

# Data Aggregation for Range Query in Wireless Sensor Networks<sup>\*</sup>

Tzung-Shi Chen<sup>1</sup> Yi-Shiang Chang<sup>1</sup> Hua-Wen Tsai<sup>2</sup> Chih-Ping Chu<sup>2</sup>

<sup>1</sup>Department of Information and Learning Technology  
National University of Tainan  
Tainan 700, Taiwan  
chents@mail.nutn.edu.tw

<sup>2</sup>Department of Computer Science and Information Engineering  
National Cheng Kung University  
Tainan 701, Taiwan  
chucp@csie.ncku.edu.tw

**Abstract**—This work proposes efficient data aggregation algorithms based on a virtual grid for range query in wireless sensor networks. A sensor node is selected to be a manager, called head, in each grid. The responsibilities of head are to detect generated event, announce to all other heads, and respond to a moving user. A user obtains the occurred event information from its grid head. If a user is interested in an event, it issues a query to acquire data from a specified regular-shape or spreading irregular-shape ranges. Users can oversee the spreading event via querying the incurred irregular-shape range. In addition, this work proposes efficient approaches to gather data from sensor networks while voids exist. Finally, experimental results show that the proposed approaches are more energy-efficiency than the existing approach.

**Index Terms**—data aggregation, grid, tracking, routing, wireless sensor networks

## I. INTRODUCTION

Manufacturing of small and low-cost sensors has become technically and economically feasible recently due to technological advances [4]. Wireless sensor networks (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. In a scenario of monitoring environment, moving users expect to obtain the information immediately when the normal or unexpected events occurred and they can inquire the interested data.

In the previous researches, data dissemination protocols with the mobile sinks can be classified into two categories, tree-based and grid-based protocols. The tree-based protocols include SEAD [3] and SAFE [4]. SEAD is an approach for routing sensor's data to mobile sinks. SEAD is suitable for applications with less strict delay requirements. Another protocol SAFE uses geographically flooding to forward the query towards to the source that detects the event. SAFE is effective in small-to-medium size sensor networks. However, in a very

large network, the initial sensor flooding may consume too much energy. The d-tree would be frequently reconstructed due to mobile sinks.

The other category is based on grid-based protocol. TTDD [9] and CODE [8] are the event-driven data dissemination methods. In TTDD, the source announces the event by grid dissemination nodes when the source detects occurred event. The dissemination nodes are to transmit packets when the mobile sink queries the event. But the path from the source to the sinks 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 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 the coordinator in its grid. An efficient data dissemination path is established based on the location of event and grid IDs. If the sink moves out the original grid, it reconstructs a new one. But CODE does not to solve the routing problem with obstacles or voids in a sensor field.

In CODE and TTDD, the sinks have to reissue a query to request data or using local flooding to request data when they moved out the original grid. This will increase the energy consumption and the collision.

In this paper, we propose a novel protocol which combines the data aggregation and data dissemination to gather data from the regular-shape and irregular-shape ranges. Our protocol is based on grid sensor networks. A sensor node, called head, is selected from the sensors in a grid for announcing and routing. Each mobile sink can obtain the occurred event from a grid head where it is located. If the sink is interested in event, it queries the source via the grid heads. By regular-shape range query, the sink designates the range for 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 for tracking the diffused event and aggregating data. The proposed proto-

<sup>\*</sup> This work is supported in part by National Science Council under grant NSC-95-2221-E-024-012, Taiwan

col has three main properties. First, a tree is constructed to track the diffusing event. Second, a link maintenance is supported the mobile sink. Third, this work studied [2] 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 II presents the data aggregation for two kinds of range queries. Section III proposes the data aggregation protocol with voids. Experimental results are described in Section IV. We draw the conclusions of our approach in Section V.

## II. DATA AGGREGATION OF RANGE QUERYING

This section presents an algorithm of data aggregation for range querying. First, the basic assumptions are presented. A large number of homogeneous sensor nodes are deployed to monitor a field and detect some interested events. Sensors can communicate with each other through short-range radios. In addition, 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 networks 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 or other location approaches.

### A. Virtual Grid Structure

The monitored area is divided into virtual grids. We notate  $Grid_{(x, y)}$  as the grid coordinate, where  $x$  (or  $G.x$ ) is grid  $x$ -coordinate and  $y$  (or  $G.y$ ) is grid  $y$ -coordinate. The sensors in a grid elect a node as head to record the information of the occurred event and route. The maximum value of  $d$  is limited to  $R/\sqrt{5}$ , where  $R$  is the transmission distance of radio signal and  $d$  is the side length of grid. The side length of grids guarantees that the grid heads can communicate with neighboring grids directly [1]. If a head has no enough resources, one of other nodes in the same grid will be selected to replace it.

