Data gathering using mobile agents for reducing traffic in dense mobile wireless sensor networks

Keisuke Goto, Yuya Sasaki, Takahiro Hara* and Shojiro Nishio
Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Osaka, Japan

Abstract. Recently, there has been increasing interest in Mobile Wireless Sensor Networks (MWSNs) that are constructed by mobile sensor nodes held by ordinary people, and it has led to a new concept called *urban sensing*. In such MWSNs, mobile sensor nodes densely exist, and thus, there are basically many sensor nodes that can sense a geographical point in the entire sensing area. To reduce the communication cost for gathering sensor data, it is desirable to gather the sensor data from the minimum number of mobile sensor nodes which are necessary to guarantee the sensing coverage or the quality of services. In this paper, to achieve this, we propose a data gathering method using mobile agents in dense MWSNs. The proposed method guarantees the sensing coverage of the entire area using mobile agents that autonomously perform sensing operations, transmit sensor data, and move between sensor nodes. By gathering only sensor data generated by sensor nodes where mobile agents are running, our proposed method can achieve efficient gathering of sensor data.

Keywords: Mobile wireless sensor networks, data gathering, mobile agent, geo-routing

1. Introduction

Recently, there has been a great deal of interest in Wireless Sensor Networks (WSNs [12,17,22]) because advances in semiconductor and wireless communication technologies have led to the development of small and inexpensive sensor devices. One of typical applications of WSNs is a monitoring application. A lot of sensor nodes sense an environment and send their readings (sensor data) to the sink.

Due to the advances in radio communication and computer technologies, Mobile Ad Hoc Networks (MANETs [1,8,9]), which are constructed by only mobile nodes, have also been actively studied for recent years. In MANETs, every mobile node plays the role of a router, and they communicate with each other. As an integration of WSNs and MANETs, Mobile Wireless Sensor Networks (MWSNs), which are constructed by mobile nodes with sensor devices, have recently attracted much attention.

Urban sensing, in which ordinary people cooperate for data gathering with some incentive, is a typical example of MWSNs. Traditionally, urban sensing generally assumes some infrastructures, i.e., the Internet for communication and data gathering. However, since many applications share and compete for

---

*Corresponding author: Takahiro Hara, Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan. Tel.: +81 6 6879 4513; Fax: +81 6 6879 4514; E-mail: hara@ist.osaka-u.ac.jp.

limited network resources in the Internet, it is desirable to minimize traffic that MWSNs injects into the Internet. For this aim, MWSNs constructed by mobile sensor nodes (e.g., PDA and smart phones with sensor devices held by ordinary people) have attracted much attention for urban sensing because they do not require any infrastructures [3,13,14,16]. Sensor nodes in such MWSNs automatically sense an environment (e.g., sound and temperature), and a sink gathers the sensor data using multi-hop radio communication between sensor nodes (and the sink).

In MWSNs constructed by sensor nodes held by ordinary people, the number of sensor nodes is generally very large, and thus, there are basically many sensor nodes that can sense (cover) a geographical point in the entire sensing area (i.e., dense MWSNs). From the perspective of applications, a lot of same sensor data are not useful, but just waste limited network bandwidth. Rather, in most cases, applications require a certain geographical granularity of sensing, e.g., sensor data of every 100 [m]×100 [m] square. In such a situation, if a sink gathers sensor data from all sensor nodes, it unnecessarily wastes the network bandwidth and the battery of sensor nodes. To reduce the data traffic for data gathering, it is desirable to efficiently gather sensor data so that the geographical granularity required from an application can be guaranteed with the minimum number of sensor nodes.

Achieving this is not easy because sensor nodes move freely, and thus, sensor nodes that can sense a specific point dynamically change. A naive method is that a sink determines sensor nodes that perform sensing and send back the data every time of sensing. However, this method requires exchange of requests and replies between the sink and these nodes, which produces large traffic and cause a long delay for message exchanges particularly when the distances between the sink and sensor nodes are long.

In this paper, we propose a data gathering method that efficiently gathers sensor data using mobile agents in dense MWSNs. A mobile agent is an application software that autonomously operates on a sensor node and moves between sensor nodes. In our proposed method, mobile agents are generated by the sink and allocated on sensor nodes located near the sensing points, which are determined from the requirement on geographical granularity of sensing. Each mobile agent moves from the currently located sensor node to the nearest sensor node from the sensing point responding to the movement of sensor nodes. This makes the mobile agent continuously locates near the sensing point. Every time when the sensing time comes, sensor nodes where mobile agents locate perform sensing and send the sensor data to the sink. Our proposed method can reduce the traffic for gathering the sensor data since mobile agents control sensor nodes for sensing operations and transmissions of sensor data. The contributions of this paper are summarized as follows.

- Our proposed method reduces the traffic for sensor data gathering, which is effective in dense MWSNs because the network bandwidth and batteries of sensor nodes are limited.
- It is natural to assume that applications require a certain geographical granularity of sensor data. Thus, in our proposed method, the sink divides the sensing area into lattice-shaped sub-areas according to the required granularity, and determines sensing points as the center points of the sub-areas. Then mobile agents, which are allocated by the sink on sensor nodes located near the sensing points, guarantee the requirement of geographical granularity of sensing by controlling transmissions of sensor data. To the best of our knowledge, this is the first effort that addresses the issue of guaranteeing the geographical granularity of sensing in an efficient manner in dynamic MWSNs.
- Through extensive simulations assuming practical and various situations, we show that our proposed method works very well in terms of traffic reduction and guaranteeing the requirement of geographical granularity of sensor data.

The remainder of this paper is organized as follows. In Section 2, we introduce related works. In Section 3, we describe assumptions in this paper. In Section 4, we explain the details of our proposed

method. In Section 5, we show the results of the simulation experiments. Finally, in Section 6, we summarize this paper.

2. Related works

In [10], the authors proposed a hierarchical data gathering method in WSNs. In this method sensor nodes are hierarchically arranged, where a sensor node in a lower level sends the sensor data to a node in the higher level, and the sensor node in the highest level sends the (aggregated) sensor data to the sink. It can reduce the traffic for data gathering since nodes in higher levels aggregate and compress the sensor data which are sent from nodes in lower levels. In [6], the authors proposed a data gathering method using mobile agents in WSNs. Mobile agents are generated by a sink and gather sensor data by visiting sensor nodes. This method can reduce the traffic for data gathering since mobile agents compress the sensor data every time when they visit a sensor node. In [21], the authors proposed a topology control protocol, which splits the sensing area into multiple grid cells. The size of a grid cell is determined based on the communication range of sensor nodes so that any pair of sensor nodes in adjacent two cells can directly communicate with each other. Sensor data are forwarded to the sink through the cells, where one node in each cell plays the role of a router. By doing so, other sensor nodes can reduce the energy consumption.

In [5,7], the authors proposed a duty cycling algorithm, which aims to reduce the cost of topology management by changing the state of redundant sensor nodes to the sleep mode. In the algorithm proposed in [5], a sensor node changes its state to the sleep mode for a certain time in order to save its battery when its neighbor nodes can communicate with each other directly or via another sensor node without itself. In the algorithm proposed in [7], each sensor node periodically sends a HELLO message to its neighbor nodes. Then, sensor nodes that received HELLO messages more than a certain predetermined number change their state to the sleep mode because it means that they have many activate neighbors.

In [4,18,19], the authors proposed a network configuration method that guarantees the coverage in the entire area, i.e., all positions are covered by at least a predetermined number of sensor nodes. In the method proposed in [4], each sensor node periodically sends a HELLO message including the information on its position and state to its neighbor nodes. Sensor nodes with a small amount of remaining battery change their state to the sleep mode until the next period in order to save their battery if their sensing areas are covered by their neighbor nodes. In the method proposed in [18], similar to the method in [4], sensor nodes with less remaining battery move to the sleep mode. In doing so, the network connectivity is also taken into account. Specifically, this method adopts the technique proposed in [5] to guarantee the network connectivity when deciding the transit of mode. In the method proposed in [19], sensor nodes have two different working phases; *initialization phase* and *sensing phase*. Sensor nodes in the initialization phase make a working schedule by sharing the information on their positions and synchronizing times with their neighbors. The working schedule is determined so that it can guarantee that all positions in the entire sensing area are covered by at least one sensor node. In the sensing phase, sensor nodes save their battery by switching their modes (active and sleep) based on the working schedule.

