JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 23, 1103-1121 (2007)

Data Aggregation of Range Querying for Grid-based Sensor Networks*

TZUNG-SHI CHEN+, YI-SHIANG CHANG+, HUA-WEN TSAI AND CHIH-PING CHU
+*Department of Information and Learning Technology*
National University of Tainan
Tainan, 700 Taiwan
E-mail: chents@mail.nutn.edu.tw
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, 701 Taiwan
E-mail: chucp@csie.ncku.edu.tw

This work proposes efficient data aggregation algorithms for range query in wireless sensor networks. The proposed aggregation and query mechanisms are based on a virtual grid in sensor networks. A sensor node is selected to be a manager, named head, in each grid. The head node detects the generated event, announces event to all heads, and responds the query of mobile users. A user obtains the occurred event information from its grid head. If a user is interested in the event, it issues a query to acquire data of a specified regular-shape or spreading irregular-shape ranges. User can designate a regular range for querying. To track the diffusing event, this work proposes an efficient distributed algorithm for a dynamic tree structure in sensor networks. Users can oversee the spreading event via querying the incurred irregular-shape range. Additionally, this work proposes efficient approaches to gather data from sensor networks when voids exist. Finally, experimental results show that the proposed approaches are more energy-efficiency than the existing approach.

Keywords: data aggregation, grid, tracking, routing, wireless sensor networks

1. INTRODUCTION

Manufacturing of small and low-cost sensors has become technically and economically feasible recently due to technological advances. Sensors can detect environment, collect data, and communicate with each other. Wireless sensor networks (WSNs) are composed by a large number of sensors. WSNs are quickly gaining popularity due to the fact that they are potentially low cost solutions to a variety of real-world challenges [1]. WSNs have been applied on military and civil applications effectively, such as target field imaging, intrusion detection, weather monitoring, security and tactical surveillance, distributed computing, and so on. However, the limited resource is one of challenges in designing protocols for wireless sensor networks.

In a scenario of monitoring environment, mobile users (mobile sinks) expect to obtain information immediately when the normal or unexpected events occurred. They can inquire the interested data with a designated range. One of effective methods is that the

---

Received September 15, 2006; accepted February 6, 2007.
Communicated by Ten H. Lai, Chung-Ta King and Jehn-Ruey Jiang.

[*] This work was supported in part by the National Science Council of Taiwan, R.O.C., under grants No. NSC 95-2221-E-024-012 and NSC 95-2524-S-156-001.

sensor announces event by dissemination protocol when it detects an event occurred. The users obtain information and aggregate data efficiently [14].

In the previous researches, data dissemination protocols [8, 10, 11, 20-22] with the mobile sinks can be classified into two categories, tree-based and grid-based protocols. The tree-based protocols include SEAD [10] and SAFE [11]. SEAD is an approach to route sensor's data to mobile sinks. SEAD is suitable for applications with less strict delay requirements. Another protocol SAFE uses geographically flooding to forward query to nodes along the direction of the source. SAFE is effective in small-to-medium size sensor networks. However, the initial sensor flooding may consume too much energy in a very large network. The d-tree would be frequently reconstructed due to mobile sinks.

The other category is based on grid-based protocol, *e.g.* TTDD [21] and CODE [20]. TTDD and CODE are the event-driven data dissemination methods. In TTDD, a source announces event by grid dissemination nodes when the source detects an occurred event. The dissemination nodes relay the query and data for the mobile sink querying event. But the path from a source to a sink is not optimization. When a sensor communicates with a sink, the restriction of grid structure may multiply the length of a straight-line path by $\sqrt{2}$. Therefore, this approach incurs more energy and longer delays. The mobile sink also increases the energy consumption.

Another, CODE [20] is proposed for a grid-based sensor network. In each grid, a coordinator is selected to cache the occurred event. When a sink wants to get event data, it sends a query to its grid head. Based on the location of event and grid IDs, an efficient data dissemination path is established. If a sink moves out the original grid, it reconstructs a new route. CODE does not to solve the routing problem with obstacles or voids in a sensor field.

In CODE and TTDD, a sink has to reissue a request to query data or use local flooding to query data when it moved out the original grid. It increases the energy consumption and data collision in the networks. Additionally, the query is not only to collect the data of source grid. A sink may want to collect a flexible regional range of grids [4, 12, 16]. This work proposes a novel protocol which combines the data aggregation and data dissemination to gather data from the regular-shape and irregular-shape ranges. The proposed protocol is based on grid sensor networks. A sensor node, called head, is selected from the sensors in a grid for announcing and routing. When a head detects an event, it announces event to all head and this head is called source. A mobile sink can obtain the occurred event from the head of its grid. If the sink is interested in the sensed event, it queries the source via the grid head that called agent. By regular-shape range query, the sink designates the range of data aggregation. The previous researches do not focus on the monitoring of diffusing event such as diffused smoke, gas, fire, and so on. This work proposes an irregular-shape region querying to track the diffused event and aggregate data. The proposed protocol has three main properties. First, this work constructs data link of tree structure to track the diffusing event. Second, a link maintenance is supported the mobile sink. Third, this work studied [2, 3, 7, 9] to solve the sensor field with voids. As our best knowledge, this proposed protocol is the first protocol not only to combine with the data dissemination and data aggregation, but also to solve the problem of tracking the diffused event and data aggregation in the sensor field with voids.

The rest of this paper is organized as follows. Section 2 presents the data aggregation for two kinds of range queries. Section 3 proposes the data aggregation protocol with voids. Experimental results are described in section 4. We draw the conclusions of our approach in section 5.

2. DATA AGGREGATION OF RANGE QUERY

This section presents efficient data aggregation algorithms for range query. First, the basic assumptions of protocol are presented. Second, a virtual grid structure is introduced. Finally, two kinds of range querying, regular-shape and irregular-shape range querying, are presented.

2.1 Assumptions

This subsection describes the basic assumptions of protocol. A large number of homogeneous sensor nodes are deployed to monitor a field and detect some interested events or unexpected events. Sensors communicate with each other through short-range radios. Additionally, the sensors have global time synchronization. Each sensor is stationary and has the limited battery energy. Mobile sink(s) (equipped with mobile devices such as PDAs) can communicate with sensor nodes and request them to acquire and aggregate the interested data. Suppose that multiple sinks are moving around in the sensor fields. Sensors and mobile sinks can obtain their own locations by GPS [5] or other location approaches [17].