### B. Data Dissemination

Mobile sink ( $MS$  for short) does its job or task in the sensor field to monitor the territory. The  $MS$  expects to obtain the information instantly when someone event occurred. An event usually no expects when to happen. Therefore, the sensor has to signal the occurred event when it detects an event. It propagates a *register* packet to all grid heads. The format of *register* packet is  $\langle P\_type, Src\_id, Loc_{(x, y)}, G_{(x, y)}, hc, event\_type, expired\_time \rangle$ , where  $P\_type$  is packet type,  $Src\_id$  is source id,  $Loc_{(x, y)}$  is source's physical location,  $G_{(x, y)}$  is source's grid location, and  $hc$  is hop count. The sensor called

source detects the event.

When heads receive this packet, they store the register information in their register table. This information is kept in an expired time. If a head does not receive any further *register* packet and the time has expired, it clears up the event information from table. The event information is distributed in the grid heads. The  $MS$  obtains the event information from the nearest grid head. The distributed method can decrease the query overhead. The  $MS$  does not need to periodically broadcast for querying event. If the information of occurred event is kept in a centralized directory,  $MS$  has to maintain the routing between the source and  $MS$ . The routing maintenance spends many extra overheads.

If a  $MS$  is interested in the occurred event, it sends a query packet to acquire the interested data. The  $MS$  designates a range to collect the data by regular-shape range querying or monitors 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  $\langle P\_type, agg\_type, Src\_id, Loc_{(x, y)}, G_{(x, y)}, A\_id, S\_id, event\_type, R_{(x, y)} \rangle$ , where  $agg\_type$  is aggregation type,  $Src\_id$  is source id,  $A\_id$  is agent id,  $S\_id$  is sink id, and  $R_{(x, y)}$  is the designated region.

According to the aggregation type and event type,  $MS$  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  $MS$  selects the nearest head as *agent*. The request packet is forwarded via the agent. The routing of our method is geographic routing [5] (i.e. greedy-forwarding). A relay node forwards the packet to the neighboring head that is closest to the source.

### C. Querying Regular-Shape Range

This subsection proposes data aggregation for regular-shape range querying. The mobile sink requests the source to collect the data in a designated range. One of four designated directions of rectangle range is shown in Fig. 1 (a). Fig. 1 (a) shows that the source collects the data in 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  $G_{(3, 3)}$  to  $G_{(1, 5)}$ .

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

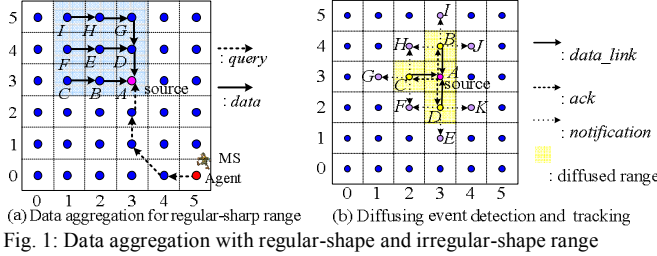
S1: The source computes its child grid according to  $R_{(x, y)}$ . The head in the child grids are the source's children. The source forwards *agg\_reg* packet to the neighboring heads. The format of *agg\_reg* packet is  $\langle P\_type, Src\_G_{(x, y)}, Rly\_G_{(x, y)}, Chd\_G_{(x, y)}\_list, event\_type, R_{(x, y)} \rangle$ , where  $Src\_G_{(x, y)}$  is source's grid,  $Rly\_G_{(x, y)}$  is relay node's grid, and  $Chd\_G_{(x, y)}\_list$  is a list of child's grid.

S2: When a head receives the *agg\_reg* packet, it judges whether its grid is listed in  $Chd\_G_{(x, y)}\_list$ . If yes, the head is in child grid and performs S3. Otherwise, it stops to forward the packet. If  $G.x = Src\_G.x$ , the head forwards packet to  $x$ - and

y-directions. Otherwise, the head forwards packet to x-direction.

S3: If head's  $G.x$  is not equal to  $Src\_G.x+R.x$ , it forwards this packet and records the forwarding node as its parent.

S4: If head's  $G.x$  is equal to  $Src\_G.x+R.x$ , it means the head is a leaf node and it sends the sensing data to its parent. The sensing data is aggregated from leaf to the source.



