A Mobile Agents Control Scheme for Multiple Sinks 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, Yamadaoka 1-5, Suita-shi, Osaka, Japan
{goto.keisuke,sasaki.yuya,hara,nishio}@ist.osaka-u.ac.jp

Abstract. In Mobile Wireless Sensor Networks (MWSNs) where mobile sensor nodes densely exist, it is desirable to gather sensor data from the minimum number of sensor nodes which are necessary to guarantee the sensing coverage in order to reduce communication traffic. In the past, we have proposed a data gathering method using mobile agents in dense MWSNs. However, since this method assumes that only one sink is present in a network, it cannot effectively reduce traffic in environments where multiple sinks exist. In this paper, we propose a mobile agents control scheme which guarantees multiple sinks' coverages and efficiently gathers sensor data. In the proposed method, mobile agents are communalized if their sensing points overlap, and sensor data are aggregated to transmit them to same direction.

Keywords: Mobile wireless sensor networks · Data gathering · Mobile agent

1 Introduction

Recently, *participatory sensing* by ordinary people having mobile sensor nodes such as PDA and smart phones with sensor devices has attracted much attention [3,7]. In mobile wireless sensor networks (MWSNs) for participatory sensing, 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 resource of communication bandwidth. Rather, applications require a certain geographical granularity of sensing in most cases. To reduce the data traffic for data gathering, it is desirable to effectively and reliably gather sensor data so that the geographical granularity required from an application can be guaranteed with the minimum number of sensor nodes.

In [4], we have proposed a data gathering method that efficiently gathers sensor data by using *mobile agents* which control sensor nodes' transmission of

I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 52–65, 2014.
DOI: 10.1007/978-3-319-11569-6_5

sensor data. A mobile agent is an application software that autonomously operates on a sensor node and moves between sensor nodes. This method reduces traffic for gathering sensor data since the number of sensor nodes that transmit sensor data is minimized by mobile agents. However, this method cannot effectively reduce traffic in environments where multiple sinks gather sensor data based on different conditions (e.g., gathering cycle and geographical granularity) each other, because this method basically assumes only one sink gathering sensor data, and thus when multiple sinks are present, each sink respectively deploys mobile agents and separately gathers sensor data.

In this paper, we propose a mobile agents control scheme to guarantee the sensing coverages designated by multiple sinks and efficiently gather the sensor data. In the proposed method, mobile agents are communalized if sensing points overlap, and sensor data are aggregated to transmit to some sinks which locate in the same direction. We verify that the proposed method achieves small traffic while keeping high delivery ratio, through extensive simulation experiments.

The remainder of this paper is organized as follows. In Sect. 2, we introduce related work. In Sect. 3, we describe assumptions in this paper. In Sect. 4, we explain the details of our proposed method. In Sect. 5, we show the results of the simulation experiments. Finally, in Sect. 6, we summarize this paper.

2 Related Work

First, we introduce existing researches for data gathering in WSNs. In [9], the authors proposed a hierarchical data gathering method. In this method, sensor nodes are hierarchically arranged, where a sensor node in a lower level sends the sensor data to a node in a higher level, and then the sensor node in the highest level sends the aggregated sensor data to the sink. This method can reduce the traffic for data gathering since nodes in higher levels aggregate and compress the sensor data. In [1], the authors proposed a grid-based routing protocol for extending network lifetime. A master node is elected from sensor nodes in each grid that is predefined by the sink. Each master node monitors node density in its handling grid and share the information each other. Other sensor nodes send their sensor data to the master node in each grid, and then the master node transmits the sensor data to the sink through dense areas (grids). This helps saving non-master nodes energy and maintaining network connectivity, and thereby extending network lifetime. The existing studies presented above do not assume the movements of sensor nodes, and thus, cannot handle the change of network topology.

Next, we introduce existing researches for data routing in mobile ad hoc networks (MANETs). In [11], the author assumed location-based services, and proposed a data gathering and disseminating method in MANETs. This 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 the location. 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. In [10], the authors proposed a group communication algorithm in MANETs. If group nodes receive a join request packet from source nodes, they reply their location and velocity. When a source node sends a data packet to its group nodes, it predicts their mobility and constructs a multicast tree based on the Euclidean Steiner tree [6]. The group communication algorithm efficiently transmits messages through the multicast trees and reduces traffic for location update by predicting mobility of group nodes. This algorithm is not efficient for data gathering because each source node constructs a multicast tree and messages to the same destination are individually transmitted.