2.2 Virtual Grid Structure

The monitored area is divided into virtual grids as shown in Fig. 1. We notate $G_{(x,y)}$ as the grid coordinate, where x (or $G.x$) is grid x-coordinate and y (or $G.y$) is grid y-coordinate. Let R be the transmission distance of radio signal and d be the side length of grid. When the sensors have been deployed, the sensors in a grid elect a node to act head to record the information of the occurred events and route. At first, each node obtains neighboring information by hello message. Utilizing this information, a node that is closest to the central of grid is selected as head. An example is shown in Fig. 2 (a). If a head has no enough resources, one of other nodes in the same grid will be selected to replace it.

We assume that the side length of grids to guarantee that the grid heads can communicate with neighboring grids directly [6]. Here, we also assume that the communication range of node is able to communicate with the neighboring grids. The relationship between d and R is predefined. The maximum value of d is limited to $R/\sqrt{5}$. An example is shown in Fig. 2 (b). In this relationship, node A can communicate with node B which is in the farthest diagonal point.

2.3 Data Dissemination

Mobile sink (*MS* for short) does its job or task in the sensor field to monitor the territory. A *MS* expects to obtain the event information instantly when someone event occurred. An event usually no expects when to happen. Therefore, a sensor has to signal the

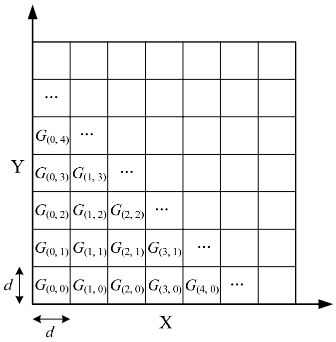
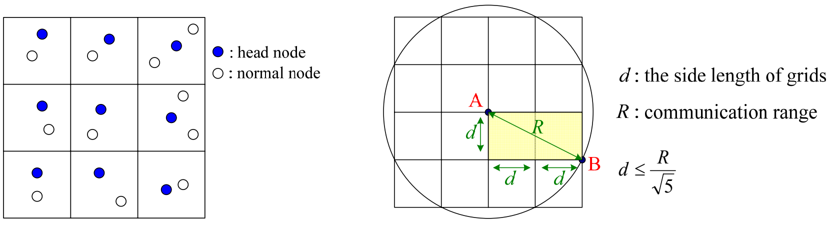


Fig. 1. A physical area divided into virtual grids.



(a) (b)

Fig. 2. Grid structure.

occurred event when it detects an event. This sensor that detects the event is called source node. The source propagates a *register* packet to all grid heads. The format of *register* packet is <*P_type*, *Src_id*, $Src\_Loc_{(x,y)}$, $Src\_G_{(x,y)}$, *hc*, *event_type*, *expired_time*>, where *P_type* is packet type, *Src_id* is source id, $Src\_Loc_{(x,y)}$ is source's physical location, $Src\_G_{(x,y)}$ is source's grid location, and *hc* is hop count.

When heads receive this packet, they store the register information in their register table. This information is kept within an expired time. If a head does not receive any further *register* packet and the time has expired, it clears up the information of event from table. The information of event is distributed in the grid heads. A *MS* can obtain the information of event from grid heads. The *MS* queries the closest grid head about the interested event.

The distributed method can decrease the overhead of *MS*. The *MS* does not need to periodically broadcast for querying event. If the information of occurred event is kept in a centralized directory, the *MS* has to maintain the routing between the source and *MS*. The routing maintenance consumes many extra overheads.

If a *MS* is interested in the occurred event, it sends a query packet to acquire the interested data. The *MS* can collect the data in a designated range by regular-shape range querying or monitor the diffusing event by irregular-shape range querying. The *MS* issues *query* packet to request the source to collect data. The format of *query* packet is <*P_type*, *agg_type*, *Src_id*, $Src\_Loc_{(x,y)}$, $Src\_G_{(x,y)}$, *A_id*, *MS_id*, *event_type*, $R_{(x,y)}$>, where *agg_ type* is aggregation type, *Src_id* is source id, *A_id* is agent id, *MS_id* is mobile sink id, and $R_{(x,y)}$ is the designated region.

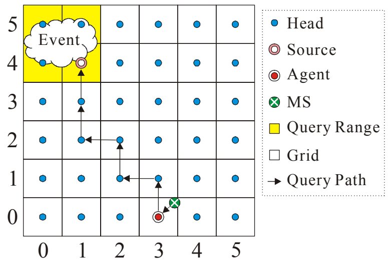


Fig. 3. An example for query forwarding.

According to the aggregation type and event type, a *MS* requests the source to aggregate data of regular-sharp or irregular-sharp range. If the aggregation type is 0, the range is regular-shape. Otherwise, the range is irregular-shape and the event must be diffusing event. The *MS* select the nearest head as agent. The request packet is forwarded via the agent. The routing in the proposed protocol uses geographic routing [9, 13, 15] (*i.e.* greedy-forwarding). A relay node forwards packet to the neighboring head that is closest to the source as shown in Fig. 3.

2.4 Querying Regular-Shape Range

This subsection proposes a data aggregation for regular-shape range querying. The mobile sink requests the source to collect data of designated range. The four designated directions of rectangle range are shown in Fig. 4. Fig. 4 (a) shows that the source collects the data of the northwest, where $R_{(x,y)}$ is set as $R_{(-2,+2)}$ (*i.e.* $R.x = -2$, $R.y = 2$). The source collects the data in a rectangular range that is from $G_{(2,3)}$ to $G_{(0,5)}$. Figs. 4 (b), (c) and (d) show the collecting of northeast, southwest and southeast, respectively.