## D. Querying Irregular-Shape Range

### 1) Detecting and Tracking the Diffusing Event

This work proposes 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 the event is spread to the neighboring grids, the source tracks the diffusing event. A MS can query the source to obtain the information of diffusing event.

When a head node  $n$  detects an event, it first checks register table whether the same event is detected in adjacent grid. If no, it broadcasts *register* packet to network (as mentioned in Section II.B). If yes, we assume this event is diffusing from adjacent grid. Node  $n$  records the adjacent head as its parent and sends *data\_link* packet to its parent. The format of *data\_link* packet is  $\langle P\_type, N\_id, Loc_{(x,y)}, G_{(x,y)}, P\_id \rangle$ , 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 the closest node to the source. When the parent receives the *data\_link* packet, it records the forwarded node as its child. Next, the parent sends *ack* packet to its child. When the child receives *ack* packet, it sends *notification* packet  $\langle P\_type, N\_id, Loc_{(x,y)}, G_{(x,y)}, hc, event\_type, timeout \rangle$  to its four adjacent grid heads. When the adjacent heads receive the *notification* packet, they keep this information in register table. The source constructs a tree to track the diffusing event and aggregates the information of diffused event easily.

In Fig. 1 (b), the 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 whether the same event is registered in their register tables. Because source  $A$  has registered the event by data dissemination, they send *data\_link* packet to  $A$ . 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  $B$ ,  $C$  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  $G_{(3,3)}$ .

### 2) Aggregating Data in Irregular-Shape Region

A source tracks the diffusing event by tree structure. When the source receives query packet and *agg\_type* is equal to 1, it sends *agg\_irr* packet to its children to collect data. The format of *agg\_irr* packet is  $\langle P\_type, Src\_id, S\_id, event\_type, C\_id\_list \rangle$ , 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. 2 (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 the *agg\_irr*, they send their sensing data to their parent. The aggregated data flows are shown in Fig. 2 (b). When source  $A$  receives the data from its children  $B$  and  $C$ , it sends the final aggregated data to  $MS$  along the query path.

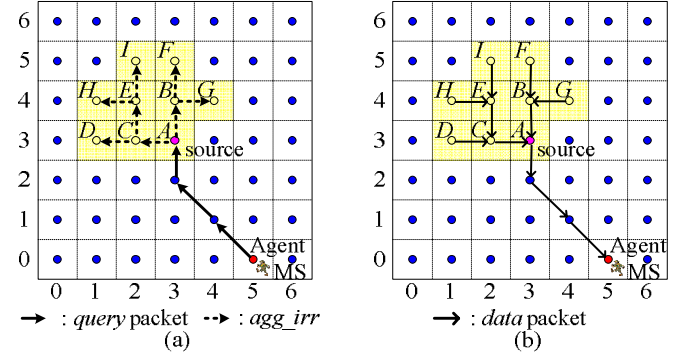


Fig. 2: An example for querying irregular-shape range

Each node supports cache mechanism in 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 forwarding 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.

### E. Sink Mobility Maintenance

To monitor the interested regions,  $MS$  receives data from the source continuously. This work also considers how to maintain the routing between  $MS$  and source. The  $MS$  checks its location every second. If  $MS$  detects it moves out the original grid, it selects the head of new grid as new agent. Next, the  $MS$  sends *moving* packet to new agent. When the new agent receives this packet, it checks whether it is in data path. If no, it sends the *moving* packet to old agent for constructing new link. Old agent forwards data to new agent when it receives data from the source. If new agent is in the forwarding data path, it sends *removing* packet to delete the path between new and old agent to avoid looping problem.

## III. DATA AGGREGATION OF RANGE QUERY WITH VOIDS

This section discusses how to aggregate data 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.

#### A. Void Discovery

This work utilizes the proposed *face routing* [2] to discover void regions and detour void regions. After constructing grid structure, this work employs the Gabriel Graph to construct a graph as shown in Fig. 3 (a). Each head sends *face* packet by clockwise rule to collect the face's information. The format of *face* packet is  $\langle P\_type, N\_id, Loc_{(x,y)}, G_{(x,y)}, Rly\_id, face\_list, hc \rangle$ , 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. 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. 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  $i$ th face routing. In Fig. 3 (b),  $Face_1$  is found in the network.  $Face_1$  is composed of head nodes  $A \sim N$ . Each head in the  $Face_1$  knows the node information of  $Face_1$ .

