PAPER *Special Section on Mobile Multimedia Communications*

Aggregation Efficiency-Aware Greedy Incremental Tree Routing for Wireless Sensor Networks

Shinji MIKAMI[†], ***Student Member***, **Takafumi AONISHI**[††], **Hironori YOSHINO**[††*], ***Nonmembers***, **Chikara OHTA**[†††a)], ***Member***, **Hiroshi KAWAGUCHI**[†††], ***Nonmember***, ***and*** **Masahiko YOSHIMOTO**[†††], ***Member***

SUMMARY In most research work for sensor network routings, perfect aggregation has been assumed. Such an assumption might limit the application of the wireless sensor networks. We address the impact of aggregation efficiency on energy consumption in the context of GIT routing. Our questions are how the most efficient aggregation point changes according to aggregation efficiency and the extent to which energy consumption can decrease compared to the original GIT routing and opportunistic routing. To answer these questions, we analyze a two-source model, which yields results that lend insight into the impact of aggregation efficiency. Based on analytical results, we propose an improved GIT: "aggregation efficiency-aware GIT," or AGIT. We also consider a suppression scheme for exploratory messages: "hop exploratory." Our simulation results show that the AGIT routing saves the energy consumption of the data transmission compared to the original GIT routing and opportunistic routing.
key words: *sensor networks, aggregation efficiency, greedy incremental tree routing*

1. Introduction

Recent advances in micro-sensors, integrated circuit technology and low-power wireless communications will enable the deployment of extremely small, low-cost sensor nodes with remarkable computation capability. Applications of sensor networks comprising numerous such sensor nodes include remote environmental monitoring, smart spaces, military surveillance, precision agriculture, and so on [2].

Sensor networks are expected to operate under severe energy constraints because it is not practical to replace their batteries because of the large number of sensor nodes. A salient issue is reduction of the amount of transmitted data because wireless communications at sensor nodes consume more power than any other activity [12]–[16].

Data centric routing is a promising paradigm for sensor network routing [14]. With data centric routing, routing decisions are based on the contents of the payloads of packets rather than their destination addresses. A sensor node might aggregate receiving packets that are temporally buffered, generate a new packet, and then send it to the next hop. Such a means of operation is expected to reduce the amount of transmitted data, engendering remarkable power savings. An example of data centric routing is directed diffusion (DD) [11].

An aggregation scheme should be chosen carefully according to applications. Data aggregation can be categorized into two classes: lossy and lossless [1]. Perfect aggregation [12] and beam-forming [17] are lossy aggregations. With perfect aggregation, a sensor node aggregates received data into one unit of data and then sends it to the next hop, where average, maximum, and count operations are examples of perfect aggregation functions [18]. Such an operation can remarkably reduce the amount of transmitted data. Perfect aggregation is quite efficient in this sense, whereas available applications are limited. Examples of lossless aggregations are linear aggregation [12] and coding by order [16]. Linear aggregation performs a simple operation: header elimination. A sensor node concatenates the payloads of buffered packets whose next-hops are equal and then puts it into one packet. The efficiency of the linear aggregation is lower than that of perfect aggregation, whereas lossless aggregation is versatile for all applications. The linear aggregation leads to maximum energy consumption in the class of lossless aggregation. More complicate aggregation like coding by order to achieve higher compression may require more power consumption in processor. Consideration about this aspect is left as a future issue.

In most studies, perfect aggregation has been assumed (e.g. [3], [7], [12], [13]). However, perfect aggregation is not universal and possibly limits applications of sensor networks, as mentioned above. Unfortunately, we do not have sufficient insight into the influence of the diversity of the aggregation to sensor network routings.

In the case of perfect aggregation, the most efficient data path from sources to a sink forms a Steiner tree or a minimal spanning tree on hop-count basis. This fact encourages research of heuristic distributed algorithms such as Greedy Incremental Tree (GIT) [12] for Steiner tree and the Nearest Neighbor Tree (NNT) [13] for minimal spanning tree. In the former case, some sources are assumed to send sensing data to a sink. On the other hand, all nodes are assumed to be sources in the latter case. In this paper, we focus on the former case, and the latter case is left as a future issue.

As mentioned in [14], the task to form a data path with optimal data aggregation in the case where some sources

Manuscript received December 21, 2005.
Manuscript revised April 8, 2006.
[†]The author is with the Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa-shi, 920-1192 Japan.
[††]The authors are with the Graduate School of Science and Technology, Kobe University, Kobe-shi, 657-8501 Japan.
[†††]The authors are with the Faculty of Engineering, Kobe University, Kobe-shi, 657-8501 Japan.
[*]Presently, with the Ehime Prefectural Government.
a) E-mail: c-ohta@cs.kobe-u.ac.jp
DOI: 10.1093/ietcom/e89–b.10.2741

Copyright © 2006 The Institute of Electronics, Information and Communication Engineers

send sensing data to a sink is NP-hard. This is because the minimum Steiner problem is NP-complete. The GIT is a well-known approximation algorithm for this problem, and the GIT construction runs in polynomial time with respect to the number of nodes.

Our questions are how the most efficient incremental aggregation point changes according to aggregation efficiency in the context of the GIT routing and how much energy consumption can decrease. In order to answer these fundamental questions, we analyze a simple two-source model.

Based on results of our analysis, we improve the GIT routing algorithm to find a more efficient aggregation point according to aggregation efficiency. In this paper, we call the improved GIT "aggregation efficiency-aware GIT (AGIT)."

This paper is organized as follows: Section 2 describes the original GIT algorithm. In Sect. 3, we analyze a simple two-source model to investigate the impact of the efficiency of aggregation on energy consumption. In Sect. 4, we propose AGIT routing. Section 5 shows some simulation results. Finally, we present conclusions in Sect. 6.

2. Greedy Incremental Tree

2.1 Directed Diffusion

GIT routing is based on directed diffusion (DD), which is a typical data-centric routings for sensor networks. Before describing GIT routing, we briefly explain DD (See Fig. 1).

In DD, a task described as a list of attribute-value pairs is flooded into a network as an *interest*. Through the interest diffusion process, a sensor node receives the interest sets (or updates) a *gradient* toward the neighbor which sends the interest, and resends the interest to some subset of its neighbors (or broadcast) if it is different from the previously received one. A sensor node to take the task described in the interest sends an *exploratory* message to each neighbor to whom a gradient is set. Intermediate nodes relay the exploratory message toward the sink along gradients of the interest to match the task of the exploratory message. Because the sink possibly receives multiple exploratory messages originating at a source from its neighbors, it reinforces a preferable path by sending reinforcing messages to particular ones among the neighbors from which it received an exploratory message. Intermediate nodes receiving this reinforcing message treat it similarly, so that it is relayed in the reverse direction on the path. As a result, a data path is established from the source to the sink. Refer to [11] for more detail on DD.