The existing studies presented above do not assume the movements of sensor nodes, and thus, cannot handle the change of network topology. Furthermore, these studies are different from our work that aims data gathering to guarantee the geographical granularity of sensing which is required from an application.
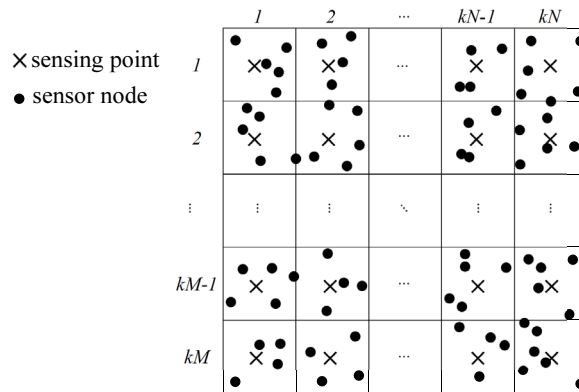
Fig. 1. Sensing area and sensing points.

In [20], the authors assumed location based services and proposed a data gathering and disseminating method in MANETs. The method uses a mobile agent that stays within a certain geographical area by moving between mobile nodes. This work assumes services to disseminate location based information that is generated by disseminating nodes and passed to mobile nodes located near those locations. This work is different from our work which assumes that sensor data generated by sensor nodes are sent to the sink located far from them.

3. Assumptions

In this paper, we assume dense MWSNs constructed by mobile sensor nodes which are held by ordinary people and equipped with a radio communication facility. These sensor nodes periodically observe (sense) a physical phenomenon (e.g., sound, temperature and light), and communicate with each other using multi-hop radio communication. According to the requirement from an application, the sink periodically monitors the sensing area while guaranteeing the geographical granularity of sensing. More specifically, the sink gathers sensor data from sensor nodes located near the sensing points which are determined from the requirement of the geographical granularity at the timing of sensing.

3.1. System environment

The sensing area is assumed to be a two-dimensional plane whose horizontal to vertical ratio is $M{:}N$ (M and N are positive integers). The application specifies its requirement of the geographical granularity of sensing as an integer of $k^2 \cdot M \cdot N$ ($k = 1, 2, \cdots$). Then, the sink divides the sensing area into $k \cdot M \times k \cdot N$ lattice-shaped sub-areas and determines the center point of each sub-area as a sensing point, which is the target of data gathering (see Fig. 1).

As mentioned, we assume MWSNs constructed by mobile sensor nodes held by ordinary people. Since no infrastructure for communication is available in the sensing area, the sink gathers sensor data by using a MANET constructed by the sensor nodes. The communication range of each sensor node is a circle with a radius of r. Each sensor node is equipped with a positioning device such as GPS, and they communicate with each other using multi-hop radio communication based on their positions (i.e., geo-routing described in the next Subsection). The position information is represented as a pair of longitude and latitude.
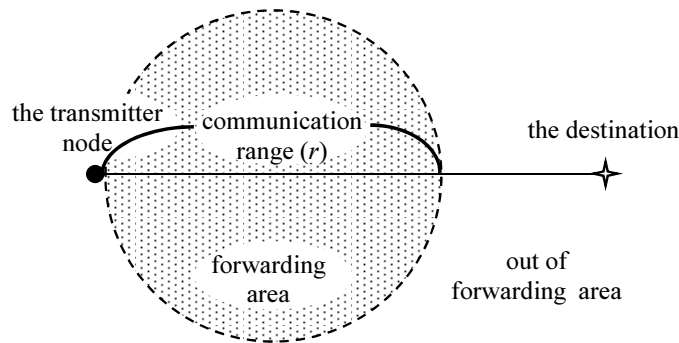
Fig. 2. An example of forwarding area.

Each sensor node freely moves in the sensing area, while the sink is stationary. Since the number of mobile nodes is very large, there are many sensor nodes that can sense (cover) a geographical point in the entire sensing area.

3.2. Geo-routing

Sensor nodes adopt a geo-routing protocol based on that proposed in [11] to transport a message to the destination specified as a position (not a node).

In this protocol, nodes perform a transmission process using the information on positions of the transmitter node and the destination, which is specified in the packet header. Specifically, the transmitter node writes the information on positions of the destination and itself into the packet header of the message, and broadcasts it to its neighboring nodes. Each node that received this message judges whether it locates within the forwarding area. Here, the forwarding area is determined based on the positions of the transmitter node and the destination and the communication range, so that any node in the forwarding area is closer to the destination than the transmitter node and can communicate directly (by one hop) to all nodes in the forwarding area. Figure 2 shows an example of the forwarding area. The forwarding area is represented by the shaded circle whose diameter is the communication range of sensor nodes (r).

The node within the forwarding area sets the waiting time, and then it forwards the message after the waiting time elapses unless it has detected the message sent by another node during the waiting time. Here, the waiting time (WT) is determined by the following equation, where it gets shorter as the distance (P) between the node and the destination gets shorter:

$$WT = Max\_delay \cdot \left(\frac{r - P}{r} \right) \tag{1}$$

Max_delay is the maximum waiting time which is a positive constant. Each node within the forwarding area cancels its transmission process when it detects the message forwarded by another node. By repeating this procedure, the message is forwarded to nodes which are closer to the destination. If the transmitter node is within half of the communication range ($r/2$) from the destination, each node that received the message sends an ACK to the transmitter node after the waiting time elapses instead of forwarding the message. As the result, the nearest node from the destination (that has sent the ACK) can find that it is the nearest one because all nodes within $r/2$ from the destination can detect the ACK sent by the node and cancel to send own ACK. If the transmitter node did not receive an ACK from any node, it also can find that it is the nearest node.

Here, it is very rare but possible that multiple nodes try to reply ACKs at the same timing when their distances to the destination of the message are equivalent. However, even in this case, each node does not send an ACK at the same timing by using a multiple access method (e.g., CSMA/CA) in the MAC layer. Therefore, nodes that receive (overhear) an ACK can stop sending their own ACK, which makes only one node replies an ACK.

4. A data gathering method using mobile agents

In this section, first we describe the design policy of our proposed method and explain about mobile agents. Then, we describe the outline and detailed behaviors of our proposed method.

4.1. Design policy

In MWSNs constructed by mobile sensor nodes held by ordinary people, there are a large number of sensor nodes that move freely, which causes frequent network topology changes. In such an environment, managing topology information and keeping routing tables up-to-date using beacons are impractical because it generates large traffic for exchanging beacons, which consumes a large amount of energy and also might cause frequent packet collisions due to the limited network bandwidth. Therefore, as described in Section 3.2, we adopt a geo-routing protocol, which transmits a packet to the sensor node located near the destination (sensing point) by selecting sensor nodes forwarding the packet based on their positions.

Moreover, it is natural that sensor nodes held by ordinary people do not know the information for data gathering (e.g., the position of a sink and the requirement of geographical granularity of sensing from an application). Therefore a sink has to disseminate the information to sensor nodes located near the sensing points. However, disseminating the information to all of those sensor nodes every time when the sensing time comes is not efficient because it produces large traffic as well as causes some delay for data gathering. To solve this problem, in our proposed method, a sink gathers sensor data using mobile agents each of which operates on a sensor node that is located near a sensing point and covers it. By doing so, sensor data can be gathered from the minimum number of sensor nodes necessary to guarantee the requirement of geographical granularity of sensing. Moreover, it is not necessary to send the information for data gathering from the sink every time when the sensing time comes. As the result, our proposed method can reduce traffic and delay for gathering sensor data.

4.2. Mobile agent

In our proposed method, a mobile agent is an application software that autonomously operates on a sensor node and moves between sensor nodes. A sensor node boots up a mobile agent by referring to the agent data, which consists of the information on geographical granularity of sensing, the sensing cycle and the position of the sink. The role of mobile agents is to transmit a sensor data to the sink at every sensing time. Mobile agents are deployed on sensor nodes so that they can guarantee the requirement from the application regarding the geographical granularity of sensing.

Table 1
Forwarding directions of agent data

Location of the sensor node that previously forwarded the agent data	Forwarding directions of the agent data
– (sink)	up, down, left, right
sub-area next to right	up, down, left
sub-area next to left	up, down, right
sub-area next to down	up
sub-area next to up	down

4.3. Outline of the proposed method

In our proposed method, initially, the sink generates mobile agents and deploys them in the network. Specifically, the sink sends the agent data to $k^2 \cdot M \cdot N$ sensing points that are determined as described in Subsection 3.1. According to Subsection 4.4, the sensor node closest to each sensing point receives the agent data and boots up a mobile agent.

At each sensing time, according to the method described in Subsection 4.5 the mobile agents send sensor data generated by sensor nodes on which they run to the sink. If a mobile agent disappears, the sink and other mobile agents redeploy the disappeared agent and the redeployed mobile agent retransmits the sensor data to the sink, according to the method described in Subsection 4.6. Moreover, if a sensor node on which a mobile agent runs moves away from the sensing point, the mobile agent moves from the sensor node to another node which is the closest to the sensing point, according to the method described in Subsection 4.7.

4.4. Deployment of mobile agents

Our proposed method deploys mobile agents in the network with small traffic. Specifically, the sink sends the agent data along the forwarding tree that is constructed based on the geographical relationships among sensing points. The procedures of the sink and mobile agents to deploy mobile agents are as follows.