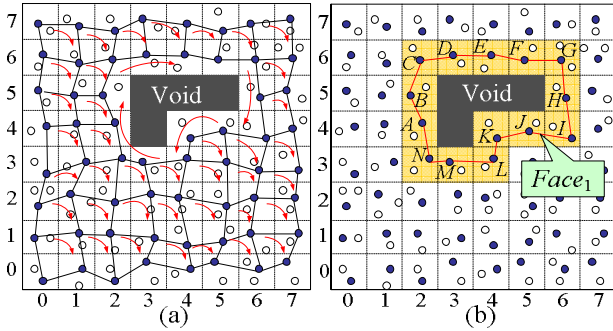


Fig. 3: An example for void discovery

#### B. Routing with Void Region

In greedy-forwarding, a node forwards packet to a relay node which is closest to the source. When a node cannot find out a relay node from neighboring table, it selects a relay node from face table. The packet will be delivered along the face routing to a node closest to the source. Finally, the last node forwards the packet to the source by greedy-forwarding.

Fig. 4 shows an example for routing with void region. Node  $K$  cannot find out a relay node from its neighbors. Next,  $K$  delivers the packet along face routing  $K \rightarrow L \rightarrow M \rightarrow N \rightarrow A \rightarrow B \rightarrow C$ . When  $C$  receives the packet, it forwards the packet to the source by greedy-forwarding.

#### C. Querying Regular-Shape Range with Void Region

We construct the comb tree to aggregate the data in regular-shape range. When some void regions exist, we utilize the face routing to detour void regions. The detailed procedure is described in the following steps:

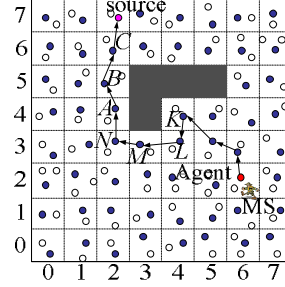


Fig. 4: An example for routing with void

S1: The source computes its children according to  $R_{(x,y)}$  and it sends *agg\_reg* packet to its children along comb tree.

S2: If the source or relay node cannot find out child from neighbor table, it means the adjacent child grid in a void region. The node has to find an alternate child from face table.

S3: The node sends *agg\_reg* packet to the alternate node with clockwise rule. The steps of querying regular-shape range are same as described in Subsection 2.4 (steps S2 to S4).

In Fig. 5 (a),  $K$  cannot find out its child in  $G_{(3,4)}$ , so  $K$  finds an alternate child  $A$  in  $G_{(2,4)}$  from face table and sends *agg\_reg* packet to  $A$  by face routing. The forward path is  $K \rightarrow L \rightarrow M \rightarrow N \rightarrow A$ . When  $A$  receives *agg\_reg* packet, it records  $K$  as its parent. Due to  $A$  is a leaf node, it relays its sensing data to parent  $K$  along path  $A \rightarrow N \rightarrow M \rightarrow L \rightarrow K$  as shown in Fig. 5 (b).

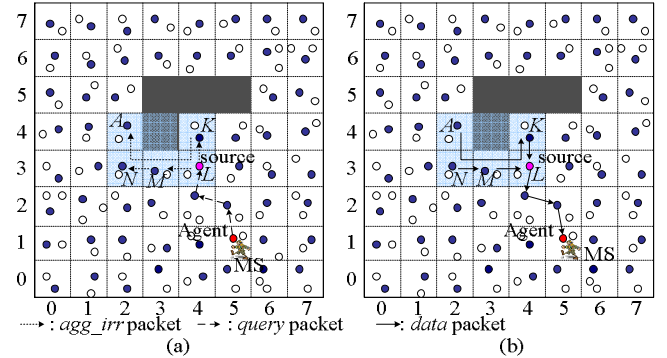


Fig. 5: Querying Regular-shape range in northwest direction with void

#### D. Querying Irregular-Shape Range with Void Region

Diffusing event may cross void regions and spread to the opposite side. To guarantee tracking 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 an event, it first checks whether the adjacent grid has registered the same event. If true, it constructs the tracking tree as mentioned in Section II.D.1. Otherwise, it checks whether the grid in  $face_i$  has registered the same event. The node  $n$  sends *data\_link* packet to check the event along the face routing (clockwise). If a head  $p$  in  $face_i$  has the same event and it has no child, it records  $n$  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 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 its child. The original child  $m$ 's new parent is changed

as  $n$ . In this approach, the parent in  $face_i$  has only one virtual link that connects to child. This work constructs virtual link to track the event that is diffused through void regions. Though void regions exist in the network, the  $MS$  can monitor and aggregate data for diffusing event efficiently.