When a source receives *query* packet, it constructs an aggregate data tree that is like comb. The detailed procedures are described in the following steps:

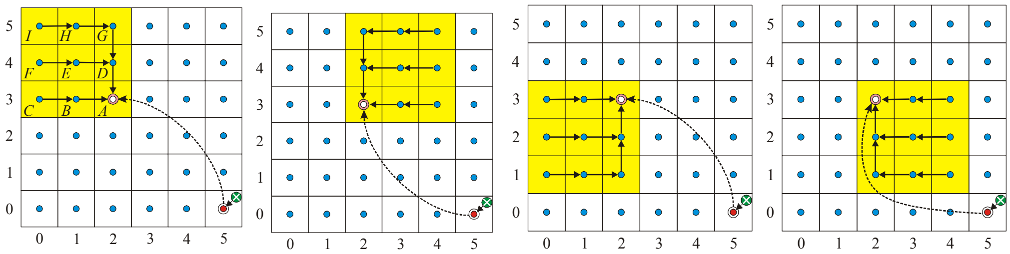
Step 1: The source computes its child grid according to $R_{(x,y)}$. If a head is within the query range ($G.x + R.x$, $G.y + R.y$), it is source's child. The source forwards *agg_reg* packet along a comb tree path. The source sends packet along grid y-coordinate, and then the packet is forwarded along grid x-coordinate. The forwarding path is like as comb tree. The format of *agg_reg* packet is <*P_type*, *Src_G*$_{(x,y)}$, *Rly_G*$_{(x,y)}$, *Chd_Glist*, *event_type*, $R_{(x,y)}$>, where *Src_G*$_{(x,y)}$ is source's grid, *Rly_G*$_{(x,y)}$ is relay node's grid, and *Chd_Glist* is a list of children grids.

Step 2: When a head receives *agg_reg* packet, it judges whether its grid is listed in *Chd_Glist*. If yes, it is a child and performs step 3. Otherwise, it stops to forward the packet.

Step 3: If child's $G.x$ is equal to *Src_G.x*, it forwards this packet to grid x-coordinate and y-coordinate and records the forwarding node as its parent.

Step 4: If child's $G.x$ is not equal to *Src_G.x* + $R.x$ and *Src_G.x*, it forwards this packet to grid x-coordinate and records the forwarding node as its parent.

Step 5: If child's $G.x$ is equal to *Src_G.x* + $R.x$, it means child is a leaf node. The child sends the sensed data to its parent. The data is aggregated from leaf to the source.



(a) (b) (c) (d)

Fig. 4. Data aggregation of regular-shape ranges with four directions.

An example illustrates how to collect the northwest regional range. In Fig. 4 (a), the source *A* receives the *query* packet, where *agg_type* is 0 and $R_{(-2,+2)}$. Source *A* forwards *agg_reg* packet with $R_{(-2,+2)}$ to *B* and *C*. When heads *B* and *D* receive the packet from source *A*, they record source *A* as their parent. Next, *B* and *D* forward the *agg_reg* packet to their children due to they are not the leaf nodes. Node *B* forwards packet to *C* and node *D* forwards packet to *E* and *G*. When head *C* receives the packet from *B*, it relays the sensed data to its parent *B* because node *C* is a leaf node. The parent waits for receiving the data from its children. Next, it aggregates data and forwards to its parent. When the source received the aggregated data from its children, it sends the final aggregated data to *MS*.

2.5 Querying Irregular-Shape Range

This work proposes the irregular-shape range querying to monitor the diffusing event. The diffusing event is dynamically extended to other region such as fire, smoke, or gas diffusion. When an event is spread to the neighboring grids, the source tracks the diffusing event. A *MS* can query a source to obtain the information of diffusing event. This work presents the methods of event tracking and data aggregating in the following subsections.

2.5.1 Detecting and tracking the diffusing event

This subsection presents an approach that constructs a tree to track the diffusing event. When a head node *n* detects an event, it first checks its register table whether the same event has been detected by its adjacent grid. If no, it broadcasts *register* packet to network (*i.e.* data dissemination phase as mentioned in section 2.3). If yes, we assume this event is diffusing. Node *n* records the adjacent head as its parent and sends a *data_link* packet to its parent. The format of *data_link* packet is <*P_type*, *N_id*, $N\_Loc_{(x,y)}$, $N\_G_{(x,y)}$, *P_id*>, where *N_id* is this node's id and *P_id* is parent's id. If the same event is detected by more than two adjacent grids, node *n* selects one of adjacent nodes as its parent that is closest to the source.

When the parent receives *data_link* packet, it records the forwarded node as its child. Next, the parent sends *ack* packet to the child. When the child receives *ack* packet, it sends *notification* packet to its four adjacent grid heads. The format of *notification* packet is <*P_type*, *N_id*, $N\_Loc_{(x,y)}$, $N\_G_{(x,y)}$, *hc*, *event_type*, *timeout*>. When the adjacent heads receive *notification* packet, they keep this information in register tables. If the event is diffused to neighbor grids, this event will be tracked by a tree structure efficiently. In other words, the source constructs a tree to track the diffusing event. A *MS* can query the source to obtain the information of diffusing event.

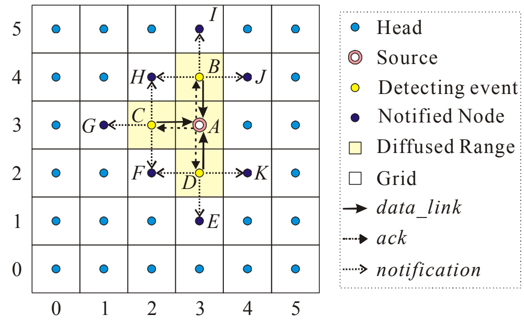


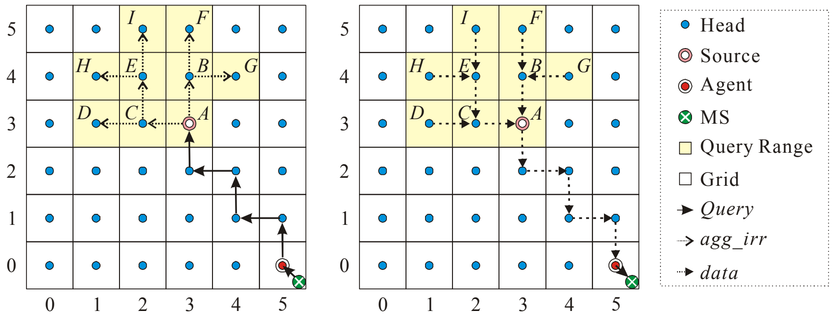
Fig. 5. An example for detecting and tracking diffusing event.

In Fig. 5, an event is diffused from $G_{(3,3)}$ to three grids $G_{(2,3)}$, $G_{(3,2)}$ and $G_{(3,4)}$. When heads *B*, *C* and *D* detect this event, they check their register tables whether the same event has been registered. Because source *A* has registered the event by data dissemination, they send *data_link* packet to the source *A*. When source *A* receives the packets from *B*, *C* or *D*, it replies *ack* packet to them respectively and records them as its children. When children *B*, *C* and *D* received *ack*, they broadcast *notification* packet to their adjacent grids instantly. This packet notifies their adjacent grids where some event has occurred, *e.g.* heads *H*, *I* and *J* know an event has occurred in $G_{(3,4)}$.