1. The sink generates the agent data and sends it to the sensing point in the sub-area in which the sink locates using the geo-routing protocol described in Subsection 3.2.
2. When the sensor node located closest to the sensing point in its existing sub-area receives the agent data, it boots up a mobile agent. As the initial operation, the mobile agent retransmits the agent data to the sensing points in some of sub-areas which are adjacent to its existing sub-area. Table 1 shows the directions to which the mobile agent forwards the agent data in the lattice-shaped sub-areas. For example, if the agent data is transmitted by the sink, the mobile agent forwards the agent data to upper, lower, left and right sub-areas. If the location (sub-area) of the sensor node that forwarded the agent data previously is the sub-area next to left, the mobile agent forwards the agent data to upper, lower and right sub-areas.
   Then, the procedure returns to step 2. Here, mobile agents in sub-areas at the top and bottom edges of the sensing area do not retransmit the agent data.

Through the above procedures, mobile agents dynamically construct a tree structure for forwarding the agent data (e.g., Fig. 3), which we call the *forwarding tree*. The forwarding tree is also used for collecting sensor data, which enables mobile agents to reduce traffic by aggregating sensor data received from child agents and that of itself when they reply the sensor data to their parent agents (describe in the next subsection).
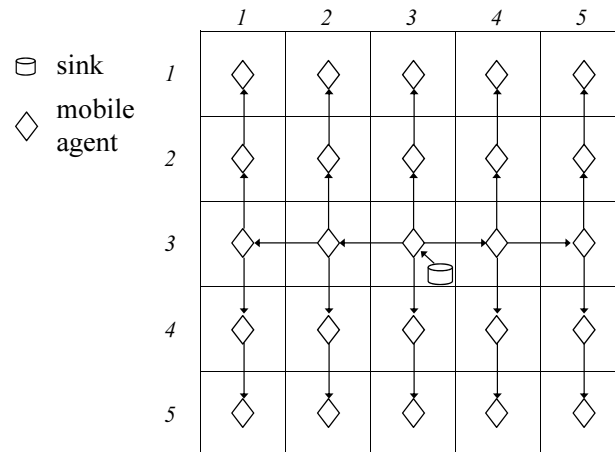
Fig. 3. Example of forwarding agent data between mobile agents.

We show an example that the sink deploys mobile agents to all sensing points using Fig. 3. Let us assume that the sensing area is divided into 5×5 sub-areas and the sink locates in the sub-area $(3, 3)$. Initially, the sink forwards the agent data to the sensing point in the sub-area in which the sink exists. The mobile agent that is booted up by the sensor node closest to the sensing point retransmits the agent data to the four sensing points in the upper, lower, left, and right sub-areas. Then, mobile agents in the four sub-areas are booted up and they retransmit the agent data to directions shown in Table 1. By repeating these procedures, the sink can deploy mobile agents near all the sensing points.

4.5. Transmission of sensor data

Mobile agents deployed at (near) sensing points send the sensor data held by sensor nodes on which these agents operate to the sink at every sensing time. Our proposed method can reduce the traffic of sending sensor data since mobile agents compress the received sensor data that are sent along the forwarding tree in the reverse direction. The detailed procedures of mobile agents are as follows.

1. At every sensing time, mobile agents located in sub-areas at the top and bottom edges of the sensing area start to send their sensor data to the mobile agents that forwarded the agent data to them. In doing so, the geo-routing protocol described in Subsection 3.2 is used. Here, due to the movement of sensor nodes on which mobile agents operate, the mobile agents may become far away from the sensing points. In our geo-routing protocol, sensor data cannot be received by the mobile agent which is responsible for a sensing point if the mobile agent has moved farther than $r/2$ from the sensing point. Therefore, in our method, a mobile agent moves from the sensor node on which it operates to the node closest to the sensing point so that it keeps to locate within $r/2$ from the sensing point. The procedure to achieve this is described in Subsection 4.7. By doing so, mobile agents can always receive the sensor data sent from child nodes in the forwarding tree.
2. When mobile agents located in sub-areas except for that the sink exists receive the sensor data from all mobile agents which are children in the forwarding tree, they aggregate the received sensor data and their own sensor data, and then forward the aggregated data to the mobile agents which are parent in the forwarding tree. The procedure returns to step 2.
3. The mobile agent located in the sub-area where the sink exists aggregates the received sensor data and its own sensor data, and then sends it to the sink.

4.6. Redeployment of mobile agents and retransmission of sensor data

If mobile agents and the sink cannot receive sensor data from their children in a given time it shows that, a packet loss may occur or a mobile agent may disappear. When a packet loss occurs, the mobile agent that cannot receive the sensor data simply requests its child to re-send the data. However, when a mobile agent disappears, the missing mobile agent must be redeployed because otherwise, sensor data cannot be transmitted to the sink. Therefore, a mechanism is necessary to retransmit sensor data and redeploy a mobile agent. The detailed procedures of redeployment of mobile agents and retransmission of sensor data are as follows.

1. At every sensing time, all mobile agents transmit sensor data based on the procedures described in Subsection 4.5.
2. When mobile agents located in sub-areas other than the sub-areas at the top and bottom edges of the sensing area and where the sink exists cannot receive sensor data from any of their children within time period δ, they send a data request message to the child using the geo-routing protocol described in Subsection 3.2.
3. When mobile agents located in sub-areas at the top and bottom edges of the sensing area receive a data request message, they reply by sending the requested sensor data. When mobile agents located in sub-areas other than those at the top and bottom edges of the sensing area receive a data request message, they reply by sending the sensor data unless they have already received the sensor data from all mobile agents which are children in the forwarding tree. If not, then they reply with an existence message instead of the sensor data.
4. Mobile agents that have sent a data request message restart the transmission of sensor data if they receive the sensor data from all mobile agents that are children in the forwarding tree. If they cannot receive the sensor data or an existence message, they determine that the mobile agent that is its child has disappeared, and they send new agent data to the sensing point located in the sub-area where the mobile agent has disappeared.
5. The sensor node located closest to the sensing point in the sub-area in which the mobile agent has disappeared receives the agent data and boots up a new mobile agent to that area. In the initial operation, if a mobile agent in the sub-area at the top and bottom edges of the sensing area is redeployed, it sends its sensor data. If a mobile agent located in a sub-area other than the sub-areas at the top and bottom edges of the sensing area is redeployed, it sends a data request message to all mobile agents that are children in the forwarding tree. The procedure returns to step 3.
   On the other hand, when an existing mobile agent detects the agent data because the transmission of its sensor data has failed (e.g., in packet collisions), the mobile agent replies with an ACK and executes the same process as if it had received a data request message (step 3), and then discards the agent data.

Through the above procedures, mobile agents and the sink send a data request message to mobile agents that are their children in the forwarding tree only if they cannot receive sensor data from the child agents within time period δ. After that, these mobile agents and the sink send agent data only if they cannot receive a response from their children within δ. In this way, our proposed method can redeploy mobile agents and retransmit sensor data without additional traffic in the location (sub-areas) in which mobile agents successfully transmit their sensor data and are still alive.

4.7. Movement of mobile agent

If a sensor node on which a mobile agent operates moves away from the sensing point, it may not be able to sense (cover) the point. Also, as mentioned, it may not be able to receive the sensor data sent

from its children in the forwarding tree. Therefore, in our method, to avoid such situations, a mobile agent moves from the current sensor node to the other node which is the closest to the sensing point.

Specifically, a mobile agent starts moving when the distance between the sensing point and itself becomes longer than threshold α. Here, α is a system parameter which is set as a constant smaller than $r/2$ and the range of sensing of each sensor node (sensing coverage), which can guarantee that a sensor node on which a mobile agent operates can communicate with all sensor nodes located near (within $r/2$) from the sensing point and can sense the data at the sensing point.

In order to move to the sensor node closest to the sensing point, the mobile agent broadcasts a message containing the agent data, to neighbor sensor nodes within $r/2$ from the sensing point. Sensor nodes that received this message set the waiting time according to equation (1) and send an ACK to the transmitter node. Since the sensor node closest to the sensing point firstly sends an ACK, it boots up a mobile agent in the same way as in Subsection 3.2. Other sensor nodes detect this ACK and cancel to send own ACK. After receiving the ACK, the original mobile agent stops its operation.

Here, it is possible (but very rare) that there are multiple sensor nodes which locate closest to the sensing point with the same distance when a mobile agent needs to move. Even in this case, since each node sends an ACK at different timings with each other by using a multiple access method in the MAC layer, only the node that first sent an ACK boots up a new agent and the other nodes discard the agent data.

5. Simulation experiments

In this section, we show the results of simulation experiments regarding the performance evaluation of our proposed method. For the simulations, we used the network simulator, Scenargie 1.3 (Subsections 5.2 $\sim$ 5.5) and 1.4 (Subsection 5.6) [15].[1] First, in order to verify the effectiveness of using mobile agents, we evaluate our method without the mechanism of retransmission of sensor data and redeployment of mobile agents described in Subsection 4.6. Then, we evaluate our method with that mechanism in Subsection 5.6.