2.2 Finding of Aggregation Point in GIT Routing

GIT routing is a heuristic distributed algorithm to construct a Steiner tree on hop-count basis, and assumes perfect aggregation. Each source, one by one, tries to find the shortest hop from itself to the existing path tree or the sink. Figure 2 shows this process as an example. In this figure, the path from the source 1 to the sink has already been established, and the source 2 is following.

To realize this process, each exploratory message in GIT routing involves an additional attribute E, which denotes the additional cost (hop-count) from the source originating itself to the current node. The value of E is set to zero initially. Whenever resending an exploratory message, the nodes increment the value of E by one. The exploratory message is distributed through the network according to the gradient of the corresponding interest; it will arrive at nodes on the existing path tree. Consequently, the nodes on the existing path tree can know the hop-count from the source that initiated the exploratory message. In Fig. 2(a), E denotes hop-count from the source 2.

Each source involved in the existing path tree initiates an incremental cost message whenever it receives a previously unseen exploratory message that was initiated by other sources. The incremental cost message conveys two addi-

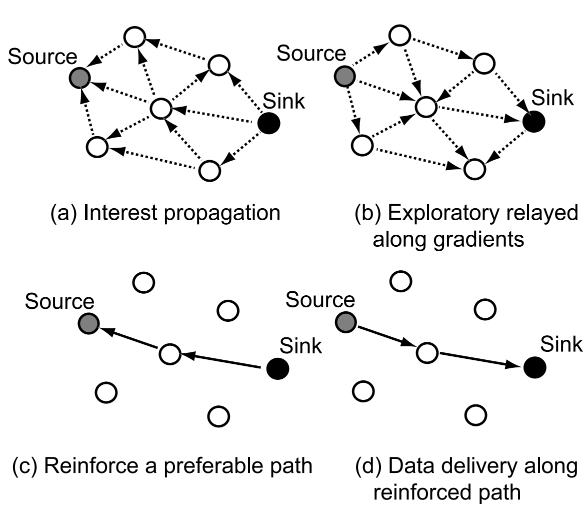

Fig. 1 A simplified schematic for directed diffusion.

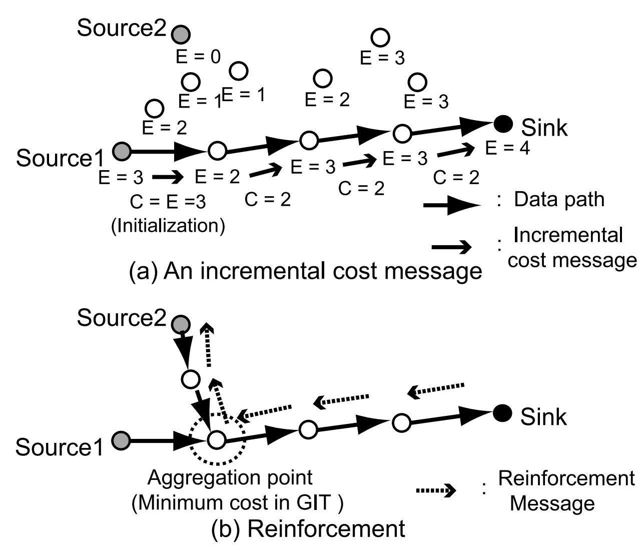

Fig. 2 An example of path establishment in GIT routing.

tional attributes: the random identifier of its corresponding exploratory message and the cost (hop-count) from the additional source (which initiated the exploratory message) to the existing path. The incremental cost message is relayed on the existing path from its originating source to the sink. The intermediate nodes update if the value of C in the incremental cost message is greater than or equal to the cached value of E. Thus C indicates the minimum value of E on the existing path. In Fig. 2(a), the source 1 initiates incremental cost message after receiving the exploratory message, and initially sets C to its $E = 3$. While traveling from the source 1 to the sink, the incremental message updates its value of C to the minimum value of E on the existing path.

The sink waits directly and late-arriving exploratory messages and other incremental cost messages for the predefined interval immediately after the arrival of the first incremental cost message. Then, the sink reinforces a neighbor, which sends an exploratory message or an incremental cost message with a lower additional energy cost C or E, respectively. In the case where an incremental cost message has the lowest additional energy cost C, the reinforcing message containing the value of C travels toward to the initiator of the incremental cost message on the existing established path until it encounters an intermediate node with $E = C(\min(E))$. This intermediate node becomes the aggregation point for the additional source that initiated the exploratory message. Then, the reinforcing message is diverted to the additional source. In Fig. 2(b), a reinforcement message travels on the existing path in the opposite direction of the incremental message, and is redirected to the source 2 at the aggregation point where E is minimal on the existing path. Thus the additional path from the source 2 to the existing path is established.

As a result of the procedure described above, the lowest cost (minimal hop-count) branch is added to the existing path tree. Refer to [12] for more details regarding GIT routing.

2.3 Discussion

In the case of perfect aggregation, the energy consumption for data transmission on the newly added branch can be regarded as the net increase of that on the entire data path. This fact, however, is not always true in different aggregation schemes.

Let us consider the case where a packet has size $L_{\text{packet}} = L_{\text{header}} + L_{\text{payload}}$, where L_{header} is the header length and L_{payload} is the payload length in bytes and N packets are incoming and one packet is outgoing at an aggregation point. In the case of the perfect aggregation, the outgoing packet after aggregation has the same size $L_{\text{header}} + L_{\text{payload}}$ as an incoming packet. On the other hand, in the case of linear aggregation, the outgoing packet has larger size $L_{\text{header}} + N \times L_{\text{payload}}$ ($N > 1$) than that of perfect aggregation. Both aggregations are simple enough for power consumption of processing to be neglected. Therefore, we assume that the energy consumption for packet transmission and reception is dominant. Generally speaking, a longer packet consumes more energy to transmit and receive it. As with [5], [6], [13], in this paper, it is assumed that the energy consumption for transmission and reception of a packet is proportional to length of the packet.

Consequently, the linear aggregation consumes more energy on the path from the aggregation point to the sink than the perfect aggregation, since total transmission data size increases. Thereby implying that a node near to the sink on the path tree might be more efficient as the aggregation point. But, how much energy can be saved if we choose the aggregation point more carefully? We analyze a simple two-source model in the next section to estimate the possible improvement.