2.5.2 Aggregating data of irregular shape

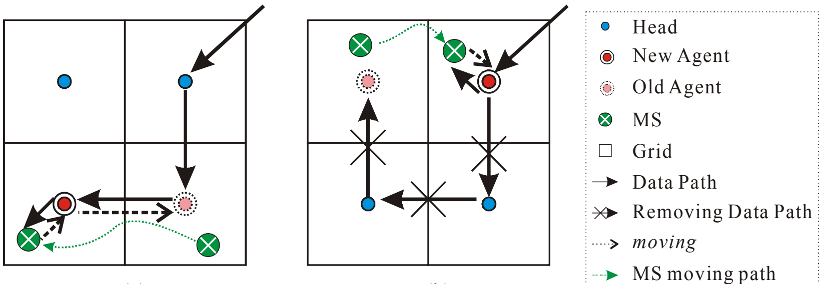
A source tracks the diffusing event by tree structure. When a source receives query packet and *agg_type* is equal to 1, it sends *agg_irr* packet to its children for collecting data. The format of *agg_irr* packet is <*P_type*, *Src_id*, *S_id*, *event_type*, *C_id_list*>, where *Src_id* is source id, *S_id* is sink id, and *C_id_list* is child id list. An example is shown in Fig. 6 (a). Source *A* sends *agg_irr* packet to its children *B* and *C*. Next, nodes *B* and *C* send *agg_irr* packet to their children *D*, *E*, *F* and *G*. When leaf nodes *D*, *F*, *G*, *H*, and *I* receive *agg_irr*, they send their sensed data to their parent. The aggregated data flows are shown in Fig. 6 (b). When source *A* receives data from its children *B* and *C*, it sends the final aggregated data to *MS* along the data path.

Each node supports cache mechanism in the proposed approach. When a node receives the aggregated data from the source to *MS*, it caches this data within *TTL* period. If *TTL* expired, it clears up data from its cache. All nodes on the data path between the source and *MS* have to cache data. When the cached node receives the same *query* from other *MS*, it replies the cached data to *MS* instantly. This cache mechanism decreases the cost of querying and aggregating data.



(a) (b)

Fig. 6. An example for querying irregular-shape range.



(a) (b)

Fig. 7. An example for sink mobility maintenance.

2.6 Sink Mobility Maintenance

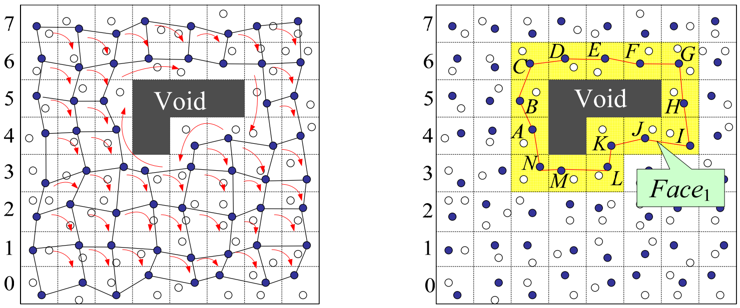
To monitor the interested regions, a *MS* receives data from the source continuously. We also consider how to maintain the routing between the mobile sink and the source. A *MS* checks its location every second. If a *MS* detects it moves out the original grid, it selects the head of new grid as new agent. Next, it sends *moving* packet to new agent as shown in Fig. 7 (a). When the new agent receives this packet, it checks whether it is in data path. If no, it sends *moving* packet to old agent for constructing new link. The old agent forwards data to new agent when it receives data from the source. If new agent is already in the data path, it sends *removing* packet to delete the data path between new and old agent to avoid looping problem. An example is shown in Fig. 7 (b).

3. DATA AGGREGATION OF RANGE QUERY WITH VOIDS

This section discusses how to aggregate data in the sensor field with void regions. The void region means that no node is in a grid due to some obstacles exist in the grid, no node is deployed, or the node has already died. This work utilizes the proposed *face routing* [7] to discover void regions and detour void regions. When a node cannot deliver packets by greedy-forwarding, it uses the face routing to detour void regions.

3.1 Void Region Discovery

This subsection presents a method to discover void regions. First, a node closest to the central of grid is selected as head in each grid. Next, the Gabriel Graph is employed to construct a graph as shown in Fig. 8 (a). Finally, each head is to send *face* packet by right-hand rule (clockwise rule) to collect the face's information [7]. The format of *face* packet is <*P_type*, *N_id*, $N\_Loc_{(x,y)}$, $N\_G_{(x,y)}$, *Rly_id*, *face_list*, *hc*>, where *N_id* is issued node id, *Rly_id* is the relay node id, and *face_list* is a list of the node's information in face.



(a) (b)

Fig. 8. An example for void discovery.

If a head receives the packet issued by oneself, it means that a face is collected completely. If a head can collect four heads in *face_list*, it means no void region exists in this face. The head discards the *face* packet and does not need to store any information. Otherwise, it means the face path surrounds a void region. The head has to store the *face_list* in face table. If the network has k void regions, k face routes will be discovered in network. We note $Face_i$ is the ith face routing. In Fig. 8 (b), $Face_1$ is found in the network. $Face_1$ is composed of head nodes A, B, C, D, E, F, G, H, I, J, K, L, M and N. Each head in $Face_1$ knows the face's neighboring information.

3.2 Routing with Void Region

In greedy-forwarding, a node forwards packet to a relay node which is the closest to the source. If a node cannot select a relay node from the neighboring table, it finds a relay node from face table. When a reply node is found from face table, the packet will be delivered along the face routing. Next, the packet is delivered to a node closest to the source. Finally, the last node forwards the packet to the source by greedy-forwarding.

Fig. 9 shows an example for routing with void region. A *MS* sends query packet to a source by greedy-forwarding. First, node K cannot find a relay node from its neighbors. Next, node K checks its face table and finds node C which is the closest to the source. K delivers the packet along face routing $K \to L \to M \to N \to A \to B \to C$. When C receives the packet, it forwards the packet to the source by greedy-forwarding.