5.1. Simulation model

There are n mobile sensor nodes $(M_1, \cdots, M_n)$ and a sink (S_1) in a two-dimensional field of is 1000 [m] $\times$1000 [m]. The sink is fixed at the point of (580 [m], 580 [m]) from the left and the bottom edges of the sensing field. Each sensor node moves according to the random walk mobility model [2] where it selects a random direction and a random speed from 0 to 1 [m/sec] at intervals of 50 [sec]. The sink and sensor nodes communicate with IEEE 802.11g whose transmission rate is 6 [Mbps] and communication range r is about 100 [m]. Each sensor node continuously senses the field and the valid range of its sensing (i.e., sensing coverage) is 100 [m]. The sink divides the field into G lattice-shaped sub-areas whose the size is $1000/\sqrt{G}$ [m]$\times 1000/\sqrt{G}$ [m] and sets the center point of each sub-area as a sensing point.

The sink deploys a mobile agent at each sensing point after 400 [sec] from the start time of the simulation. The size of the agent data is set as 128 [B], assuming that each sensor node has the source code of mobile agent in advance. The size of sensor data generated at each sensor node is set as D [B].

---

[1]The differences between the two versions (1.3 and 1.4) of Scenargie have no impact on our simulation results.

Table 2
Parameter configuration.

Parameter (Meaning)	Value	(Range)
T (Sensing cycle [sec])	120	$(30 \sim 300)$
G (Number of sub-areas)	25	$(1 \sim 100)$
n (Number of sensor nodes)	2000	$(1000 \sim 3000)$
D (Size of a sensor data item)	32	$(24 \sim 240)$

Table 3
Message size

Method	Roll (Message name)		Size [B]
Our proposed method	Deploying a mobile agent	(Deployment message)	192
	Sending sensor data	(Data message)	$64 + D \cdot i$
	Moving a mobile agent	(Movement message)	160
Comparative method	Requesting sensor data	(Request message)	72
	Sending sensor data	(Data message)	$64 + D$
Common	ACK	(ACK message)	32

The sensing cycle is set as T [sec], i.e., the sensing time is $400 + mT$ [sec] $(m = 0, 1, \cdots)$ until the end of the simulation. At every sensing time, the sensor nodes on which mobile agents operate send sensor data to the sink. Moreover, a mobile agent moves to the sensor node which is closest to the sensing point when the distance between the sensing point and itself becomes longer than 47 [m] (i.e., $\alpha = 47$), which is set as an appropriate value according to our preliminary experiments.

For comparison, we also evaluate the performance of the method where the sink sends a request message to all sensing points at every sensing time and collects the sensor data. Specifically, at every sensing time, the sink individually sends messages for requesting a sensor data to each sensing point using the geo-routing protocol described in Subsection 3.2. Here, the sink sets a constant interval of 0.1 [sec] between every pair of request messages, and sends them to the sensing points from the bottom-left to top-right sub-areas in order to avoid packet collisions.

Table 2 shows the four parameters T, G, n and D, and their values are used in the simulation experiments. The parameters are basically fixed to constant values specified by numbers to the left of the parenthetic values, but are varied specified by the parenthetic elements in Table 2 in order to verify the impact on the performance of our proposed method.

In the above simulation model, we performed 50 experiments in which the initial position of each mobile sensor nodes was randomly determined where there was the same number of sensor nodes in all sub-areas. The total simulation time was 4000 [sec], and we evaluated the following four criteria.

1. Delivery ratio: We judge that the sink has succeeded the data gathering if it acquired sensor data from all sensing points. The delivery ratio is defined as the ratio of the number of successes in data gathering to the total number of sensing times occurred during the simulation.
2. Delay: The delay is defined as the average elapsed time from the start of each sensing time to the time that the sink successfully receives all sensor data.
3. Number of packets: The number of packets is defined as the total number of the packets transmitted by the sink and all sensor nodes during the simulation.
4. Traffic: The traffic is defined as the summation of the packets size of all packets sent by the sink and all sensor nodes during the simulation. Table 3 shows messages used in our method and the comparative method and their sizes at the application layer. In this table, i denotes the number of sensor data aggregated in a data message. The message size in the Mac layer is calculated by adding 64 [B] (the header size of a Mac message) to the message size at the application layer.
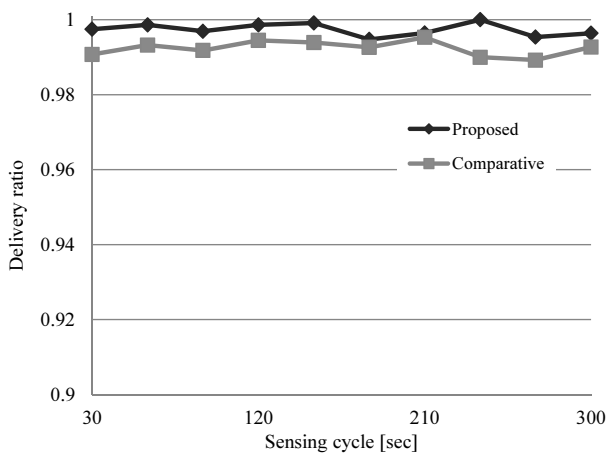
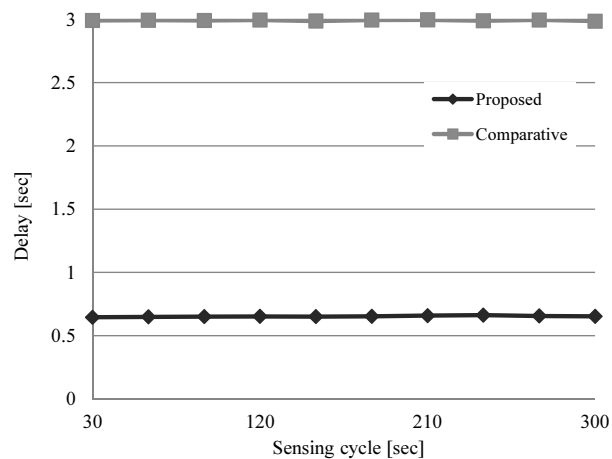Fig. 4. Effects of sensing cycle (Delivery ratio).

Fig. 5. Effects of sensing cycle (Delay).

5.2. *Effects of the sensing cycle*

First, we examine the effects of T, the sensing cycle [sec]. Figures 4 through 7 show the simulation results. In these graphs, the horizontal axes indicate T, and the vertical axes indicate the delivery ratio in Fig. 4, the delay in Fig. 5, the number of packets in Fig. 6, and the traffic in Fig. 7.

Figure 4 shows that the delivery ratio in both methods is very high. However, both methods could not achieve 100 [%] of delivery ratio because packet collisions infrequently occurred during geo-routing. In our proposed method, there is another case that data gathering fails. Specifically, a data gathering fail happens when a data message arrives at a mobile agent while the mobile agent moves between sensor nodes. On the other hand, our method produces less traffic than the comparative method (see Fig. 6), and thus, can reduce packet collisions. As the result of these two factors (negative and positive), the delivery ratio in our proposed method is almost same as that in the comparative method.

Figure 5 shows that the delay is not affected by the sensing cycle. It also shows that our proposed method can gather sensor data in much shorter time than the comparative method. In our proposed method, the sensor data are concurrently sent from mobile agents to the sink. On the other hand, in the comparative method, the sensor data are sent from sensor nodes located closest to the sensing points after receiving a request message which is successively sent from the sink with an interval (0.1 [sec]). Here, the longest waiting time until receiving a request is $0.1 \times (G-1)$ [sec]. Although it can be reduced by setting a smaller interval, we have confirmed from our preliminary experiment that packet collisions frequently occur and the delivery ratio heavily decreases if the interval is set as less than 0.1 [sec].