3. Analysis of a Two-Source Model

This section shows the analysis for the simple two-source model. We also have the result for the three-source model, but we omit it because of space limitations. The analyses shown in this section are limited to GIT-like routing in which each source, one by one, tries to find the path to the existing path tree or the sink.

3.1 Model Description

Figure 3 shows the two-source model that we analyze. In this model, we assume that a lot of nodes exist densely. In this figure, however, we show only two source nodes, sink and aggregation node and do not depict another nodes for simplicity. The aggregation point is denoted as "p" in the figure. We also assume that transmission radius is constant. From the above assumptions, the hop-count between two nodes can be proportional to Euclidean distance between them. In this model, we assume that the distance between a source and the sink is equal to one unit length for simplicity. We also assume that the energy consumption to transfer a data packet per hop is proportional to the packet size. Furthermore, we assume that the path between the first source and the sink is an existing path and that the second source is going to establish the path. Note that, in the case of the original GIT routing, the second source will have a perpendicular line as the additional path to the existing path.

Here we introduce the following notations: Let x and y respectively denote the distances between the aggregation point and the sink and the distance between source 2 to the aggregation point, $(0 \leq x \leq 1)$. We denote by θ the angle

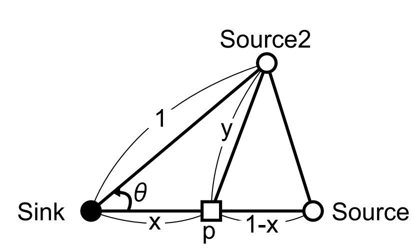

Fig. 3 Two-source model.

between source 1 to source 2, as seen from the sink ($0° \le \theta \le 90°$). Let r denote the aggregation ratio of the size of the aggregated packet to the total size of the original packets ($\frac{1}{2} \le r \le 1$). In the case of the perfect aggregation, the value r of the aggregation ratio is equal to $\frac{1}{2}$. Let E denote the energy consumed to transfer the data packets from the sources to the sink on the path tree.

3.2 Optimal Aggregation Point and Energy Consumption

The above assumptions suggest the following relationship:

$$E \propto 2rx + (1 - x) + y, \tag{1}$$

where

$$y = \sqrt{x^2 - 2x\cos\theta + 1}. \tag{2}$$

By performing some algebra for $\frac{dE}{dx} = 0$, we obtain the optimal value x_{optimum} to minimize the energy consumption for data packet transmission on the path tree:

$$x_{\text{optimum}} = \begin{cases} 0, & \frac{\cos\theta+1}{2} \le r < 1, \\ \cos\theta - \sin\theta\sqrt{\frac{1}{4r(1-r)} - 1}, & \frac{1}{2} \le r < \frac{\cos\theta+1}{2}. \end{cases} \tag{3}$$

By substituting $x = x_{\text{optimum}}$ in (1), we have the scaled value E_{optimum} of the energy consumption for data packet transmission on the path tree in the case of the optimal aggregation point. We have the value E_{GIT} expected for GIT routing, as

$$E_{\text{GIT}} \propto 2r\cos\theta + (1 - \cos\theta) + \sin\theta. \tag{4}$$

Since the aggregation point becomes nearer to the sink, the path tree will become similar to that of "opportunistic routing" [12], where data from different sources can be opportunistically aggregated at intermediate nodes along the established paths. In the case of $x = 0, y = 1$, we have the value E_{opp} expected for opportunistic routing, as

$$E_{\text{opp}} \propto 2. \tag{5}$$

To evaluate how much the optimal aggregation point saves energy compared to the original GIT routing and opportunistic routing, we introduce the following metric, "*gain*," G:

$$G(r, \theta) = \frac{\min(E_{\text{GIT}}, E_{\text{opp}}) - E_{\text{optimum}}}{\min(E_{\text{GIT}}, E_{\text{opp}})} \times 100. \tag{6}$$

3.3 Numerical Results

Figure 4 shows how the aggregation point changes according to the values of the aggregation ratio and the angle between the first and second sources. Figure 5 shows how much gain can be achieved.

From Fig. 4, we can see that the optimal aggregation point changes widely according to the value of r as the angle becomes narrower. Furthermore, the aggregation point becomes nearer to the sink compared to the foot of perpendicular from the additional source to the existing path in the case of $\frac{1}{2} < r$.

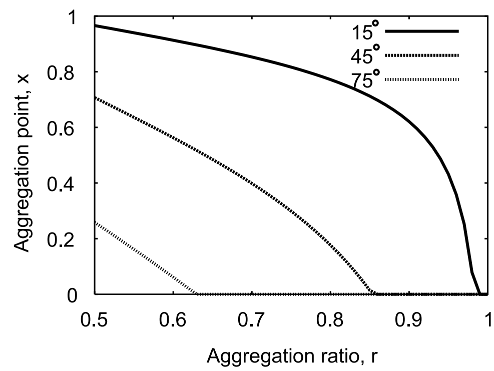

Fig. 4 Optimal aggregation point in two-source model.

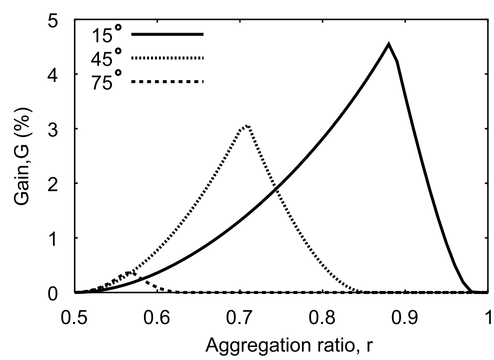

Fig. 5 Gain by optimal aggregation point in a two-source model.

Figure 5 shows that the value of gain has a peak in the middle region of r, and the larger the peak value is (up to 4.5% for 15°), the smaller the angle is. The value r to give the peak gain increases as the angle decreases. That is, the AGIT routing is more effective in the case where sources exist near and the aggregation efficiency is not so high. The value of gain converges to zero toward to the both ends. This is because the path tree becomes similar to that of the GIT routing for $r = 0.5$ and that of the opportunistic routing for $r = 1$.

Although we do not show the results of the three-source model, more gain is obtained compared to the two-source model.

4. Aggregation Efficiency-Aware GIT

In this section, we propose "aggregation efficiency-aware GIT (AGIT)" routing in order to find a more efficient aggregation point to reduce the energy consumption inherent in transmitting data packets.