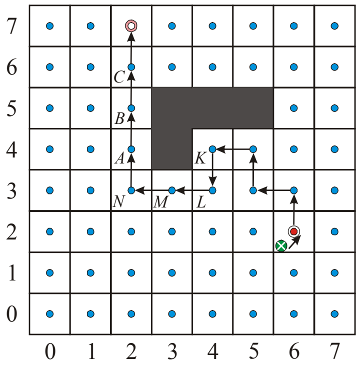


Fig. 9. An example for routing with void region.

3.3 Querying Regular-Shape Range with Void Region

A comb tree is constructed to aggregate data for regular-shape range. When some void regions exist, the face routing is utilized to detour void regions. The detailed procedure is described in the following steps:

Step 1: A source receives query form a *MS*. The source node computes its children according to $R_{(x,y)}$ and it sends *agg_reg* packet to its children.

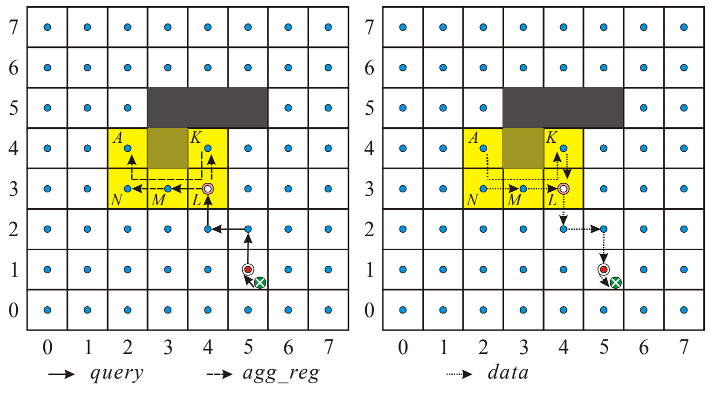
Step 2: If the source or relay node cannot find its child from its neighbor table, it means the adjacent child grid in a void region. The node has to find an alternate child from its face table. If the original adjacent child's *G.y* is equal to relay node/s *G.y*, the selection of alternate child is in step 3 else is in step 4.

Step 3: The alternate child's *G.x* has to in the requested range and the alternate child's *G.y* has to equal to the relay node's *G.y*.

Step 4: The alternate child's *G.x* has to in the requested range and the alternate child's *G.y* has equal to the relay node's *G.y* + 1.

Step 5: Next, the relay node sends *agg_reg* packet to the alternate with clockwise rule. Next steps are same as querying regular-shape range as described in subsection 2.4 from step steps 2 to 4.

Next, an example is illustrated the selecting of alternate child by face routing. In Fig. 10 (a), the source *L* receives *query* packet and the requested range is $R_{(-2,+1)}$. Node *L* first selects its children which is *K* in $G_{(4,4)}$ and *M* in $G_{(3,3)}$ from its neighboring table. Next, *L* sends *agg_reg* packet to them. They receive packet from source *L* and record *L* as their parent node. Node *M* selects *N* as its child from x-coordinate and sends *agg_reg* packet to *N*. When *N* receives the packet and it is a leaf node, *N* relays the sensed data to its parent *M*. Similarly, node *K* selects its child from $G_{(3,4)}$, but no node exists there. Next, *K* finds an alternate child *A* in $G_{(2,4)}$ from its face table and sends *agg_reg* packet to node *A* by face routing. The forward path is $K \rightarrow L \rightarrow M \rightarrow N \rightarrow A$. When node *A* receives *agg_reg* packet, it records node *K* as its parent. Because node *A* is a leaf node, *A* relays the sensed data to its parent *K* along path $A \rightarrow N \rightarrow M \rightarrow L \rightarrow K$ as shown in Fig. 10 (b). When node *K* receives data from its child *A*, node *K* sends the aggregated data to its



(a) (b)

Fig. 10. Querying Regular-shape range in northwest direction with void.

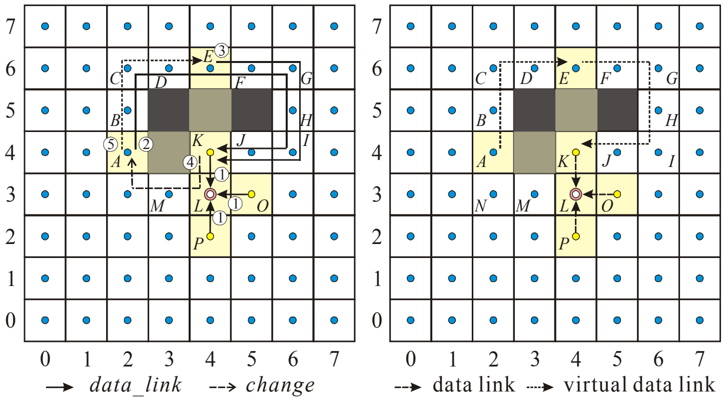
parent L. Similarly, node M sends the aggregated data to its parent L when M receives data from its child N. When the source L receives all aggregated data from its children, it replies the final data to *MS*.

3.4 Querying Irregular-Shape Range with Void Region

Diffusing event may cross void regions and spread to the opposite side. To guarantee be able to track the diffusing event in network with void regions, this work proposes a distributed approach to construct a tracking tree. When a node n in $face_i$ detects event, it first checks whether the adjacent grid has registered the same event. If true, it constructs the tracking tree as mentioned in section 2.5.1. Otherwise, it checks whether the grid in $face_i$ has registered the same event. The node n sends *data_link* packet to check event along the face routing (clockwise). If a head p in $face_i$ has the same event and it has no child, it records node n as its child and replies *ack* packet to n. When n receives *ack* packet, it records p as its parent and sends *notification* packet to its neighbors. A virtual data link is constructed between node n and its parent p. If the parent p has a child m, it has to send *change* message to m before p records node n as it child. The original child m's new parent is changed into n. In our approach, the parent in $face_i$ has only one virtual data link that connects to its child.

In Fig. 11 (a), the source L detects an event and registers this event to all heads as described in section 2.3 data dissemination. When the event is diffused to $G_{(4,2)}$, $G_{(4,4)}$ and $G_{(5,3)}$, head nodes P, K and O send *data_link* packet to their parent L. Source L returns *ack* packet to them. When P, K and O receive *ack* packet, they send *notification* packet to their adjacent grids as described in section 2.5.1.