Finally, we introduce an existing research for mobile P2P systems. In [8], the authors proposed an efficient data access method for location-based data in an environment where mobile nodes densely exist. In this method, nodes exchange their having data each other so that the data are being held by a node located within a half of the node's communication range from the geographical point corresponding to the data. Hence, it is guaranteed that any node can access data by sending a packet to the geographical point corresponding to the data. This study is different from our work that aims at gathering sensor data necessary to guarantee the geographical granularity of sensing. However, the method keeping data close to its corresponding point uses an idea similar to the method keeping mobile agents close to sensing points in this paper.

3 Assumptions

3.1 System Environment

We assume the use of dense MWSNs constructed of mobile sensor nodes that are equipped with a radio communication facility and periodically observe the physical phenomena (e.g., sound, temperature, and light). Communication infrastructures are not available in the area where the sensor nodes exist so they communicate with each other using multi-hop radio communication. There are number of sinks and they periodically monitor the sensing area while guaranteeing the geographical granularity of the sensing, according to the requirement from an application. We call the combination of a sink's position and the requirement from an application a *data gathering condition*.

The entire area is assumed to be a two-dimensional plane. Application $app_i(i = 1, 2, \cdots)$ specifies its sensing condition (Table 1). Sensing area A_i is a rectangular area whose horizontal to vertical ratio is M_i:N_i (M_i and N_i are positive integers), and the requirement of the geographical granularity of sensing is a ${k_i}^2 \cdot M_i \cdot N_i$ integer. Here, multiple sinks and requirements from the applications exist in the network and the sinks receive a requirement from the applications. A sink that receives the requirement divides its sensing area into $k_i \cdot M_i \times k_i \cdot N_i$ lattice-shaped sub-areas and determines the center point of each sub-area as a sensing point, which is the data gathering target (Fig. 1). The sink gathers sensor data from sensor nodes located within distance s from each sensing point at the time of $P_i + l_iT_i(l_i = 0, \cdots, L_i - 1)$ where P_i is app_i's start time

Table 1. App_i's sensing condition

Contents	Symbols
Sensing area	A_i
Geographical granularity of sensing	k_i
Start time	P_i
Number of gathering	L_i
Gathering cycle	T_i

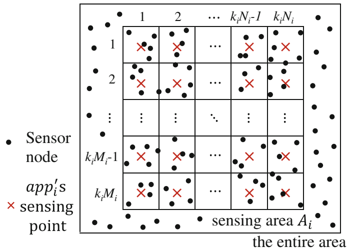


Fig. 1. Sensing area and sensing points

of data gathering and T_i is app_i's gathering cycle. The sink's location o_i is in the sensing area.

As previously mentioned, we assume the use of MWSNs constructed of mobile sensor nodes held by ordinary people. 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. Each sensor node freely moves throughout the entire area, while the sinks are stationary. We assume that sensor nodes can reliably sense location data within a radius s of their position. Since the number of mobile nodes is very large, there are multiple sensor nodes that can cover each geographical point within the entire sensing area.

3.2 Geo-routing

Sensor nodes adopt a geo-routing protocol that is based on that proposed in [5] to transmit a message to the specified destination as a location (not a node). In this protocol, the nodes perform a transmission process using the information on the positions of the transmitter and the destination, which is specified in the packet header. In particular, the transmitter writes the information on the positions of the destination and itself into the packet header of the message, and broadcasts it to its neighboring nodes. Each node that receives this message judges whether it locates within the forwarding area which is determined based on the positions of the transmitter, the destination, and the communication range. Any node in the forwarding area is closer to the destination than the transmitter and can communicate directly to all the nodes in that area. The node within the forwarding area sets the waiting time (the node closer to the destination sets a shorter waiting time), and then it forwards the message after the waiting time elapses unless it detects that the message was sent by another node during the waiting time. By repeating this procedure, the message is forwarded to the nodes that are closer to the destination. If the transmitter node exists 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 a result, the nearest node to the destination (that has sent the ACK) finds that the nearest one is itself.