4.1 Suppression of Exploratory Messages

In the DD, which is the basis of GIT routing, exploratory messages are distributed widely according to the nodes' gradients because interests do not contain any information about a sink. As a result, the gradients are set in many directions. (See Sect. 2.2.2 in [11].)

To some extent, GIT-like routing necessarily distributes exploratory messages in order to determine the aggregation point for the existing path tree. Results of our analysis showed, however, that the aggregation point becomes nearer to the sink than the foot of the perpendicular from the additional source to the existing path in the case of $\frac{1}{2} < r$.

In the AGIT routing, we consider the following scheme to suppress the excessive exploratory messages: "hop exploratory." In the following, we assume that each node can know the hop-count from the sink through interest dissemination. Each node caches the hop-count from the sink for each interest as "*own_hop*." To do so, we also assume that each interest has a random identifier to be distinguished from the others.

4.1.1 Hop Exploratory

Each exploratory message contains the additional field '*previous_hop*" to store the value *own_hop* of its sender's. In addition, each exploratory message also contains the field "*hop*" to store the hop count from the source that initiated the exploratory message. Whenever a source initiates the exploratory message with both *previous_hop* and *hop* set to *own_hop*.

When the node receives the exploratory message with *previous_hop*, it rebroadcasts the exploratory messages with *previous_hop* set to *own_hop* and with *hop* decremented by one if

$$own\_hop \le previous\_hop \quad \text{and} \quad hop > 0. \tag{7}$$

Figure 7 shows the phenomenon of dissemination of the exploratory messages, where an arrow denotes the direction in which an exploratory message is sent. From this figure, we can see that this scheme prevents network-wide diffusion compared to traditional scheme in Fig. 6, which indicates the dissemination of exploratory messages using original scheme described in [11].

4.2 Adjustment of the Incremental Cost Message Phase

The above suppression scheme involves some adjustments of the incremental cost message phase because the source nodes on the existing path tree might not receive the exploratory messages. Consequently, the incremental cost message is issued in such a case.

We take the following approach to overcome this problem. The intermediate nodes aside from the sources on the existing path tree can initiate the incremental cost message. In order to suppress the multiple incremental cost message, the more distant intermediate node from the sink issues the incremental cost message earlier. To do so, each intermediate node sets up an incremental cost message timer as

$$t_i = (max\_hop - own\_hop) \times \delta, \tag{8}$$

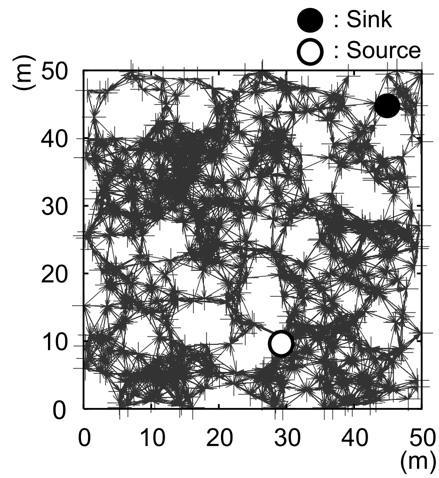

Fig. 6 Phenomenon of dissemination of exploratory messages in the traditional approach.

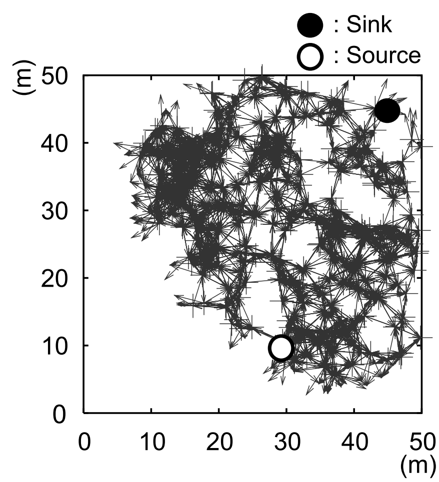

Fig. 7 Phenomenon of dissemination of exploratory messages in the hop approach.

where *max_hop* and δ respectively denote the predefined network diameter and the timer granularity. The intermediate node issues the exploratory message if its timer expires before receiving another exploratory message; otherwise it suspends the issue.

4.3 Finding of Optimal Aggregation Point

In the following, we assume that linear aggregation is employed, whereby a packet has size $L_{\text{packet}} = L_{\text{header}} + L_{\text{payload}}$ where L_{header} is the header length and L_{payload} is the payload length in bytes. Furthermore, we assume that all sources send the data packet at the same rate. The procedure shown here can be extended easily to function in different cases.

In the AGIT routing, the incremental cost message contains an additional field to store the hop-count H from an interim aggregation point. Whenever the source and/or the intermediate nodes issue a new incremental cost message, they set $H = 1$.

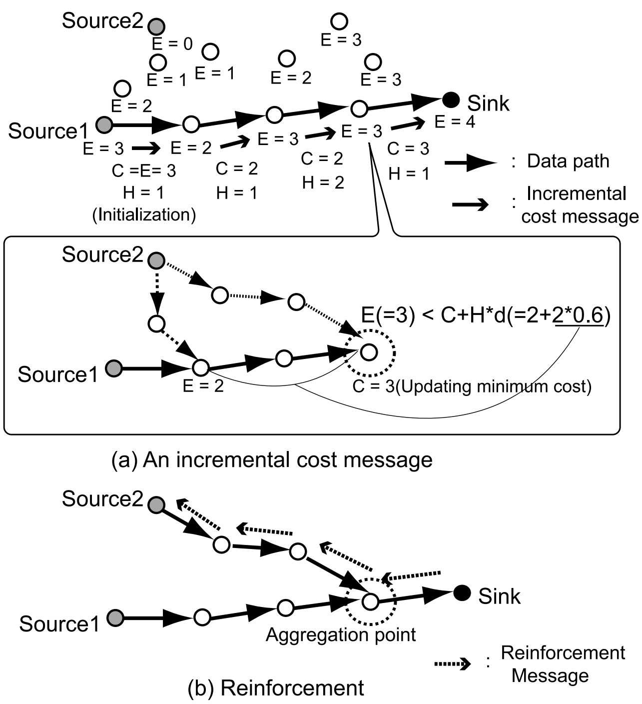

Fig. 8 An example of path establishment in AGIT.

The intermediate nodes receiving the incremental cost message execute the following:

$$\begin{array}{lll} \text{if } (E \leq C + H \cdot d) & C = E, & H = 1, \\ \text{else} & H = H + 1, & \end{array} \tag{9}$$

where $d = L_{\text{payload}}/L_{\text{packet}}$. Recall that E denotes the additional cost (hop-count) from the source joining to the existing path tree to the current node.