First, we assume the event is diffused to $G_{(2,4)}$ through void regions and node A detects this event. Because node A's adjacent grids do not have this event and A is in $face_1$, A sends *data_link* packet to check whether the same event has occurred in the face. Head K receives *data_link* packet from A and records node A as its child. Next, K sends *ack* packet to construct a virtual link between A and K. Head A receives *ack* packet and records K as its parent. Next, A sends *notification* packet to its adjacent grids.



(a) (b)

Fig. 11. Constructing the link of continuous event with void.

Next, we assume the event is diffused to $G_{(4,6)}$ through void regions as shown in Fig. 11 (a). Similarly, head *E* sends *data_link* packet along the face routing. Because head *K* has child *A* in the face, *K* sends *change* packet to *A* to change *A*'s parent into *E*. Next, *K* returns *ack* packet to its child *E*. When *E* receives *ack* packet, it records *K* as its parent and *A* as its child. *E* also sends *notification* packet to its adjacent grids. Fig. 11 (b) shows the tree data link, source *L* has children *P*, *K* and *O*; head *K* has child *E*; head *E* has child *A*.

This work constructs virtual links to track the event that is diffused through void regions. Though void regions exist in the network, the mobile sink can monitor and aggregate the data of diffusing event efficiently.

4. EXPERIMENTAL RESULTS

4.1 Simulation Model

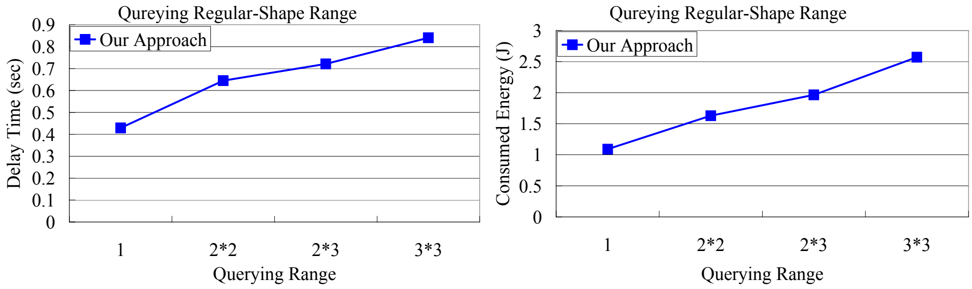
Some grid-based data dissemination protocols have been proposed such as CODE [20] and TTDD [21]. CODE is based on a virtual grid structure. This work proposes the approaches for querying regular-shape and irregular-shape range. First, we compare our approach with CODE by NS-2 [18]. Next, we discuss the number of sinks and the percentage of void grids how to impact on the performance of approach. We assume the energy consumption of transmitting, receiving and idling are 0.7W, 0.35W and 0.035W, respectively. The initial energy is equal to 10*J*. NS-2 simulator uses MAC IEEE 802.11 DCF that SENSE implements. We assume that the ratio of sensing range and the communication range is 1:2 [19]. The transmission range of sensor is 112m, and the gird size is 50m. The sensing range of sensor can fully cover with its grid. The data rate is 2m/s and the mobility model uses random waypoint model. The sensor network consists of 200 sensor nodes which are deployed uniformly in a 500m × 500m field (*i.e.* two sensor nodes per 50m × 50m grid). The radio propagation model uses *Two-ray ground* and each channel is omni-directional in the simulator. The simulation time is 100 seconds.

This work uses four metrics to evaluate the performance of methods. The *total energy consumption* is the register (data dissemination) and query energy consumption. The *delay time* is the wait time of mobile sink to obtain data. The *success ratio* is the ratio of received data packet to querying.

4.2 Simulation Results

4.2.1 Aggregating data of regular-shape range

We assume that the velocity of sink is 2m/s and sink queries and aggregates the data with varied range in this simulation. From the results, we can observe the delay time and the energy consumed for querying regular-shape range. Fig. 12 (a) shows the delay time. When *MS* requests to acquire data in a bigger range, *MS* has to wait the longer delay time. This is because the source has to collect more data in bigger range. Fig. 12 (b) shows the energy consumption. This is also similar to delay time. Bigger querying range need consume more energy.



(a) (b)

Fig. 12. Data aggregation in varied range.

4.2.2 Aggregating data of irregular-shape range

In this simulation scenario, we assume the velocity of sink is 2m/s and the event is diffusing. The sink queries data with irregular range. We assume that spread range of event is varied from 1 to 9 grids. First, we compare the energy consumption in data dissemination with CODE. CODE does not support the tracking of diffusing event. In CODE, each grid head has to run data dissemination when it detects an event. In the proposed approach, a track tree is constructed to monitor the diffusing event. Only the source needs to do data dissemination. A lot of cost is decreased in data dissemination. Fig. 13 shows CODE consumes more energy when the spread range becomes great. The proposed approach only needs the energy consumption in the data dissemination of the source and constructing track link. The energy consumption is only slightly increased.

Fig. 14 shows the delay time and energy consumption for aggregating data in irregular-shape range. When a sink wants to monitor the diffusing event, CODE has to issue multiple queries to collect data. The multiple queries will consume a lot of resource and obstruct the network. Therefore, the sink cannot obtain data fast due to the network

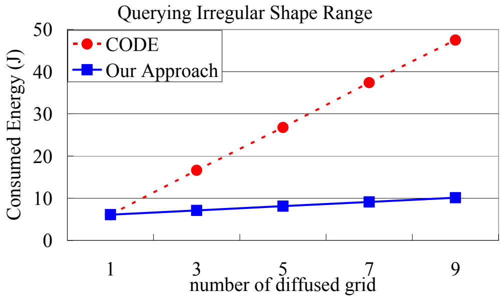
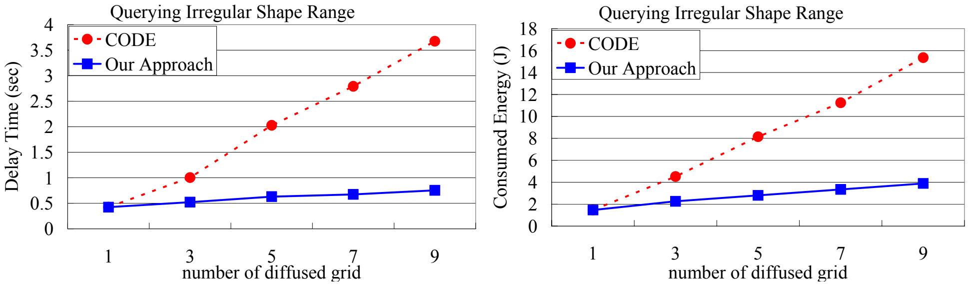


Fig. 13. Energy consumption in data dissemination.