4 Mobile Agents Control Based on Data Gathering Conditions of Multiple Sinks

4.1 Outline of the Proposed Method

In the proposed method, a sink that has initially received a request from an application app_i deploys mobile agents into $k_i \cdot M_i \times k_i \cdot N_i$ sensing points that are determined as described in Sect. 4.2. If a sensor node on which a mobile agent runs moves away from the sensing point having been deployed, the mobile agent moves from the sensor node to another node that is the closest to the point, according to the method described in Sect. 4.3. At each sensing time, the mobile agents send sensor data generated by sensor nodes on which they run to the sink according to the method described in Sect. 4.4.

4.2 Deployment of Mobile Agents

The proposed method reduces the transmission traffic for sensor data and the movement of the mobile agents by communalizing the mobile agents that handle near-by sensing points. Algorithm 1 shows the procedures to deploy mobile agents. In this pseudo code, $Address(x)$ denotes the sensing point in a neighboring sub-area in the direction of x.

Algorithm 1. Deploying mobile agents

```
1: Procedure for sink S receiving the requirement from application app
2: A ← information of sensing condition of app
3: send A to the sensing point in the sub-area where S exists
4: Procedure for sensor node receiving A
5: if the node has no agent data then
6:     boots the mobile agent
7: end if
8: C ← C ∪ A
9: for ∀ C_i ∈ C do
10:     dist_i ← the distance between the sensing points of C_i and A
11: end for
12: broadcast a message with the furthest distance dist_max
13: if A is sent from S or upward, right, or left sensing point then
14:     send A to Address(DOWN)
15: end if
16: if A is sent from S or right sensing point then
17:     send A to Address(LEFT)
18: end if
19: if A is sent from S or downward, right, or left sensing point then
20:     send A to Address(UP)
21: end if
22: if A is sent from S or left sensing point then
23:     send A to Address(RIGHT)
24: end if
25: Procedure for sensor node receiving the messages with the distance from other nodes
26: if the node has not broadcast a message then
27:     break
28: else if the dist_max in the received messages is smaller than the dist_max then
29:     C ← C − A
30: end if
```

First, a sink that receives the requirement from application *app* creates the agent data that is needed to boot the mobile agent (line 2). Then, the sink sends the agent data to the sensing point in the sub-area where the sink exists by using geo-routing (line 3). The sensor node closest to the sensing point receives the agent data and boots a mobile agent. If a mobile agent has already been running near the sensing point, it receives the agent data instead of the sensor node closest to the sensing point (lines 5 to 8). In this way, a communalized mobile agent stores several agent data, and if one of the data gathering processes completes, its role changes on the fly. More concretely, if the distances among sensing points are smaller than a threshold $\alpha(\alpha < s)$, the mobile agent charges into these sensing points.

Additionally, mobile agents in the proposed method have their territories to avoid that multiple nodes receive agent data and run the same agent. More concretely when a mobile agent receives an agent data, it broadcasts a message with the information on the furthest distance among the sensing points of its stored agent data and the received one to its neighboring nodes (lines 9 to 12). If multiple mobile agents having received the agent data exist, they receive the above messages from each other. Only the mobile agent with the minimum distance stores the new agent data and the other mobile agents discard it (lines 26 to 30).

Figure 2 shows an example where two mobile agents receive agent data. There are two mobile agents MA_1 and MA_2, two sensing points p_1 and p_2, and a destination of agent data q_1. MA_1 and MA_2 respectively handle into p_1 and p_2, and they receive the agent data whose destination is q_1 because the distance among p_1, p_2, and q_1 are smaller than α. MA_1 broadcasts a message with the information on the distance between p_1 and q_1, and, MA_2 broadcasts the distance between p_2 and q_1. They receive these messages from each other, and MA_1 stores the agent data and MA_2 discards it because the minimum distance is the distance between p_1 and q_1.