In (9), $C+H\cdot d$ represents the net increase of power consumption from the source nodes to the current node when using the current interim aggregation point. If this value is greater than or equal to the value of E, the current node is more optimal than the interim aggregation point. In such a case, the current node substitutes for the interim aggregation point, so that it sets $C = E$ and $H = 1$. Otherwise, it just increments the value of H by one. This manner is repeated until the incremental cost message arrives at the sink.

Figure 8 shows the search procedure of the optimal aggregation point. Here we assume that the packet length is one and the payload length is 0.6. The first source receives the exploratory message from the second source, sets $C = E = 3$ and $H = 1$ in a newly generated incremental cost message, and then forwards it the neighbor node on the path from the first source and the sink. The neighbor receiving it compares its value of E and the value of $C + H \cdot d$, where $E = 2$, $C = 2$, $H = 1$ and $d = 0.6$. The receiving node sets $C = E$ (like the original GIT routing) and $H = 1$ since its $E \leq C + H \cdot d$, and then forwards the incremental cost message to the next neighbor node on the existing path to the sink. The next simply increments H, i.e. $H = 1 + 1$, since its $E > C + H \cdot d$, where $E = 3$, $C = 2$ and $d = 0.6$, and forwards it to its next one on the existing path. Further, the next receiver sets $C = E$ and $H = 1$ since its $E \leq C + H \cdot d$, where $E = 3$, $C = 2$, $H = 2$ and $d = 0.6$. Finally, the sink receives the incremental cost message with $C = 3$ and $H = 1$. The sink finds that aggregation point is far from it by $H = 1$ hop since its $E > C + H \cdot d$. After that, the reinforcing message containing $C = 3$ is injected by the sink toward to the initiator of the incremental cost message on the existing established path until it encounters an intermediate node with $E = C$. The reinforcing message is redirected to the source 2 at the intermediate node. This manner is similar to the original GIT routing.

The overhead of AGIT routing compared to the original GIT routing merely comprises the hop-count field to store H; it can be negligible. Its procedure to find aggregation point is similar to that of the original GIT routing. Therefore, latency of the AGIT routing is expected to be almost same as that of GIT routing. We verified this expectation by simulation, but omit the results due to space limitation.

5. Simulation

In this section, we briefly explain our simulation conditions; then we show some simulation results. The aim of the simulation experiments is to confirm the effectiveness of the AGIT routing in more complicated situations.

5.1 Model and Assumption

We implemented the original GIT routing, opportunistic routing and the AGIT routing on a self-developed event-driven simulator engine.

In this simulator, 500 sensor nodes are deployed uniformly in a $50 \times 50\,\text{m}^2$ field. The transmission range is 5 m. One sink is located at (45, 45) of the two-dimensional coordinate. The number of sources is varied from two to nine; they are randomly chosen among 500 sensor nodes.

The packet has a 36-byte header. The payload length is varied as 4, 36, 108, and 216 bytes.

We implemented two schemes of the dissemination of the exploratory messages: "traditional exploratory" and "hop exploratory."

To evaluate only the effect of adjustment of aggregation point, we implemented the ideal media access control (MAC), which enables simultaneously transmitted packets to be received correctly without any collision. Further it suppresses overhearing at all. Unfortunately, such the ideal MAC can not be realized in practice. However, if we use wake-up radio [8] which has small overhead of collision and overhearing, we expect to have similar result.

Assuming the case by which the pass loss coefficient of $n = 2$, we modeled the energy consumption for transmission and reception of the packet of length l bits with distance R m, E_{tx} and E_{rx}, as follows:

$$E_{\text{tx}} = (\alpha_{\text{tx}} + \beta \cdot R^2) \cdot l, \tag{10}$$

$$E_{\text{rx}} = \alpha_{\text{rx}} \cdot l, \tag{11}$$

where α_{tx} and α_{rx} respectively denote the energy consumptions of the transmission circuit and the reception circuit, expressed as nanojoules per bit, and β denotes the radiation energy in appropriate units (nJ/bit/m^2) [5].

In simulation experiments, we use $\alpha_{tx} = 50$ nJ/bit, $\alpha_{rx} = 300$ nJ/bit, and $\beta = 1.6$ nJ/bit/m^2. We assumed $\delta = 0.1$ which is experimentally determined so that unnecessary incremental cost messages are suppressed. Channel rate is set to 19.2 kbps (after the specification of Mote [4]) and a packet is generated by a source every 10 seconds.

In each case, 50 simulation trials are executed. In Figs. 8, 9 and 11, we will plot out the average value of them.

5.2 Simulation Results

Figures 9 and 10 show the characteristics of gain defined in Sect. 3 as a function of the number of sources for different payload lengths. In this paper, we employ the ideal MAC which can suppress overhearing. Energy consumption caused by one transmission composes that by its transmitter and that by the intended receiver. Therefore, the energy consumption on the path depends on the absolute value of $E_{tx} + E_{rx}$ regardless the ratio of E_{tx} to E_{rx}. Therefore, gain is an universal measure to expresses the energy saving of AGIT routing since it is independent of so much as the absolute value of energy.

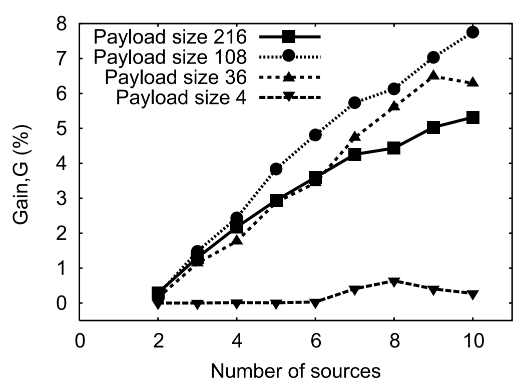

Fig. 9 Gains by nodes on the path tree in the case of the traditional exploratory scheme.

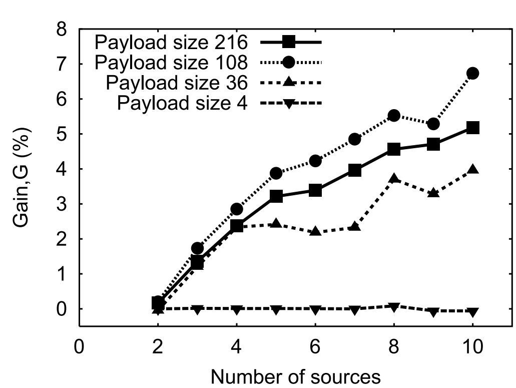

Fig. 10 Gains by nodes on the path tree in the case of the hop exploratory scheme.