(a) (b)

Fig. 14. Data aggregation for irregular querying range.

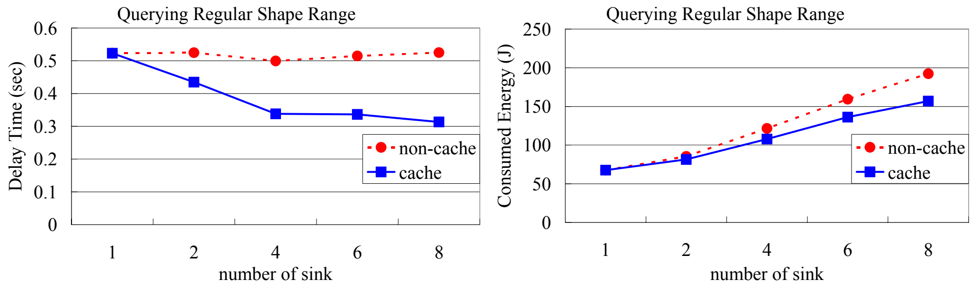
obstructed. We can observe this from Fig. 14 (a). Fig. 14 (b) shows the energy consumption for aggregating data. The proposed approach improves the drawback of CODE and provides an efficient approach to monitor and track the diffusing event.

4.2.3 Impact of cache mechanism

In this subsection, we assume the number of sinks is varied from 1 to 8. All sink query and collect data in the same range. Additionally, eight kinds of different static events are occurred in network simultaneously. The velocity of sink is 2m/s and the querying range is 2 × 2 grids. The cache mechanism is used to decrease the energy consumption, when multiple sink issued the same query. Fig. 15 (a) shows the average delay time. When the same queries increased, the cache mechanism can decrease the energy consumption of aggregating data. The sinks obtain data from the cached node immediately. If the node does not support the cache mechanism, the source has to aggregate data when it received query. The cache mechanism decreases not only the delay time but also the resource consumption. The result of energy consumption is shown in Fig. 15 (b).

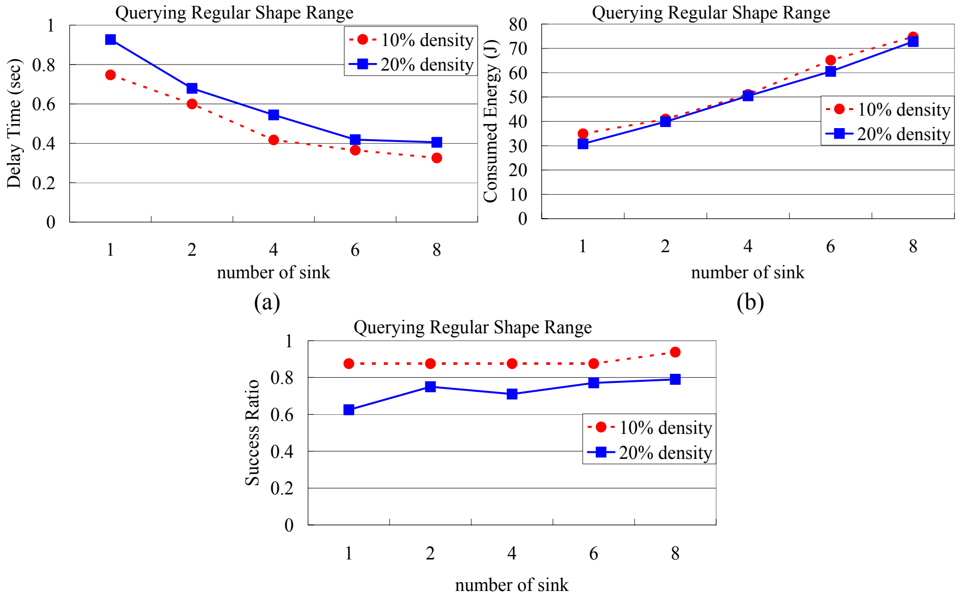
4.2.4 Impact of void region

In this scenario, we assume 100 sensors are deployed in the network. We vary the density of void grid from 10% to 20% in the network. When the density of void gird is 10%, 10 void grids exist in the network. We simulate that the different percentage of



(a) (b)

Fig. 15. The impact of cache mechanism.



(a) (b)

(c)

Fig. 16. The impact of void region.

void grids impacts on performance. The querying range is 2 × 2 grids. Eight kinds of different static events are occurred in network simultaneously. The number of sinks is varied from 1 to 8. The node supports the cache mechanism.

When the density of void grid increased, the sink has to wait for the longer time to obtain data as shown in Fig. 16 (a). This is because that the data aggregation has to avoid void regions by face routing. The existed void grids impact and increase the hop count of routing. In high void grid density, the energy consumption is decreased slightly as shown in Fig. 16 (b). This is because that the success rate in 20% density of void grid is low than that in 10% as shown in Fig. 16 (c). Some packets are not delivered successfully, so the energy consumption is decreased slightly. The mobile sink cannot communicate with sensors when it moves into a big void region. Therefore, the mobile sink cannot obtain the data from the source. The success ratio is decreased while the density of void grids increased.

5. CONCLUSION

This work presents novel and efficient approaches to aggregate data with different querying type. We addressed a regular-shape query to collect data for the designated rectangle range and an irregular-shape query to monitor and collect data for the diffusing event. The diffusing event is dynamically extended in network. The proposed approach is based on grid structure. In each grid, a node is selected as head for caching data and routing. When a head detects an event, it announces the occurred event to all heads by data dissemination. A mobile sink can obtain event from grid heads. If a mobile sink is interested in an event, it requests the source to aggregate data with a querying range. Additionally, this work presents an approach to monitor the diffusing event. When a diffusing event occurred, a track tree is constructed to monitor the diffusing region of event. A mobile sink can obtain data efficiently via the source. The cache mechanism is used to solve the same queries issued from multiple sinks. Moreover, void regions exist in network because some grids are not deployed sensors or sensors had died. The face routing is utilized to detect and detour void regions. The simulation results show that the proposed approach not only decreases the energy consumption but also increases the performances. Additionally, the proposed method can monitor and aggregate data for diffusing event effectively. When the same queries are issued from multiple mobile sinks, the cache mechanism can decrease the resource consumption and the sinks can obtain data faster. Finally, the other major contributions are to handle the routing and data aggregation with void regions in sensor networks.