Moreover, a mobile agent that is newly booted or receives the agent data retransmits it to the sensing points in some of the sub-areas based on its existing sub-area (lines 13 to 24). The agent data transmission by the mobile agent is propagated in the crosswise direction, followed by the lengthwise direction. By repeating these procedures, the sink deploys mobile agents near all its sensing points (i.e., within the circle whose center is a sensing point and radius is the sensing range s). In this procedure, the sink and the mobile agents construct their

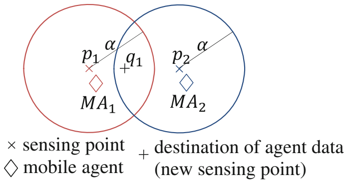


Fig. 2. Example that mobile agent exchange their territories

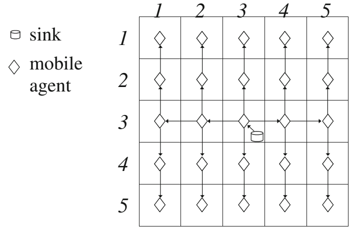


Fig. 3. Example of forwarding agent data between mobile agents

parent-child relationship with each other according to the agent data transmission (transmitter-receiver corresponds to parent-child). We call the tree structure consisting of the parent-child relationships a *forwarding tree* (e.g., Fig. 3). As a result, the number of mobile agents and thus the communication traffic for movement of the agents decrease.

4.3 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 cover that point. Additionally, it may not be able to receive the sensor data sent from its child nodes on the forwarding tree. Therefore, in the proposed method, a mobile agent moves from the current sensor node to another node that is closest to the sensing point to avoid such a situation.

In particular, a mobile agent starts moving when the distance between the sensing point and itself becomes longer than threshold β. β is a system parameter that is set as a constant value smaller than $r/2$ and s, which can guarantee that a sensor node on which a mobile agent operates can communicate with all the sensor nodes located near (within $r/2$) 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 its neighbor nodes within $r/2$ from the sensing point. The sensor node located closest to the sensing point sends an ACK and boots a mobile agent at first as in geo-routing. The other sensor nodes cancel sending own ACK and the original mobile agent stops its operation because they can detect the first ACK.

If a mobile agent handles multiple data gathering processes (i.e., it has multiple sensing points), the node must stay within less than s from any sensing points. The mobile agent start moving when the distance between any sensing point and itself becomes longer than β. The mobile agent broadcasts a message containing the all agent data which the node has. The mobile agents move to the middle point among these sensing points. However, the sensor node that receives the agent data may be more than β away from some sensing points. In such cases, the sensor node separates these agent data and individually move them to their sensing points.

4.4 Transmission of Sensor Data

Our proposed method can reduce the traffic for sending sensor data since the mobile agents send the aggregated sensor data to the sinks through the forwarding trees. Specifically, the sensor nodes on which multiple mobile agents operate aggregate sensor data for their multiple parents. The sensor nodes first group their parents based on the directions of their parents. The sensor nodes send a sensor data message to one of the parents in each group, and the sensor data for the others parents are stored as *additional data*. The additional data consists of a set of *destination groups*, each of which is composed by the destination address, and a set of sensor data that are addressed to the destination.

Algorithm 2 shows the transmission procedures. In this pseudo code, *Destination*(x) denotes a position information on destination group x. Algorithm 2 includes procedures for grouping and sending a data message (Algorithms 3 and 4, respectively). At every sensing time, sensor nodes on which mobile agents operate transmit their sensor data to the sinks of the agents. First, the sensor nodes on which mobile agents operate get sensor readings (line 2). The sensor readings are valid among communalized mobile agents because distances from their sensing points are kept smaller than the sensing range. Next, the sensor nodes on which multiple mobile agents operate group the mobile agents according to their parent directions in order to aggregate the sensor (line 3 and Algorithm 3). Then, the sensor nodes send a message including the sensor data (line 5).

Each sensor node that receives this message, stores all sensor data contained in the message and sends a message in the same way (lines 8 to 13). If the attached additional data contains a destination group whose destination address is different from any of sensing points of the agents running on the receiver node, the node composes a message consisting of sensor data for the destination group and sends it to the destination address. (lines 14 to 19). When sensor nodes on which mobile agents operate in the sub-areas where the sinks exist complete collecting all the necessary sensor data from all their child nodes, they send a message containing all the received and its own sensor data to the sink (lines 20 to 28).

Algorithm 4 shows the procedures for sending sensor data. In this pseudo code, *Position*(x) denotes a position information on sensing point of mobile agent x, and *Index*(x) denotes an index for sensor data x indicating a number as an order which x appears in a sensor data message. Before sending a sensor data message, for a certain group, a sensor node checks if it has all sensor data which should be sent from their child nodes with respect to the group (lines 1 to 3). Here, since a leaf node on a forwarding tree has no child node, it can skip this step. If this node has all sensor data to send to its parents, it creates a sensor data message, which contains the sensor data for the corresponding parent (lines 4 to 5), and sensor data for the other parents and the information of sensing points of those parents as additional data (lines 6 to 22). Then, the sensor node sends the messages to the parents by using geo-routing (lines 23 to 24).