Figure 6 shows that the number of packets in our proposed method is much less than that in the comparative method. In the comparative method, the sink individually sends the request messages to sensing points at every sensing time, and the sensor nodes located closest to the sensing point individually send back their sensor data. On the other hand, in our proposed method, the sink does not issue any request messages because mobile agents autonomously send the sensor data to the sink. In addition, the number of data messages in our proposed method is less than that in the comparative method because mobile agents aggregate the received sensor data and its own sensor data. As the sensing cycle gets longer, the number of packets in both methods decreases because the number of sensing times decreases. When the sensing cycle is long, the difference in the number of packets between our proposed method and the comparative method is small. This is because the impact of deployment and movement messages, which
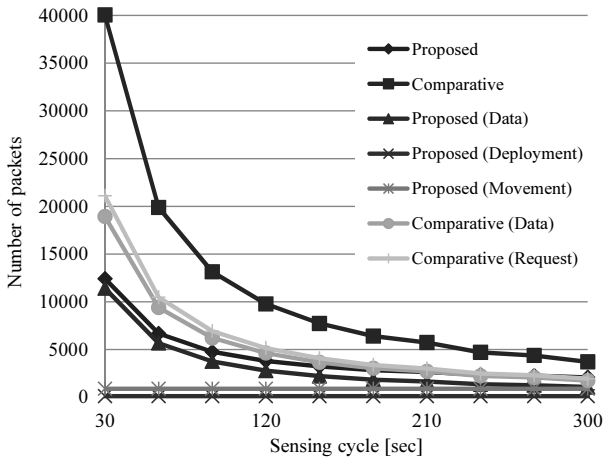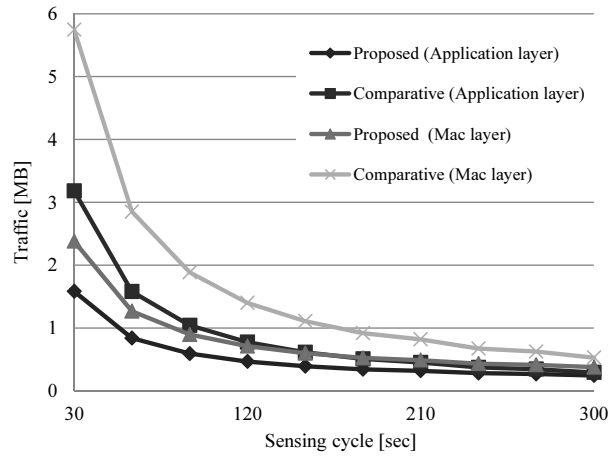
Fig. 6. Effects of sensing cycle (Number of packets).

Fig. 7. Effects of sensing cycle (Traffic).

are issued only in our proposed method and not affected by the sensing cycle, becomes more dominant while other packets decrease as the sensing cycle increases.

Figure 7 shows that the traffic in our proposed method is much smaller than that in the comparative method. This is due to the reduction of the number of messages by aggregating sensor data contributes to reduce the traffic. This can be seen from the result that the difference in traffic between our method and the comparative method in the Mac layer is larger than that in the application layer because the proportion of the header size to the total message size is larger in the Mac layer.

As the sensing cycle increases, the traffic in both methods decreases. When the sensing cycle is long, the difference in traffic between our proposed method and the comparative method is small.

5.3. Effects of the number of sub-areas

Next, we examine the effects of G, the number of sub-areas. Figures 8 through 11 show the simulation results. In these graphs, the horizontal axes indicate G, and the vertical axes indicate the delivery ratio in Fig. 8, the delay in Fig. 9, the number of packets in Fig. 10, and the traffic in Fig. 11.

Figure 8 shows that as G increases, the delivery ratio in both methods decreases because the increase of sub-areas means the increase of sensor data collected, and thus, packet collisions occur more often. Moreover, in our proposed method, as G increases, the cases that sensor data are missing while mobile agents move between sensor nodes (as described in the previous subsection) increase. As the result, the delivery ratio in our proposed method becomes smaller than that in the comparative method.

Figure 9 shows that as G increases, the delay in the comparative method increases because the sink successively sends request messages to all of G sensing points with a constant interval. More specifically, it takes at least $0.1 \times (G - 1)$ [sec] for sending request messages to all sub-areas. On the contrary, our proposed method can gather data in a short time regardless of G because our proposed method does not issue any request messages and mobile agents send the sensor data to the sink along the forwarding tree.

Figure 10 shows that as G increases, the number of packets in both methods increases because of the number of sensor data transmitted to the sink increases. However, the number of packets in our proposed method is much fewer than that in the comparative method. This is due to the same reason as that in Fig. 6.
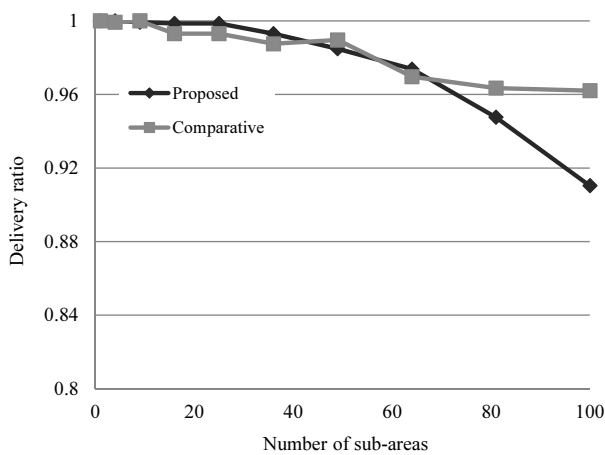
Fig. 8. Effects of number of sub-areas (Delivery ratio).
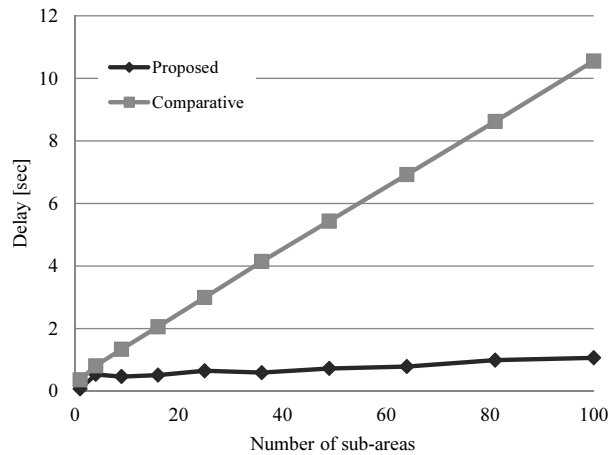

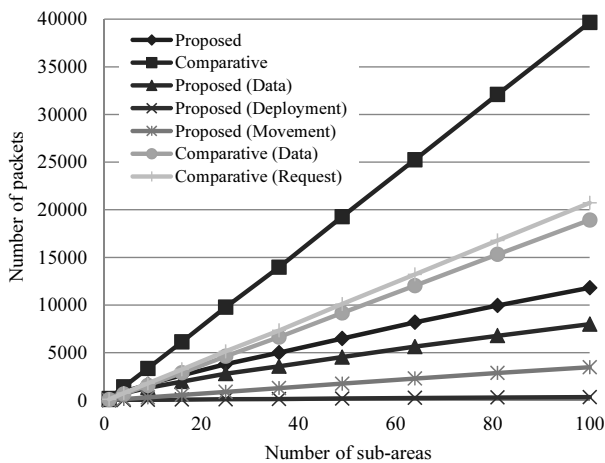
Fig. 9. Effects of number of sub-areas (Delay).

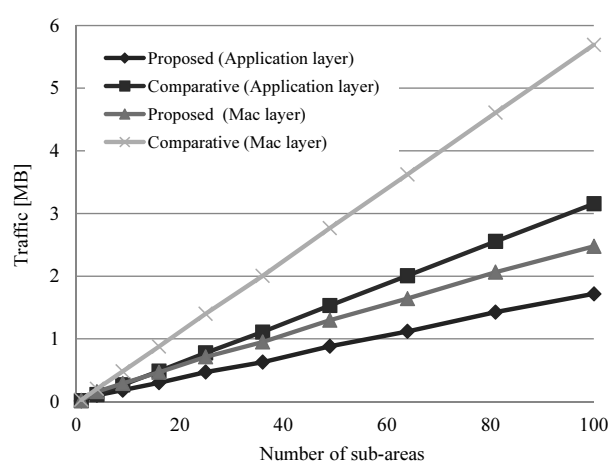Fig. 10. Effects of number of sub-areas (Number of packets).

Fig. 11. Effects of number of sub-areas (Traffic).

Figure 11 shows that as G increases, the traffic in both methods increases. However, the traffic in our method is much less than that in the comparative method. This is due to the same reason as that in Figure 7.

5.4. Effects of the number of sensor nodes

We examine the effects of n, the number of sensor nodes. Figures 12 through 15 show the simulation results. In these graphs, the horizontal axes indicate n, and the vertical axes indicate the delivery ratio in Fig. 12, the delay in Fig. 13, the number of packets in Fig. 14, and the traffic in Fig. 15.

