emission

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 [5]. Thus the frequency of data fusion, which is controlled by the interval of beacon signals from the base station, should be the smallest in the sensor network.

Figure 4 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, for example, larger than the number of sensor nodes. The initial value does not affect the time to reach global synchronization. When a sensor node receives a radio signal from a sensor node whose level has already been determined, it can identify its level. In the figure, sensor 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, sensor nodes s11 and s12 received radio signals from sensor nodes on the first circle at 0.712 and set their levels to two. Sensor 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 5 shows how the sensor network reaches the global synchronization. Dots on lines stand for instants when sensors emit messages. It can be seen that each sensor 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 sensor nodes on the same circle and earlier than sensor 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 sensor nodes, it can be seen that sensor nodes emit messages in synchrony

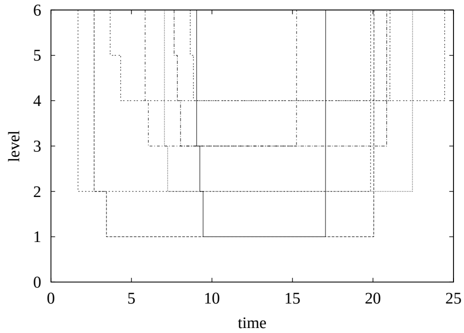

Fig. 6. Transition of level (dynamic sensor deployment)

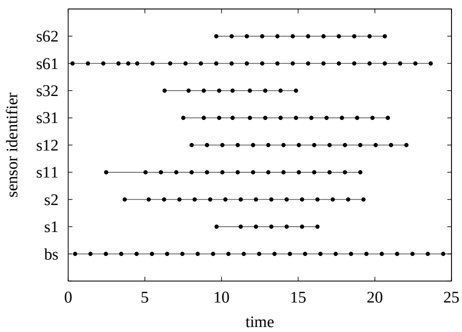

Fig. 7. Timing of message emission (dynamic sensor deployment)

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

4.2 Dynamic Deployment and Removal of Sensor Nodes

In the experiments described in the preceding section, all 100 sensor nodes were fully deployed at the initial stage. However, in an actual situation, sensor nodes are added to a sensor network as well as removed or stopped occasionally. Our scheme can reach the desired condition on such dynamic sensor networks.

The following figures were obtained from simulation experiments where sensor nodes were deployed in the region at random one by one. In addition, they also 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 initial level of a newly deployed sensor node is set to a large value. The level is gradually adapted as it encounters another sensor node through reception of radio signals, as described in Section 3.2. 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 ti 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.

Figure 7 shows a series of message emissions of sensor nodes s1, s2, s11, s12, s31, s32, s61, and s62, as in Fig. 5. 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.

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 again when it is moved closer to one of the other sensor nodes. If new sensor nodes are deployed between the isolated sensor node and the sensor network, it can join the network through the mediation of the new nodes. An isolated sensor node can find neighboring sensor nodes by extending the range of radio propagation to inform another sensor node of its existence. When another sensor node adjusts the range of its radio signal to reach the isolated sensor node, that node can join the sensor network and attain synchronization again.

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 synchronization 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.

4.3 Larger Sensor Network

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" [5], we need further detailed investigation into the influence of those parameters, the number of sensor nodes, and the range of stimulus.

4.4 Frequency of Data Fusion

The frequency and the timing of data fusion can be controlled through adjusting the emission of beacons from the base station. The beacon dominates the synchronization. In Fig. 8, 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 described in Section 3.2: Sensor node S_i emits a message at $\phi_i = 1.0 - \delta_i$. 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

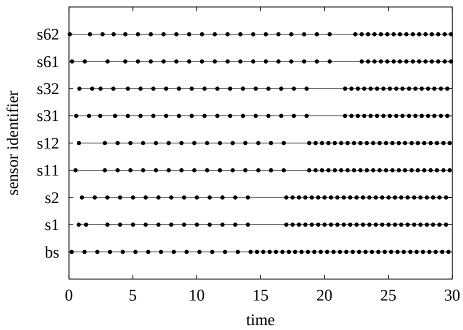

Fig. 8. Timing of message emission (changing frequency)

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 sensored information earlier than emission of sensor node S_j by δ_i as expected.

5 Further Discussions

In the preceding sections, we verified that our scheme was fully distributed and self-organizing, adaptable to dynamic changes of sensor networks, and scalable to the number of sensor nodes. In this section, we give further consideration to our data fusion scheme from the viewpoints of robustness and energy efficiency.

5.1 Robustness to Failure of Sensor Nodes

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

When a radio transmitter fails, a sensor node cannot emit its sensored information. Before global synchronization, the broken node cannot contribute toward synchronization because it cannot stimulate other sensor nodes. However, as long as all sensor nodes can find a sensor node whose level is smaller, global synchronization can be accomplished. The broken node itself can synchronize with others. Thus, its sensored information successfully reaches the base station. After global synchronization, the failure of a radio transmitter has no influence on synchronized data fusion.

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 sensored 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. Consequently, global synchronization cannot be accomplished in this case. 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.