Figure 9 shows the results of the traditional exploratory scheme. In this case, the exploratory messages are distributed network-wide. In the case of the small payload, 4 bytes, the gain is quite low because the aggregation ratio of the linear aggregation is almost identical that of the perfect aggregation. However, in the case of the medium payload, 36 bytes, which is the same as the header, the gain increases concomitant with the number of sources. This tendency is more remarkable in the case of the large payload, 108 bytes. However, in the case of too large payload, 216 bytes, the path tree will become similar to that of the opportunistic routing. Therefore, the gains decrease. These results coincide with predictions by our analysis shown in Sect. 3.

Figure 10 shows results of the hop exploratory scheme. From this figure, we can see that the AGIT routing is still more efficient than GIT routing and opportunistic routing, but the values of gain are decreased in comparison to those of the traditional exploratory scheme. This degradation can be explained as follows. The number of candidate aggregation points decrease in the hop exploratory scheme. This fact possibly shifts position of an aggregation point finally chosen to the sink in the GIT routing compared to the traditional exploratory scheme. However, the AGIT routing is hardly impacted since it tends to find aggregation points near to the sink compared to the original GIT routing. Therefore, in the hop exploratory scheme, the path tree constructed by the original GIT routing resembles that by the AGIT routing compared to the traditional exploratory scheme. This is reason of the above result.

The gain values are smaller than the expected values obtained from analysis. For analysis, we assume a dense network. However, in the simulation, the nodes are deployed in a discrete fashion. For that reason, the range of choices for the optimal aggregation point in the simulation is smaller than that for the analysis.

In Figs. 9 and 10, the gain fluctuates especially for the cases of 36-byte and 108-byte payloads, even though the average value for each case is led from 50 trials. This is because the variance of gain in this case is relatively large compared to the other cases. In Appendix, we will explain the dependency of the variance of gain on the aggregation ratio.

Figures 9 and 10 show the gain in the data transmission phase. From this viewpoint, the traditional exploration is preferable. However, it consumes more energy to construct the path tree. Therefore, we investigate the amount of the energy in the path setup phase. Figure 11 shows the total energy consumption of the entire network before completion of the path tree construction. This figure indicates that the traditional exploratory scheme has more overhead to construct the path tree than that the hop exploratory scheme. This overhead comes from excess flood of exploratory messages. Thus a trade-off exists between the gain of the data transmission phase and the overhead of the path tree construction phase. The answer to the problem depends on the applications: more precisely, it depends on how long the data transmission phase lasts.

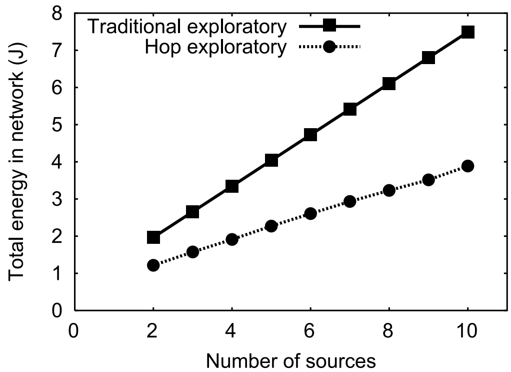

Fig. 11 Total energy consumption in whole network until the path tree is constructed.

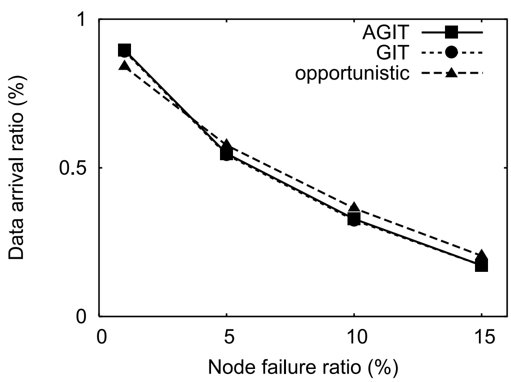

Fig. 12 Impact of node failure in the case of the traditional exploratory scheme (payload size = 4 bytes, number of sources = 5).

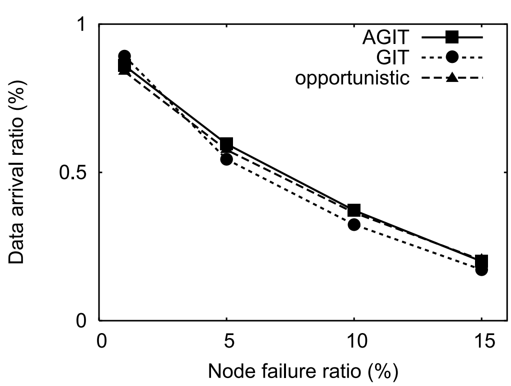

Fig. 13 Impact of node failure in the case of the traditional exploratory scheme (payload size = 216 bytes, number of sources = 5).

Figures 12–14 show robustness of opportunistic, GIT and AGIT routing with the traditional exploratory scheme. We also have similar trends for the hop exploratory scheme, but omit due to space limitation. We define the data arrival ratio as the ratio of the number of data received by the sink to that of data generated by all sources. In this simulations, it is assumed that some nodes fail after path tree construction phase.

Figures 12 and 13 show the characteristics of data arrival ratio when the number of sources is five. From these figures, we notice the following. The AGIT routing has similar characteristics to the GIT routing and the opportunistic routing in the case of shorter (4-byte) payload and longer (216-byte) payload, respectively. This is because, in the case of shorter (4-byte) payload, the path tree constructed by the AGIT routing resembles that by the GIT routing. Otherwise it resembles that by the opportunistic routing. Figure 14 show the characteristics of data arrival ratio in the case where 10 sources exist. From Figs. 12 and 14, we notice the following. The opportunistic routing is superior to the GIT routing in the case of 10 sources while the former is inferior to the latter in the case of five sources. Note that the AGIT routing has the data arrival ratio between both extremes. This is because the opportunistic routing has different paths from sources to the sink while the GIT routing shares the path among the sources.