Figure 12 shows that when n is very small, the delivery ratio in both methods is small because it happens frequently that no sensor nodes exist in the forwarding area in geo-routing or no sensor nodes exist that can sense a certain sensing point. Moreover, in our proposed method, if there are no sensor nodes within the circle with radius α from the sensing point, the mobile agent which is responsible for that point cannot move to a new sensor node, and thus, the agent for that point disappears from the
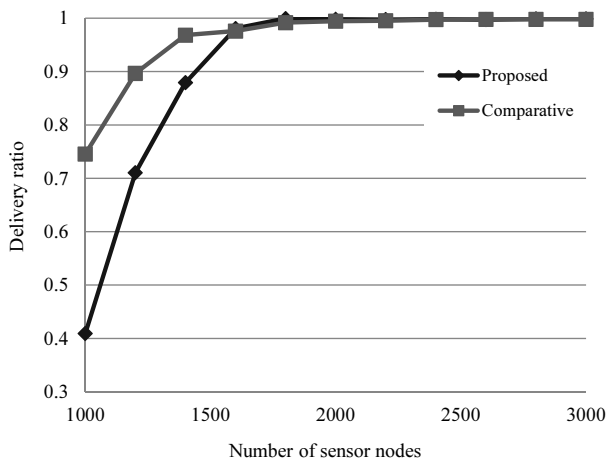
Fig. 12. Effects of the number of sensor nodes (Delivery ratio).
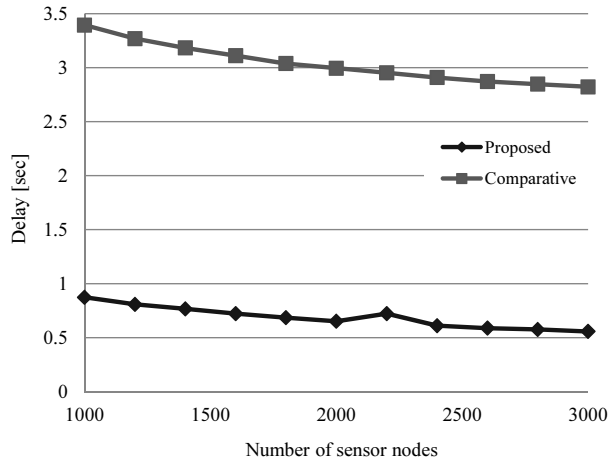
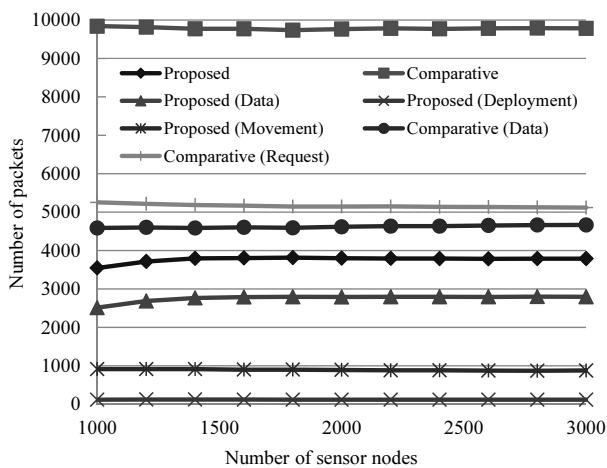Fig. 13. Effects of the number of sensor nodes (Delay).

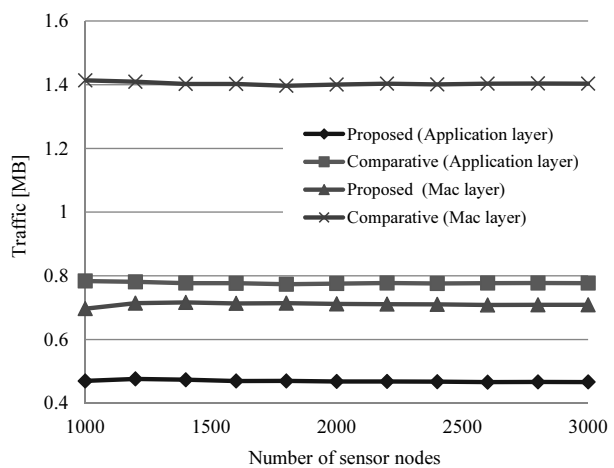Fig. 14. Effects of the number of sensor nodes (Number of packets).

Fig. 15. Effects of the number of sensor nodes (Traffic).

network. As the result, the delivery ratio significantly decreases because the corresponding sensor data cannot be sent to the sink. The effectiveness of handling such situations by the procedures described in Subsection 4.6 is shown in Subsection 5.6.

Figure 13 shows that the delay in our proposed method is much shorter than that in the comparative method due to the same reason as that in Fig. 5. As n increases, the delay in both methods slightly decreases. This is because the distance between the destination and the node near the destination in the forwarding area for geo-routing tends to be shorter due to the increase of node density, which results in reduction of the waiting time for geo-routing (shown in Eq. (1)).

Figure 14 shows that the number of packets in our proposed method is much lower than that in the comparative method due to the same reason as that in Fig. 6. The number of packets in both methods is rarely affected by the number of nodes. In our proposed method, the number of packets slightly decreases as the number of sensor nodes decreases except for the case of Proposed (Movement). Proposed (Movement) gives slightly larger number of packets when the number of sensor nodes is 1000. This is
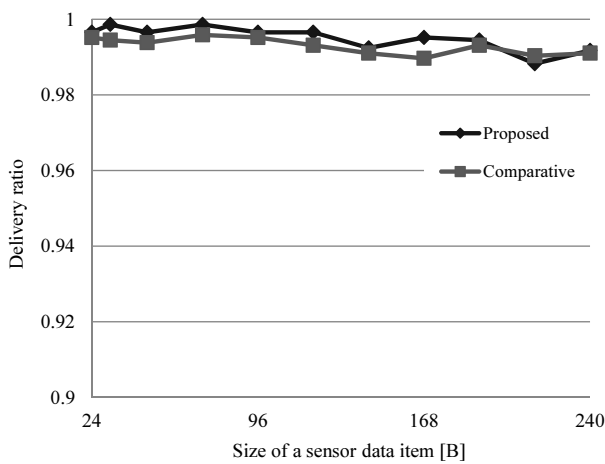
Fig. 16. Effects of size of a sensor data item (Delivery ratio).
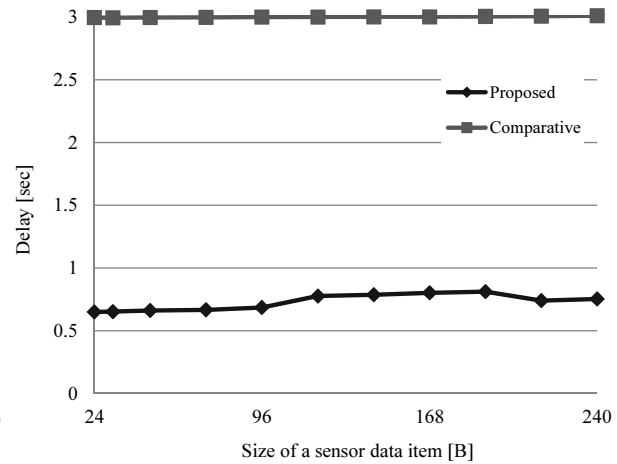

Fig. 17. Effects of size of a sensor data item (Delay).

because when the number of sensor nodes is small, the average distance between the sensing point and the nearest node from the sensing point becomes long, and thus, the mobile agent moves more frequently between sensor nodes.

Figure 15 shows that the traffic in our proposed method is much less than that in the comparative method due to the same reason as that in Fig. 7. The traffic in both methods is rarely affected by the number of nodes. This is because only one node sends the sensor data for each sensing point regardless of the number of sensor nodes.

5.5. Effects of the size of a sensor data

We examine the effects of D, the size of a sensor data item. Figures 16 through 19 show the simulation results. In these graphs, the horizontal axes indicate D, and the vertical axes indicate the delivery ratio in Fig. 16, the delay in Fig. 17, the number of packets in Fig. 18, and the traffic in Fig. 19.

Figure 16 shows that the delivery ratio in both methods is very high regardless of D. As the size of sensor data increases, the delivery ratio slightly decreases in both methods because packet losses occur a bit more frequently. However, the impact of this is small because the size of sensor data is still small. (In this paper, we assume that the size of a sensor data item is small, such as with temperature data.)

Figure 17 shows that the delay in our proposed method is much lower than that in the comparative method. The delay in our proposed method slightly increases as the size of sensor data increases, whereas it is almost constant in the comparative method. This is because the time for sending sensor data along the forwarding tree increases as the size of sensor data becomes larger. In our proposed method, because the sensor data are aggregated, the effect is greater than in the comparative method.

Figure 18 shows that the number of packets in our proposed method is much lower than that in the comparative method. The number of packets in both methods is rarely affected by the size of sensor data, which means that packet losses were rare in this simulation setting.