5.2 Energy Efficiency

Since a sensor node is typically operated with a battery, which often cannot be replaced, power management plays a vital role in sensor networks. All of the components that constitute a sensor node consume battery life when they are active and idle [10, 9, 11]. In addition, a radio transceiver expends battery power in sending and receiving data. Previous research work on data fusion for sensor networks [2–4] took into account power consumption in gathering data from sensor nodes in order to prolong the lifetime of sensor networks. In this subsection, we investigate how energy-efficient data fusion can be accomplished with our proposed scheme.

Since our scheme can attain a global synchronization that effectively schedules the emission of sensored 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.

The power-saving mode introduces another problem when it is activated before global synchronization. As shown in Figs. 4 and 6, a sensor node occasionally misidentifies its level. When a sensor node reaches synchronization with a wrongly identified level, it cannot correct the level while in the power-saving mode. For example, consider the case where newly deployed sensor node S_i, whose actual level is $k-1$, accidentally receives a radio signal from sensor node S_j of level $l_j = k-1$. It wrongly considers its level to be $l_i = k$ and becomes synchronized with sensor node S_j. The power-saving mode starts because synchronization is attained. Sensor node S_i sleeps from $\phi_i = 0$ to $1.0 - \delta_i$, but it can hear a radio signal from sensor node S_j at $\phi_i = 1.0$. To correct the level, sensor node S_i must receive a radio signal from a sensor node of level $k-2$. However, because of the offset δ_j, the signal reaches sensor node S_i at $\phi_i = 1.0 + \delta_j = \delta_j$. Since δ_j must be small enough from the viewpoint of energy efficiency, $\delta_j < 1.0 - \delta_i$ usually holds. Thus, sensor node S_i cannot receive a radio signal from a sensor node of level $k-2$ and thus cannot correct its level. One possible solution to this problem is to turn on a radio transceiver around $\phi_i = \delta_j$.

6 Conclusion

In this paper, inspired by biological systems, we proposed a novel scheme for data fusion in sensor networks that is fully-distributed, self-organizing, robust, adaptable, scalable, and energy efficient. Through simulation experiments, we confirmed that our scheme provides those advantages.

We are now considering ways to make even more efficient data fusion. When sensor nodes are deployed densely, there are areas that are monitored by two or more sensor nodes. In such a case, it is a waste of energy to collect duplicated information from all of those sensor nodes. We first organize a cluster of the sensor nodes that monitor the same area and then make one of the sensor nodes in the cluster advertise the sensored information. In a pulse-coupled oscillator model, we can attain a phase-lock condition, where oscillators fire with a constant phase difference, by using negative stimulus. We need to consider how sensor nodes are clustered and how the stimulus should be determined in a distributed way.

In some cases, a variety of sensor nodes is deployed in a region, and they might have different sensing frequencies. In our scheme, all sensor nodes are synchronized, and sensored information is gathered from all sensor nodes at an interval identical to the fastest timer period among the sensors. Such global synchronization apparently wastes the battery life of infrequent sensors. The proposed scheme seems useful for overcoming this problem, but we need to further study this issue in more detail.

Acknowledgement

This research was partly supported by "The 21st Century Center of Excellence Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks (Elsevier) Journal **38** (2002) 393–422
2. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the Hawaii International Conference on System Sciences. (2000)
3. Lindsey, S., Raghavendra, C., Sivalingam, K.: Data gathering in sensor networks using the energy*delay metric. In: Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS-01). (2001)
4. Dasgupta, K., Kalpakis, K., Namjoshi, P.: An efficient clustering–based heuristic for data gathering and aggregation in sensor networks. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC). (2003)
5. Mirollo, R.E., Strogatz, S.H.: Synchronization of pulse-coupled biological oscillators. Society for Industrial and Applied Mathematics Journal on Applied Mathematics **50** (1990) 1645–1662
6. Guardiola, X., Díaz-Guilera, A., Llas, M., Pérez, C.: Synchronization, diversity, and topology of networks of integrate and fire oscillators. The America Physical Society Physical Review E **62** (2000) 5565–5569
7. Rhouma, M.B.H., Frigui, H.: Self-organization of pulse-coupled oscillators with application to clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001)
8. Wokoma, I., Liabotis, I., Prnjat, O., Sacks, L., Marshall, I.: A weakly coupled adaptive gossip protocol for application level active networks. In: Proceedings of EEE 3rd International Workshop on Policies for Distributed Systems and Networks - Policy 2002. (2002)
9. Aslam, J., Li, Q., Rus, D.: Three power-aware routing algorithms for sensor networks. Wireless Communications and Mobile Computing (2003) 187–208
10. Sinha, A., Chandrakasan, A.: Dynamic power management in wirelss sensor networks. IEEE Design & Test of Computers **18** (2001) 62–74
11. Schurgers, C., Tsiatsis, V., Ganeriwal, S., Srivastava, M.: Topology management for sensor networks: Exploiting latency and density. In: Proceedings of Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'02). (2002)