REFERENCES

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayircr, "Wireless sensor network: a survey," *Computer Networks*, Vol. 38, 2002, pp. 393-422.
2. C. Y. Chang, C. T. Chang, and S. C. Tu, "Obstacle-free geocasting protocols for single/multi-destination short message services in ad hoc networks," *Wireless Networks*, Vol. 9, 2003, pp. 143-155.
3. Y. S. Chen, S. Y. Ann, and Y. W. Lin, "VE-mobicast: a variant-egg-based mobicast routing protocol for sensornets," *Wireless Networks*, (in press).
4. P. Coschurba, K. Rothermel, and F. Dürr, "A fine-grained addressing concept for geocast," in *Proceedings of the International Conference on Architecture of Computing Systems: Trends in Network and Pervasive Computing*, 2002, pp. 101-113.
5. B. Hoffman-Wellenhof, H. Lichteneger, and J. Collins, *Global Positioning System: Theory and Practice*, 4th ed., Springer-Verlag, 1997.
6. Y. Hu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th ACM Annual International Conference Mobile Computing and Networking*, 2001, pp. 70-84.
7. Q. Huang, S. Bhattacharya, C. Lu, and G. C. Roman, "FAR: face-aware routing for mobicast in large-scale sensor networks," *ACM Transactions on Sensor Networks*,

Vol. 1, 2005, pp. 240-271.

8. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, Vol. 11, 2003, pp. 2-16.
9. B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking*, 2000, pp. 243-254.
10. H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, "Dissemination: minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, 2003, pp. 193-204.
11. S. Kim, S. H. Son, J. A. Stankovic, S. Li, and Y. Choi, "SAFE: a data dissemination protocol for periodic updates in sensor networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops*, 2003, pp. 19-22.
12. Y. B. Ko and N. H. Vaidya, "GeoTORA: a protocol for geocasting in mobile ad hoc networks," in *Proceedings of the 8th International Conference on Network Protocols*, 2000, pp. 240-250.
13. E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proceedings of the 11th Canadian Conference on Computational Geometry*, 1999, pp. 51-54.
14. B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the International Workshop on Distributed Event-Based Systems*, 2002, pp. 575-578.
15. F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing of theory and practice," in *Proceedings of the 22nd ACM Symposium on the Principles of Distributed Computing*, 2003, pp. 63-72.
16. W. H. Liao, Y. C. Tseng, and J. P. Sheu, "GRID: a fully location-aware routing protocol for mobile ad hoc networks," *Telecommunication Systems*, Vol. 18, 2001, pp. 37-60.
17. A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks", in *Proceedings of ACM Workshop on Wireless Sensor Networks and Applications*, 2002, pp. 105-111.
18. VINT Project, "Network simulator version 2 (NS-2)," Technical Report, http://www.isi.edu/nsnam/ns, 2001.
19. G. Xing, C. Lu, R. Pless, and Q. Huang, "Impact of sensing coverage on greedy geographic routing algorithms," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 17, 2006, pp. 348-360.
20. H. L. Xuan and S. Lee, "CODE: a coordination-based data dissemination protocol for wireless sensor networks," in *Proceedings of the IEEE Intelligent Sensors Conference on Sensor Networks and Information*, 2004, pp. 14-17.
21. F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "TTDD: two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, Vol. 11, 2005, pp. 161-175.
22. W. Zhang, G. Cao, and T. L. Porta, "Dynamic proxy tree-based data dissemination schemes for wireless sensor networks," in *Proceedings of 1st IEEE Mobile Ad-Hoc and Sensor System Conference*, 2004, pp. 21-30.

Tzung-Shi Chen (陳宗禧) received the B.S. degree in Computer Science and Information Engineering from Tamkang University, Taiwan, in June 1989 and the Ph.D. degree in Computer Science and Information Engineering from National Central University, Taiwan, in June 1994, respective. He joined the faculty of the Department of Information Management, Chung Jung Christian University, Tainan, Taiwan, as an Associate Professor in June 1996. Since November 2002, he has become a Professor at the Department of Information Management, Chung Jung Christian University. He was a visiting scholar at the Department of Computer Science, University of Illinois at Urbana-Champaign, U.S.A., from June to September 2001. He was the chairman of the Department of Information Management at Chung Jung Christian University from August 2000 to July 2003. Since August 2004, he has become a Professor at the Department of Information and Learning Technology, National University of Tainan, Tainan, Taiwan. Currently, he is the chairman of the Department of Information and Learning Technology, National University of Tainan. He has served as a Guest Editor of Journal of Internet Technology, special issue on "Wireless Ad Hoc and Sensor Networks," 2005. He has also served as PC members on many international conferences. He co-received the best paper award of 2001 IEEE ICOIN-15. He is a member of the IEEE Computer Society. His current research interests include mobile computing and wireless networks, grid computing, mobile learning, and data mining.

Yi-Shiang Chang (張益祥) received the B.S. degree in Department of Information Management from Shih Hsin University, Taiwan, in June 2004 and the M.S. degree in Department of Information and Learning Technology from National University of Tainan, Taiwan, in July 2006. His research interests include mobile computing, wireless networking, and mobile learning. In wireless networking, he is particularly interested in tracking, routing, and energy saving issues in sensor networks.

Hua-Wen Tsai (蔡華文) received the B.S. degree in Information Management from Chang Jung Christian University, Taiwan, in June 1998 and the M.B.A. degree in Business and Operations Management from Chang Jung Christian University, Taiwan, in June 2001. Since September 2001, he has been working towards the Ph.D. degree and currently is a doctoral candidate in the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan. His research interests include wireless communications, ad hoc networks, and sensor networks.

Chih-Ping Chu (朱治平) received a B.S. degree in Agricultural Chemistry from National Chung Hsing University, Taiwan, an M.S. degree in Computer Science from the University of California, Riverside, and a Ph.D. degree in Computer Science from Louisiana State University. He is currently a professor in the Department of Computer Science and Information Engineering of National Cheng Kung University, Taiwan, R.O.C. His research interests include parallelizing compilers, parallel computing, parallel processing, internet computing, DNA computing, and software engineering.