By the above procedures, the sink can receive sensor data at all the sensing points from the sensor nodes on which mobile agents operate.

Figure 4 shows an example in which mobile agents send sensor data whose sensing times overlap. In Fig. 4, there are four mobile agents MA_1, MA_2, MA_3, and MA_4, and the parents of MA_1 are MA_2 and MA_3, and a parent of MA_2 is MA_4. MA_1 sends its sensor data SD_1 and additional data for MA_2 and MA_3 to MA_2 because both MA_2 and MA_3 exist in the right direction and MA_2 is closer to the center of MA_1's sensing points. Here, the message has additional data that is the pair of the position information of MA_3 and an index to SD_1. MA_2 receives SD_1 from MA_1, and then sends SD_1 and its sensor data SD_2 to its parent MA_4. MA_2 also independently sends SD_1 to MA_3 by referring to the additional data of the received message.

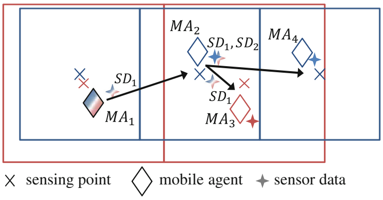


Fig. 4. Example of transmission of sensor data in multiple data gathering

Algorithm 2. Transmitting sensor data

Every time when the sensing cycle of any agent data comes,
get the sensor reading
$G \leftarrow$ Group(all agent data: A)
for G_i(i =LEFT, RIGHT, UP, and DOWN) **do**
  SendData(G_i)
end for

Procedure having received a message containing sensor data and additional data: D **and** L
for $\forall$ $E \in$ sensor data in D **do**
  store E
end for
for G_i(i =LEFT, RIGHT, UP, and DOWN) **do**
  SendData(G_i)
end for
for $\forall$ destination group: $T \in L$ **do**
  if *Destination*(T) is not the sensing point of any of its own agent data **then**
    sensor data: $D^* \leftarrow$ all sensor data containing or indexed in L
    send D^* to *Destination(T)*
  end if
end for
if $G_{\text{SINK}} \neq \phi$ **then**
  for $\forall$ $A \in G_{\text{SINK}}$ **do**
    if A receives sensor data from its all child nodes **then**
      sensor data: $D^* \leftarrow$ sensor data received from A's all child nodes
      send D^* to A's sink
      $G_{\text{SINK}} \leftarrow G_{\text{SINK}} - A$
    end if
  end for
end if

Algorithm 3. Group(A)

for $\forall A_i \in A$ **do**
  $P \leftarrow A_i$'s parent direction
  if A_i's parent is its sink **then**
    $G_{\text{SINK}} \leftarrow G_{\text{SINK}} \cup A_i$
  else if P = UP **then**
    $G_{\text{UP}} \leftarrow G_{\text{UP}} \cup A_i$
  else if P = RIGHT **then**
    $G_{\text{RIGHT}} \leftarrow G_{\text{RIGHT}} \cup A_i$
  else if P = DOWN **then**
    $G_{\text{DOWN}} \leftarrow G_{\text{DOWN}} \cup A_i$
  else if P = LEFT **then**
    $G_{\text{LEFT}} \leftarrow G_{\text{LEFT}} \cup A_i$
  end if
end for

Algorithm 4. SendData (G)

if $G = \phi$ **or** $\exists A \in G$ does not receive sensor data from its child nodes **then**
  return
end if
$A_* \leftarrow$ the agent data $\in G$ whose sensing point is closest to the center of the node's sensing points
sensor data message: $D \leftarrow$ sensor data sensed by the node and A_*'s all descendant nodes
$G \leftarrow G - A_*$
additional data: $L \leftarrow \phi$
if $G \neq \phi$ **then**
  for $\forall$ $A \in G$ **do**
    if If *Position*(A's parent) $\neq$ *Destination*($\forall$ destination group $\in L$) **and** *Position*(A's parent) $\neq$ *Position*(A's parent) **then**
      destination group: $T \leftarrow$ *Position*(A's parent)
      for $\forall$ $E \in$ sensor data received from A's all child nodes **do**
        if E is included in $D \cup L$ **then**
          $T \leftarrow T \cup$ *Index*(E)
        else
          $T \leftarrow T \cup E$
        end if
      end for
      $L \leftarrow L \cup T$
    end if
  end for
end if
send D and L to *Position*(A_*'s parent)
$G \leftarrow \phi$