Figure 19 indicates that as the size of sensor data increases, the traffic in both methods also increases. Here, the increment in traffic in our proposed method is larger than that in the comparative method. This is because in the comparative method sensor data are sent directly to the sink (along the shortest paths), while in our proposed method the sensor data are sent along the forwarding tree (not the shortest paths), i.e., the path length is longer. Since our proposed method does not issue request messages and it aggregates sensor data, the size of sensor data is a dominant factor in traffic.
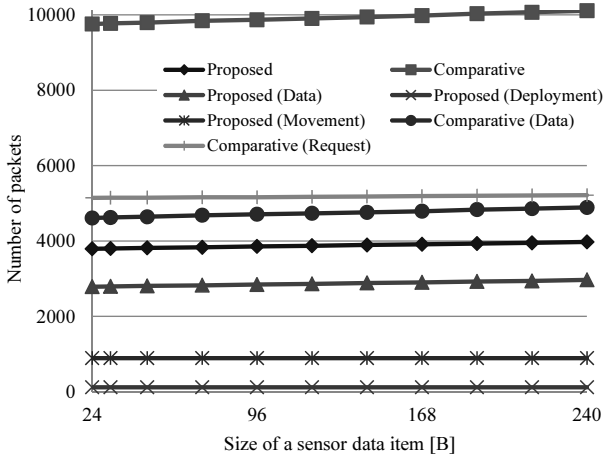
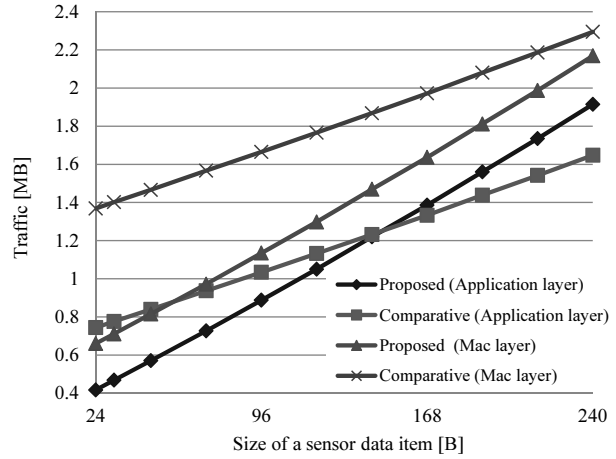Fig. 18. Effects of size of a sensor data item (Number of packets).

Fig. 19. Effects of size of a sensor data item (Traffic).

5.6. Effects of node failure

Finally, we evaluate the effectiveness of applying mechanism of redeployment of mobile agents and retransmission of sensor data described in Subsection 4.6. Each sensor node moves according to the random waypoint mobility model with a home area [2]. Specifically, it selects a random destination in the sub-area that it is assigned to (home area) with the probability of 90% or a destination in the entire sensing area with 10%, and moves at a constant speed uniformly determined from 1 [m/sec] to 2 [m/sec]. It stops at the destination for 60 [sec]. Additionally, each sensor node disappears from the network with $F\%$ every 50 [sec], and the disappeared nodes return to the network after 100 [sec]. The sink and sensor nodes communicate using the IEEE 802.11a protocol. In our proposed method with the mechanism in Subsection 4.6 (extended method), when mobile agents and the sink cannot receive sensor data from any of their children within 2 [sec] ($\delta = 2$), they redeploy the missing mobile agents and request them to retransmit the sensor data. For retransmission of sensor data in the comparative method, the sink sends request messages at intervals of 0.1 [sec] to the sensing points whose sensor data the sink cannot receive within 2 [sec]. We evaluated the extended method, the comparative method, and our proposed method without the mechanism of retransmission of sensor data and redeployment of mobile agents (basic method).

Figures 20 through 23 show the simulation results. In these graphs, the horizontal axes indicate F (ratio of node failure), and the vertical axes indicate the delivery ratio in Fig. 20, the delay in Fig. 21, the number of packets in Fig. 22, and the traffic in Fig. 23.

Figure 20 shows that the delivery ratio in the extended and comparative methods is very high regardless of F. With the basic method, however, as the ratio of node failure increases, the delivery ratio drastically decreases because in this method the sink only initially deploys mobile agents, and data are not gathered until the end of the simulation after a mobile agent disappears (the sensor node on which a mobile agent operates fails). In the extended method, the delivery ratio does not decrease because when a mobile agent disappears, its parent in the forwarding tree redeploys it and the redeployed agent retransmits the sensor data. In the comparative method, the ratio of node failure does not affect the delivery ratio because the comparative method does not use mobile agents.

Figure 21 shows that as the ratio of node failure increases, the delay increases in the extended method because the ratio of redeployment of mobile agents and the retransmission of sensor data increases, and
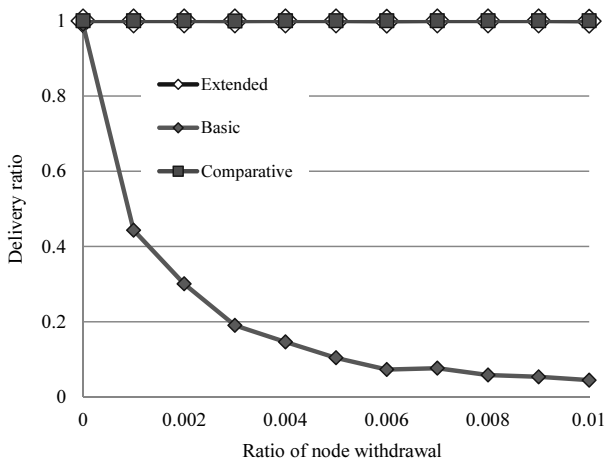
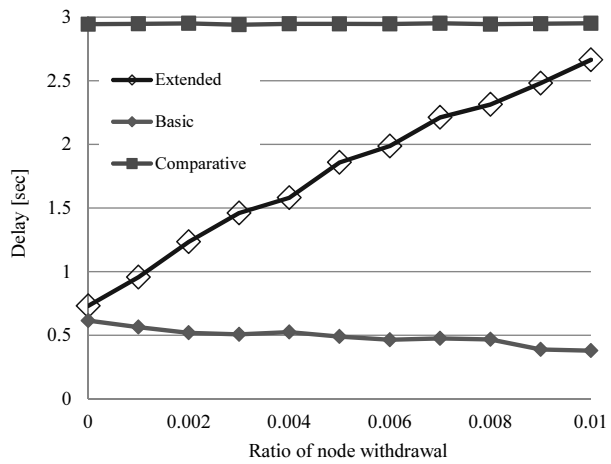Fig. 20. Effects of ratio of node withdrawal (Delivery ratio).

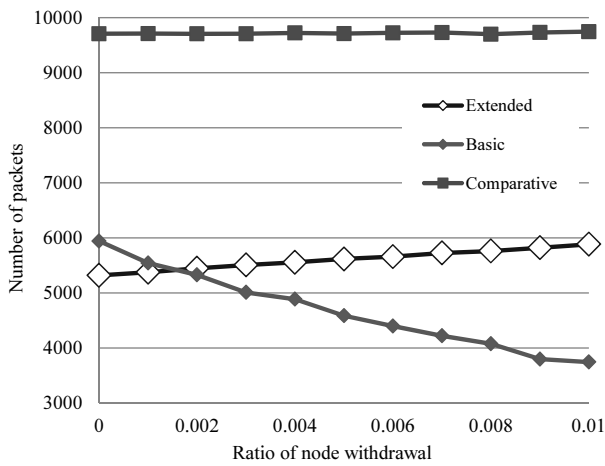Fig. 21. Effects of ratio of node withdrawal (Delay).

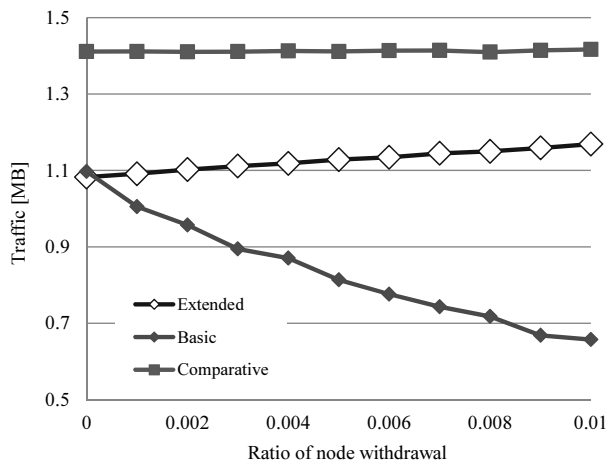Fig. 22. Effects of ratio of node withdrawal (Number of packets).

Fig. 23. Effects of ratio of node withdrawal (Traffic).

the time for these procedures adds to the delay. In the basic method, the ratio of node failure does not affect the delay because the basic method does not retransmit sensor data. The ratio of node failure also does not affect the delay in the comparative method because this method does not use mobile agents, thus, retransmission of sensor data does not happen very often.

Figures 22 and 23 show that as the ratio of node failure increases, the number of packets and the traffic increase in the extended method because the ratio of redeployment of mobile agents and retransmission of sensor data increases, and the messages for these procedures increase. In the basic method, as the ratio of node failure increases, the number of packets and the traffic decrease because the disappeared agents and their ancestors in the forwarding tree do not transmit their sensor data after the mobile agents disappear. The result for the comparative method shows the same delay tendency.