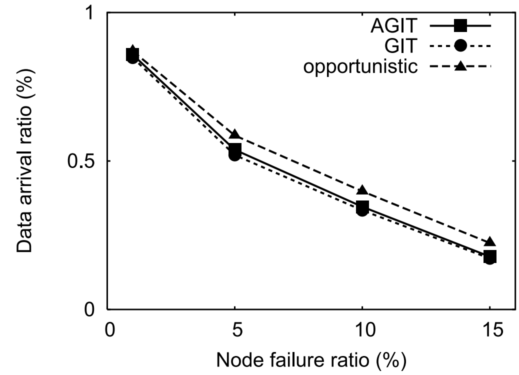

Fig. 14 Impact of node failure in the case of the traditional exploratory scheme (payload size = 4 bytes, number of sources = 10).

6. Conclusions

This paper presented the aggregation efficiency-aware GIT (AGIT) routing, and also described analyses incorporating the suppression scheme for exploratory messages: hop exploratory.

The AGIT routing can construct a more efficient path tree than the original GIT routing and the opportunistic routing. The improvement becomes more remarkable as the payload packet length becomes larger and/or more sources exist. Our simulation results demonstrate that the AGIT routing achieves about 8% of the gain for the energy consumption of the data transmission compared to the original GIT routing. However, our simulation results also emphasize that the suppression scheme, hop exploratory, reduces energy consumption up to 40%.

Acknowledgment

This research work was partially supported by a Grant-in-Aid for Young Scientists (B), No. 16700066, 2005, from the Ministry of Education, Culture, Sports, Science and Technology, Japan, and by the Kayamori Foundation for Advancement of Information Science.

References

[1] T.F. Abdelzaher, T. He, and J.A. Stankovic, "Feedback control of data aggregation in sensor networks," IEEE Conference on Decision and Control, Dec. 2004.
[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Comput. Netw. J., vol.38, pp.393–422, March 2002.
[3] K. Akkaya, M. Younis, and M. Youssef, "Efficient aggregation of delay-constrained data in wireless sensor networks," Proc. Internet Compatible QoS in Ad Hoc Wireless Networks 2005, Jan. 2005.
[4] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance measurements of motes sensor networks," Proc. ACM MSWiM, pp.174–181, Oct. 2004.
[5] M. Bhardwaj, T. Garnett, and A.P. Chandrakasan, "Upper bounds on the lifetime of sensor networks," Proc. ICC, pp.785–790, June 2001.
[6] A. Depedri, A. Zanella, and R. Verdone, "An energy efficient protocol for wireless sensor networks," Proc. AINS 2003, June 2003.
[7] M. Enachescu, A. Goel, R. Govindan, and R. Motwani, "Scale free aggregation in sensor networks," Proc. First International Workshop on Algorithmic Aspects of Wireless Sensor Networks, pp.71–84, July 2004.
[8] C. Guo, L.C. Zhong, and J.M. Rabaey, "Low power distributed MAC for ad hoc sensor radio networks," Proc. Globecom, vol.5, pp.2944–2948, 2001.
[9] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," Proc. ACM Symposium on Operating Systems Principles, Oct. 2001.
[10] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," IEEE Micro, vol.22, no.6, pp.12–24, Nov. 2002.
[11] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," IEEE/ACM Trans. Netw., vol.11, no.1, pp.2–16, Feb. 2003.
[12] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of density on data aggregation in wireless sensor networks," Proc. 22nd International Conference on Distributed Computing Systems, Nov. 2001.
[13] M. Khan, G. Pandurangan, and B. Bhargava, "Energy-efficient routing schemes for wireless sensor networks," Tech. Rep. of Department of Computer Science, Purdue University, CSD TR 03-013, July 2003.
[14] B. Krishnamachari, D. Estrinf, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," IEEE INFOCOM, June 2002.
[15] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," Proc. 22nd International Conference on Distributed Computing Systems, July 2002.
[16] D. Petrovic, C. Shah, K. Ramchandran, and J. Rabaey, "Data funneling: Routing with aggregation and compression for wireless sensor networks," Proc. IEEE Sensor Network Protocols Applications, Anchorage, AK, May 2003.
[17] A. Wang, W.B. Heinzelman, A. Sinha, and A.P. Chandrakasan, "Energy-scalable for battery-operated microsensor networks," J. VLSI Signal Process., vol.29, pp.223–237, Nov. 2001.
[18] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," Proc. IEEE International Workshop on Sensor Network Protocols and Applications, May 2003.

Appendix: Variance of Gain

Figures A· 1 and A· 2 show the samples of path tree of GIT routing and AGIT routing, respectively, in the case where

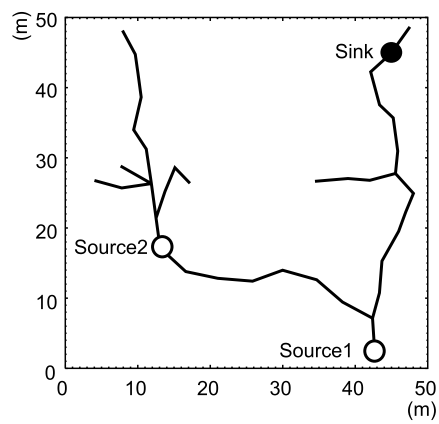

Fig. A· 1 Path tree of the GIT routing.

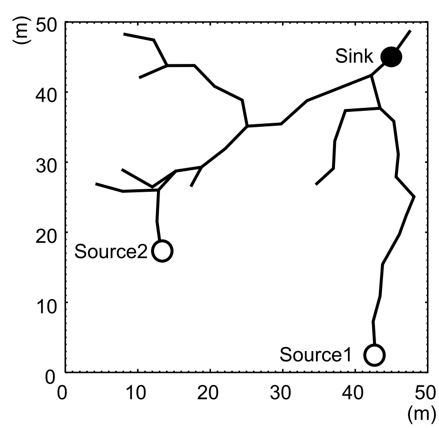

Fig. A· 2 Path tree of the AGIT routing.

the gain of AGIT routing to GIT routing is relatively large on condition of 36-byte payload (the aggregation ratio is 0.75). In both figures, the first source labeled "Source 1" established the path to the sink, and then the second source labeled "Source 2" establishes the additional path. In AGIT routing, the path from the second node grows toward to the sink while, in GIT routing, the second source makes the additional path as the perpendicular line to the existing path.

The energy consumption on the paths from the first and second further sources from the sink constitutes a great portion of the total on the path tree. Therefore, we investigate the two source model shown in Fig. 3 in the unit square where the positions of two sources, the first and second further sources from the sink, are uniformly distributed on the chord of the unit quadrant with center at the upper-right corner at which the sink is located. (See Fig. A· 3.)