It should be noted that the traffic produced by these procedures is expected to be smaller than that produced by the procedure in which mobile agents individually sends their sensor data to their parents. This is because some parents of a mobile agent may exist in the same direction, and thus it is more efficient to send the sensor data to multiple parents with one message.

5 Simulation Experiments

In this section, we show the results of simulation experiments regarding the performance evaluation of our proposed method. For the simulation, we used the network simulator, Scenargie 1.6.[1]

5.1 Simulation Model

There are 2,000 mobile sensor nodes $(M_1, \cdots, M_{2000})$ and p sinks $(S_1, \cdots, S_p)$ in a two-dimensional field of 1000 [m]×1000 [m]. $S_i (i = 1, \cdots, p)$ is fixed at the point of (PX_i [m], PY_i [m]) from the left and the bottom edges of the sensing field. Each sensor node moves according to the random waypoint model with a home area [2] where it selects a random direction and a random speed from 0.5 to 1 [m/sec] at intervals of 60 [sec]. Sinks and sensor nodes communicate with IEEE 802.11a whose transmission rate is 6 [Mbps] and communication range r is about 100 [m]. Each sensor node continuously senses the field and the sensing range s is 50 [m]. Application $app_i (i = 1, \cdots, p)$ requires the sink S_i to gather sensor data. For all app_i, P_i and k_i are 4,400 [sec] and 4, respectively. For $app_{2i-1} (i = 1, \cdots, \lfloor p/2 \rfloor)$, L_i, and T_i are 240, and 30 [sec], while for $app_{2i} (i = 1, \cdots, \lceil p/2 \rceil)$, L_i, and T_i are 120, and 60 [sec], respectively. A_i is a rectangle area whose point of (left, bottom) and point of (right, top) are (PL_i [m], PB_i [m]), (PL_i+800 [m], PB_i+800 [m]), respectively. Each sink divides its sensing field into 16 lattice-shaped sub-areas whose size is 200 [m] ×200 [m] and sets the center point of each sub-area as a sensing point.

Each sink deploys a mobile agent at each of its sensing points after 4380 [sec] from the start time of the simulation. The sensing operations start at 4400 [sec]. The sensing times of $S_{2i-1} (i = 1, \cdots \lfloor p/2 \rfloor)$ and that of $S_{2i} (i = 1, \cdots \lceil p/2 \rceil)$ are $4400 + \{30, 60\} m (m = 0, 1, \cdots, \{239, 119\})$ [sec]. The size of an agent data is set as 128 [B], assuming that each sensor node has the source code of mobile agent in advance. The size of a sensor data generated at each sensor node is set as 24 [B]. The size of an address of sensor data in the additional data is set as 1 [B]. Additionally, parameters of our proposed method α and β are respectively set as 48[m] and 49[m], according to the results of our preliminary experiments.

For comparison, we also evaluate the performances of the proposed method without aggregating sensor data (*nonAggregation*) and our previous method where each sink individually gathers sensor data without communalization of mobile agents (*comparative*).

[1] Scenargie 1.6 Base Simulator revision 10864, Space-Time Engineering, http://www.spacetime-eng.com/.

Table 2. Message size

Object	Message name	Size [B]
Deploying a mobile agent	Deployment	256
Moving a mobile agent	Movement	$128 + 128 \cdot i$
	Broadcast distance	96
Sending sensor data	Sensor data	$64 + 32 \cdot j$
	(+Additional data)	$(+\sum_{u=1}^{U}(24 + 32 \cdot k_u + 1 \cdot l_u))$
Common	ACK	96

In the above simulation model, we performed experiments in which the initial position of each mobile sensor nodes was randomly determined where there was the same number of sensor nodes in each of 200 [m] × 200 [m] rectangle areas (sub-areas). The end of sensing operations of all sinks are 8000 [sec], and we evaluated the following three criteria.