6. Conclusions

In this paper we propose a data gathering method using mobile agents in dense MWSNs. In this method, the sink deploys mobile agents on sensor nodes located near from the sensing points, and the mobile agents autonomously send sensor data to the sink at every sensing time. Our proposed method can reduce the traffic for sensor data gathering since it does not issue any request messages for data gathering and mobile agents can aggregate the sensor data while relaying the data. The results of the simulation experiments show that our proposed method can gather sensor data in short time and with small traffic.

In this paper, for the purpose of simplicity, we do not assume sensing errors (i.e., sensors generate incorrect data). However, in real environments, sensing errors do occur. Therefore, we plan to extend our proposed method to support sensing errors. A possible approach is that neighboring mobile agents exchange their sensor data to check the accuracy before sending the readings to the sink.

Acknowledgment

References

[1]   M. Abolhasan, T. Wysocki and E. Dutkiewicz, A review of routing protocols for mobile ad hoc networks, *Ad Hoc Networks* **2**(1) (2004), 1–22.
[2]   T. Camp, J. Boleng and V. Davies, A survey of mobility models for ad hoc network research, *Wireless Communications and Mobile Computing* **2**(5) (2002), 483–502.
[3]   A.T. Campbell, S.B. Eisenman, N.D. Lane, E. Miluzzo, R.A.Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor and G. Ahn, The rise of people-centric sensing, *IEEE Internet Computing* **12**(4) (2008), 12–21.
[4]   J. Carle and D. Simplot, Energy-efficient area monitoring for sensor networks, *IEEE Computer* **37**(2) (2004), 40–46.
[5]   B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, Span: An energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks, *ACM Wireless Networks* **8**(5) (2002), 481–494.
[6]   M. Chen, T. Kwon, Y. Yuan and V.C.M. Leung, Mobile agent based wireless sensor networks, *JCP* **1**(1) (2006), 14–21.
[7]   B. Godfrey and D. Ratajczak, Naps: Scalable, robust topology management in wireless ad hoc networks, In Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2004), 2004, pp. 443–451.
[8]   J. Haillot and F. Guidec, A protocol for content-based communication in disconnected mobile ad hoc networks, *Mobile Information Systems* **6**(2) (2010), 123–154.
[9]   A.M. Hanashi, I. Awan and M.E. Woodward, Performance evaluation with different mobility models for dynamic probabilistic flooding in MANETs, *Mobile Information Systems* **5**(1) (2009), 65–80.
[10] W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, In Proceedings of Hawaii International Conference on System Sciences (HICSS-33), 2000, pp. 1–10.
[11] M. Heissenbüttel, T. Braun, T. Bernoulli and M. Wälchli, BLR: Beacon-less routing algorithm for mobile ad hoc networks, *Computer Communications* **27**(11) (2004), 1076–1086.
[12] Y.S. Lee, J.W. Park and L. Barolli, A localization algorithm based on AOA for ad-hoc sensor networks, *Mobile Information Systems* **8**(1) (2012), 61–72.
[13] S. Reddy, D. Estrin, M.H. Hansen and M.B. Srivastava, Examining micro-payments for participatory sensing data collections, In Proceedings of ACM International Conference on Ubiquitous Computing (UbiComp 2010), 2010, pp. 33–36.
[14] S. Reddy, V. Samanta, J. Burke, D. Estrin, M.H. Hansen and M.B. Srivastava, MobiSense – Mobile network services for coordinated participatory sensing, In Proceedings of International Symposium on Autonomous Decentralized Systems (ISADS 2009), 2009, pp. 231–236.
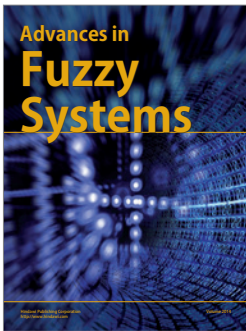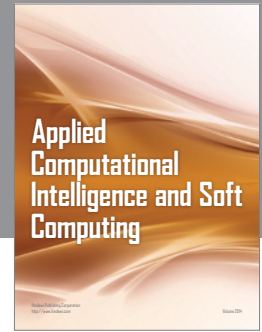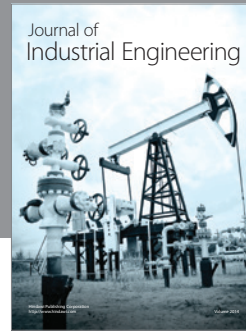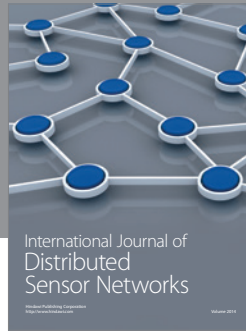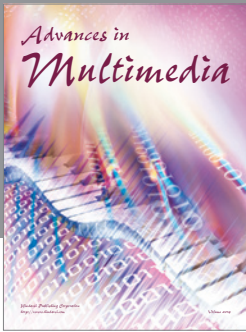[15] Scenargie Base Simulator, Space-Time Engineering, http://www.spacetime-eng.com.

[16] J. Shi, R. Zhang, Y. Liu and Y. Zhang, PriSense: Privacy-preserving data aggregation in people-centric urban sensing systems, In Proceedings of IEEE Infocom, 2010, pp. 758–766.
[17] B. Wang, Coverage problems in sensor networks: A survey, *ACM Computing Surveys* **43**(4) (2011), 32:1–32:53.
[18] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, In Proceedings of ACM Conference on Embedded Networked Sensor Systems (Sensys 2003), 2003, pp. 28–39.
[19] T. Yan, T. He and J.A. Stankovi, Differentiated surveillance for sensor networks, In Proceedings of ACM Conference on Embedded Networked Sensor Systems (Sensys 2003), 2003, pp. 51–62.
[20] T. Yashir, A new paradigm of V2V communication services using nomadic agent, In Proceedings of International Workshop on Vehicle-to-Vehicle Communications (V2VCOM 2006), 2006, pp. 1–6.
[21] Y. Xu, J.S. Heidemann and D. Estri, Geography-informed energy conservation for ad hoc routing, In Proceeding of Annual International Conference on Mobile Computing and Networking (MobiCom 2001), 2001, pp. 70–84.
[22] J. Yick, B. Mukherjee and D. Ghosa, Wireless sensor network survey, *Computer Networks* **52**(12) (2008), 2292–2330.

Keisuke Goto Keisuke Goto received the B.E. degree from Kyoto Institute of Technology and the M.I.S. degree from Osaka University, Japan, in 2010 and 2012, respectively. He is currently a research student at the Graduate school of Information Science and Technology of Osaka University. His research interests include Mobile Wireless Sensor Networks.

Yuya Sasaki Yuya Sasaki received the B.E. and M.I.S. degrees from Osaka University, Japan, in 2009 and 2011, respectively. He is currently pursuing his Ph.D. at the Graduate school of Information Science and Technology of Osaka University. His research interests include Mobile Ad Hoc Networks and Vehicular Ad Hoc Networks.

Takahiro Hara Takahiro Hara received the B.E., M.E., and Dr. E. degrees in Information Systems Engineering from Osaka University, Osaka, Japan, in 1995, 1997, and 2000, respectively. Currently, he is an Associate Professor of the Department of Multimedia Engineering, Osaka University. He has published more than 300 international Journal and conference papers in the areas of databases, mobile computing, peer-to-peer systems, WWW, and wireless networking. He served and is serving as a Program Chair of IEEE International Conferences on Mobile Data Management (MDM'06 and 10) and Advanced Information Networking and Applications (AINA'09), and IEEE International Symposium on Reliable Distributed Systems (SRDS'12). He guest edited IEEE Journal on Selected Areas in Communications, Sp. Issues on Peer-to-Peer Communications and Applications. His research interests include distributed databases, peer-to-peer systems, mobile networks, and mobile computing systems. He is a senior member of IEEE and ACM and a member of three other learned societies.

Shojiro Nishio Shojiro Nishio received his B.E., M.E., and Ph.D. degrees from Kyoto University in Japan, in 1975, 1977, and 1980, respectively. He has been a full professor at Osaka University since August 1992. He served as a Vice President and Trustee of Osaka University from August 2007 to August 2011. He also acted as the Program Director in the Area of Information and Networking, Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan from April 2001 to March 2008. His research interests include database systems and multimedia systems for advanced networks such as broadband networks and mobile computing environment. Dr. Nishio has co-authored or co-edited more than 55 books, and authored or co-authored more than 560 refereed journal or conference papers. He served as the Program Committee Co-Chairs for several international conferences including DOOD 1989, VLDB 1995, and IEEE ICDE 2005. He has served and is currently serving as an editor of several international journals including IEEE Trans. on Knowledge and Data Engineering, VLDB Journal, ACM Trans. on Internet Technology, and Data & Knowledge Engineering. He is also a fellow of IEICE and IPSJ, and is a member of six learned societies, including ACM and IEEE.