Let Θ denote the angle between two sources. The probability distribution function $\Pr(\Theta \leq \theta)$ is given as

$$\Pr(\Theta \leq \theta) = 1 - \left(1 - \frac{2}{\pi}\theta\right)^2,\ 0 \leq \theta \leq \frac{\pi}{2}.$$

Therefore, we have the probability density function $f(\theta)$ of the angle Θ as

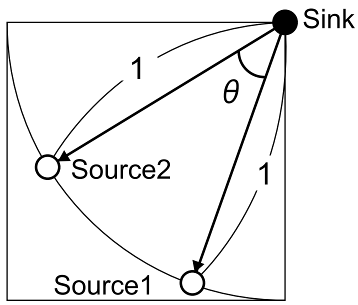

Fig. A· 3 Model of angle between two source.

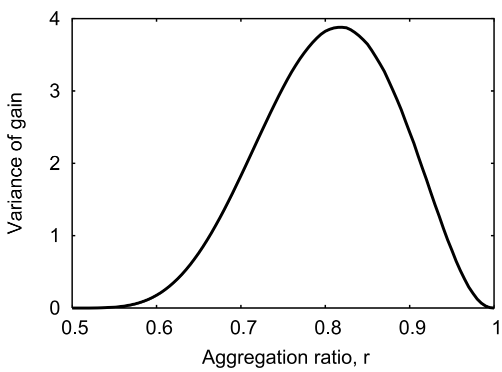

Fig. A· 4 Variance of gain in a two-source model.

$$f(\theta) = \frac{4}{\pi}\left(1 - \frac{2}{\pi}\theta\right). \tag{A· 1}$$

From (6) and (A· 1), we have the first moment $\overline{G(r)}$ and the second moment $\overline{G^2(r)}$ of the gain as follows:

$$\overline{G(r)} = \int_0^{\frac{\pi}{2}} G(r,\theta) f(\theta) \delta\theta,$$
$$\overline{G^2(r)} = \int_0^{\frac{\pi}{2}} G(r,\theta)^2 f(\theta) \delta\theta.$$

Using these expressions, we can calculate the variance $\mathrm{Var}\{G(r)\} = \overline{G^2(r)} - (\overline{G(r)})^2$ numerically.

Figure A· 4 shows the variance of the gain as a function of the aggregation ratio r. We can see that the variance of the gain has a maximum when the aggregation ratio is around 0.8. In Figs. 9 and 10, the aggregation ratio r is 0.55, 0.75, 0.88, and 0.93 when the payload size is equal to 4 bytes, 36 bytes, 108 bytes, and 216 bytes, respectively. From Fig. A· 4, the variance is relatively large in the cases of 36-byte payload and 108-byte payload compared to the other cases. This is the reason why the gain fluctuates especially for the cases of 36-byte and 108-byte payloads in Figs. 9 and 10.

Shinji Mikami received the B.E. degree in electrical and information engineering in 2002 and the M.E. degree in electronic and information system in 2004 both from Kanazawa University, Ishikawa, Japan. He is currently a Ph.D. candidate at the same university. Currently, he is involved in research project of Kobe University to develop ultra low power wireless-network-sensor-node. His research interests include low-power RF circuit designs, media access controls and routing for sensor networks.

Takafumi Aonishi received the B.E. degree in Computer and Systems Engineering from Kobe University in 2005. He is currently a Master's student at Kobe University. His interests include routing for sensor networks.

Hironori Yoshino received the B.E. degree in Computer and Systems Engineering in 2004 and the M.E. degree in graduate school of science and technology in 2006 both from Kobe University. Currently, he is an officer in the Ehime Prefectural Government.

Chikara Ohta received the B.E., M.E. and Ph.D. (Eng.) degrees in communication engineering from Osaka University, Osaka, Japan, in 1990, 1992 and 1995, respectively. From April 1995, he was with the Department of Computer Science, Faculty of Engineering, Gunma University, Gunma, Japan, as an Assistant Professor. In October 1996, he joined the Department of Information Science and Intelligent Systems, Faculty of Engineering, University of Tokushima, Tokushima, Japan, as a Lecturer, and there he had been an Associate Professor since March 2001. Since November 2002, he has been an Associate Professor of the Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University, Japan. From March 2003 to February 2004, he was a visiting scholar in the University of Massachusetts at Amherst, USA. His current research interests include performance evaluation of communication networks. He is a member of IEEE.

Hiroshi Kawaguchi received the B.E. and M.E. degrees in electronic engineering from Chiba University, Chiba, Japan, in 1991 and 1993, respectively. He received the Ph.D. degree in engineering from the University of Tokyo, Tokyo, Japan, in 2006. He joined Konami Corporation, Kobe, Japan, in 1993, where he developed arcade entertainment systems. He moved to the Institute of Industrial Science, the University of Tokyo, as a Technical Associate in 1996, and was appointed to be a Research Associate in 2003. Since 2005, he has been a Research Associate in the Department of Computer and Systems Engineering, Kobe University, Kobe, Japan. He is also a Collaborative Researcher in the Institute of Industrial Science, the University of Tokyo. He is a recipient of the IEEE ISSCC 2004 Takuo Sugano Award for Outstanding Far-East Paper. He has served as a program committee member for IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips). He is a guest associate editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. His current research interests include low-power VLSI design, wireless sensor network, and robot vision. Dr. Kawaguchi is a member of the IEEE and ACM.

Masahiko Yoshimoto received a B.S. degree in electronic engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1975, and an M.S. degree in electronic engineering from Nagoya University, Nagoya, Japan, in 1977. He received a Ph.D. degree in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. He joined the LSI Laboratory, Mitsubishi Electric Corp., Itami, Japan, in April 1977. From 1978 to 1983 he was engaged in the design of NMOS and CMOS static RAM including a 64 K full CMOS RAM with the world's first divided-word-line structure. From 1984, he was involved in research and development of multimedia ULSI systems for digital broadcasting and digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology. Since 2000, he has been a Professor of the Dept. of Electrical and Electronic Systems Engineering at Kanazawa University, Japan. Since 2004, he has been a Professor of the Dept. of Computer and Systems Engineering at Kobe University, Japan. His current activity is focused on research and development of multimedia and ubiquitous media VLSI systems including an ultra-low-power image compression processor and a low power wireless interface circuit. He holds 70 registered patents. He served on the Program Committee of the IEEE International Solid State Circuit Conference from 1991 to 1993. In addition, he has served as a Guest Editor for special issues on Low-Power System LSI, IP, and Related Technologies of IEICE Transactions in 2004. He received the R&D100 awards from R&D Magazine for development of the DISP and development of a real-time MPEG2 video encoder chipset in 1990 and 1996, respectively.