1. Traffic: The traffic is defined as the summation of the size of all packets sent by the sink and all sensor nodes during the simulation. Table 2 shows messages used in our method and the comparative method, and their sizes at the Mac layer. In this table, i denotes the number of agent data, j denotes the number of sensor data aggregated, and k_u and l_u denote the number of sensor data containing and indexed on destination group $u(u = 1, \cdots, U)$ in an additional data, respectively.
2. Delivery ratio: The delivery ratio is defined as the ratio of the number of sensor data sent to the sinks in the data gathering processes to the total number of sensor data that should be acquired during the simulation.
3. 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.

5.2 Effects of Number of Sinks

First, we examine the effects of the number of sinks p. Table 3 shows sinks' positions and sensing areas in this experiment. Figure 5 shows the simulation results. In these graphs, the horizontal axes indicate p, and the vertical axes indicate the traffic in Fig. 5(a), the delivery ratio in Fig. 5(b), and the delay in Fig. 5(c), respectively.

Figure 5(a) shows that the traffic in the proposed method is smaller than nonAggregation and the comparative method. This is because in the proposed method, the number of packets decreases by collectively sending aggregated sensor data to multiple sinks locating in the same direction.

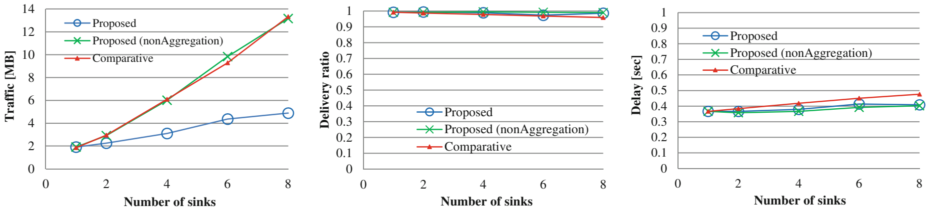
Figure 5(b) shows that the delivery ratio in all methods is high. In particular, our method always achieves almost perfect delivery ratio (i.e., 1). This shows that all sensor data which all sinks received are valid, i.e., every mobile agent

Table 3. Sink's position and sensing area in Sec. 5.2

ID	PX_i [m], PY_i [m]	PL_i [m], PB_i [m]
S_1	(140, 140)	(100, 100)
S_2	(180, 140)	(110, 100)
S_3	(220, 140)	(120, 100)
S_4	(260, 140)	(130, 100)
S_5	(140, 220)	(100, 120)
S_6	(180, 220)	(110, 120)
S_7	(220, 220)	(120, 120)
S_8	(260, 220)	(130, 120)

Table 4. Sink's position and sensing field in Sec. 5.3

ID	PX_i [m], PY_i [m]	PL_i [m], PB_i [m]
S_1	(140, 140)	(0, 100)
S_2	(180, 140)	(d, 100)
S_3	(220, 140)	(2d, 100)
S_4	(260, 140)	(3d, 100)



(a) Traffic (b) Delivery ratio (c) Delay

Fig. 5. Effects of number of sinks

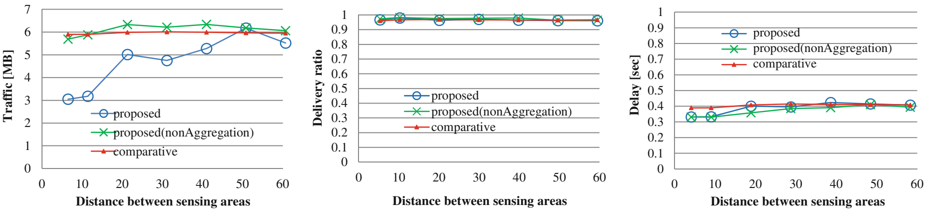
always stays within its valid range of sensing. However, the delivery ratio in the comparative method slightly decreases as the number of sinks increases. This is because packet losses occur due to increase of traffic.

Figure 5(c) shows that the proposed method and nonAggregation can gather sensor data in shorter time than the comparative method when p is larger than 1. In the proposed method, the sensor data are aggregated by communalized mobile agents and are collectively sent to the multiple sinks. On the other hand, in the comparative method, the sensor data are individually sent to the multiple sinks. This increases the number of packets, resulting in congestion of network bandwidth, and thus transmission of packets delays in the MAC layer. Though nonAggregation also increases the number of packets, it suppresses the congestion by delaying the timings of sending packets in the application layer.

5.3 Effects of Distance Between Sensing Areas

Next, we examine the effects of distance between sensing areas d. Table 4 shows sinks' positions and sensing areas in this experiment, where we fix the number of sinks as 4. Figure 6 shows the simulation results. In these graphs, the horizontal axes indicate d, and the vertical axes indicate the traffic in Fig. 6(a), the delivery ratio in Fig. 6(b), and the delay in Fig. 6(c), respectively.

Figure 6(a) shows that the traffic in the proposed method is always smaller than the comparative method except that $d = 50$. This shows that because it is



(a) Traffic (b) Delivery ratio (c) Delay

Fig. 6. Effects of distance between sensing areas

effective to aggregate sensor data especially when the distance between sensing points is small. As d increases, the traffic and the number of packets slightly increase in the proposed method. This is because, the chance of communalizing mobile agents decreases due to increase of distance between sensing points. When $d = 50$, no mobile agents are communalized because all sensing points are longer than the communalizing threshold α away from each other. As a result, traffic in the proposed method and nonAggregation is slightly larger than the comparative method when $d = 50$ due to extra messages for movement of mobile agents.

Figure 6(b) shows that delivery ratio in all methods is high as we discussed above.

Figure 6(c) shows that the proposed method can gather sensor data in shorter time than the comparative method when d is 10 because it is effective to aggregate sensor data when the distance between sensing points is small.

6 Conclusion

In this paper, we proposed an agent control method that guarantees the coverages of multiple sinks and efficiently gathers sensor data. In the proposed method, mobile agents are communalized if the sensor node on which a mobile agent operates can sense multiple sensing points.

Simulation experiments show that the proposed method decreases traffic by collectively sending aggregated sensor data to multiple sinks locating in the same direction and keep high delivery ratio even when the number of sinks is high.

Our proposed method may not work well in environments where there are obstacles such as buildings because few or no sensor nodes may exist close to some sensing points. We plan to extend our proposed method to approximate sensor readings at such sensing points.

Acknowledgment. This research is partially supported by the Grant-in-Aid for Scientific Research (S)(21220002), and (B)(24300037) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. Banimelhem, O., Khasawneh, S.: GMCAR: grid-based multipath with congestion avoidance routing protocol in wireless sensor networks. Ad Hoc Netw. **10**(7), 1346–1361 (2012)
2. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wireless Commun. Mob. Comput. **2**(5), 483–502 (2002)
3. Cha, S., Talipov, E., Cha, H.: Data delivery scheme for intermittently connected mobile sensor networks. Comput. Commun. **36**(5), 504–519 (2013)
4. Goto, K., Sasaki, Y., Hara, T., Nishio, S.: Data gathering using mobile agents for reducing traffic in dense mobile wireless sensor networks. In: Proceedings of MoMM 2011, pp. 58–65 (2011)
5. Heissenbüttel, M., Braun, T., Bernoulli, T., Wälchli, M.: BLR: beacon-less routing algorithm for mobile ad hoc networks. Comput. Commun. **27**(11), 1076–1086 (2004)
6. Hwang, F.K., Richards, D.S., Winter, P.: The Steiner Tree Problem. North-Holland, Amsterdam (1992). ISBN: 044489098X
7. Jabeur, N., Zeadally, S., Sayed, B.: Mobile social networking applications. Commun. ACM **56**(3), 71–79 (2013)
8. Landsiedel, O., Götz, S., Wehrle, K.: Towards scalable mobility in distributed hash tables. In: Proceedings of P2P 2006, pp. 203–209 (2006)
9. Lu, K.-H., Hwang, S.-F., Su, Y.-Y., Chang, H.-N., Dow, C.-R.: Hierarchical ring-based data gathering for dense wireless sensor networks. Wireless Pers. Commun. **64**(2), 347–367 (2012)
10. Mitra, P., Poellabauer, C.: Efficient group communications in location aware mobile ad-hoc networks. Pervasive Mob. Comput. **8**(2), 229–248 (2012)
11. Yashiro, T.: A new paradigm of V2V communication services using nomadic agent. In: Proceedings of V2VCOM 2006, pp. 1–6 (2006)