In Fig. 6 (a),  $L$  is source and the event is diffused to  $G_{(4,2)}$ ,  $G_{(4,4)}$  and  $G_{(5,3)}$ ;  $P$ ,  $K$  and  $O$  are the children of  $L$  (①). Next, 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  $face_1$ ,  $A$  constructs virtual link to  $K$  by  $face_1$  (②). Next, the event is diffused to  $G_{(4,6)}$  through void regions. Similarly,  $E$  constructs virtual link to  $K$  and becomes  $A$ 's parent (③-⑤). The data link is shown in Fig. 6 (b).

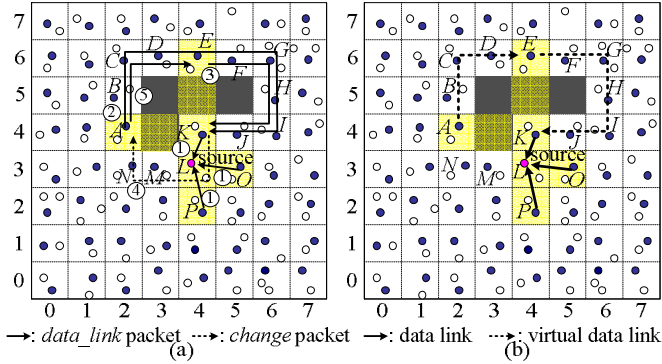


Fig. 6: Constructing the link of continuous event with void

#### IV. EXPERIMENTAL RESULTS

##### A. Simulation Model

Some grid-based data dissemination protocols have been proposed such as CODE [8] that is based on a virtual grid structure. First, this work compares the proposed approach with CODE by ns-2 [6]. Next, the work discusses the number of sinks and the percentage of void grids how to impact the performance. 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 10J. The transmission range of sensor is 112m, and the grid size is 50m. The sensing range of sensor can fully cover with its grid. The data rate is 2m/s and the mobility model is random waypoint model. The sensor network consists of 200 sensor nodes which are deployed uniformly in a  $500m \times 500m$  field (i.e. two sensor nodes per  $50m \times 50m$  grid).

##### B. Simulation Results

###### 1) Aggregating Data of Regular-Shape Range

We assume that the velocity of sink is 2m/s and sink queries and aggregates data with varied range in this simulation. From the results, we observe the delay time and the energy consumed for querying regular-shape range. The *total energy consumption* is the register (data dissemination) and query energy consumption. The *delay time* is the wait time of  $MS$  to obtain data. Fig. 7 (a) shows the delay time. When  $MS$  requests

to acquire data with a bigger range,  $MS$  has to wait the longer delay time, because the source has to collect more data in bigger range. Fig. 7 (b) shows the energy consumption. This is also similar to delay time. Bigger querying range need consume more energy.

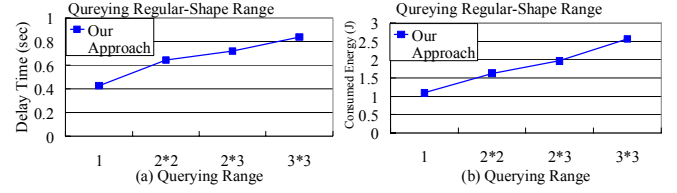


Fig. 7: Data aggregation in varied range

###### 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 in irregular range. We assume that spread range of event is varied from 1 to 9 grids. First, this work compares the energy consumption in data dissemination with CODE. CODE does not support the tracking of diffusing event and each grid head has to run data dissemination when it detects an event. Fig. 8 shows CODE consumes more energy when the spread range becomes great. In proposed approach, a tracking tree is constructed to monitor the diffusing event. Only the source needs to do data dissemination, so this approach decreases a lot of cost in data dissemination. The energy consumption is only slightly increased.

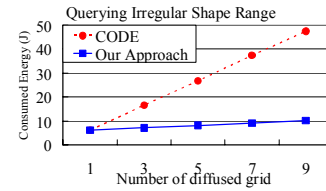


Fig. 8: Energy consumption in data dissemination

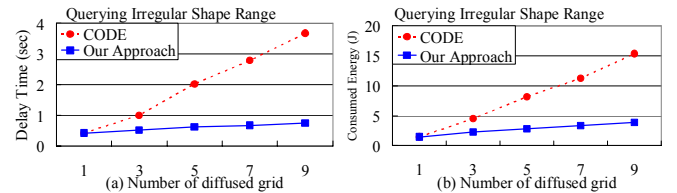


Fig. 9: Data aggregation for irregular querying range

When a sink wants to monitor diffusing event, CODE has to issue multiple queries to collect data. The multiple queries will consume a lot of resource and obstruct network. Therefore, the sink cannot obtain data fast due to network obstructed. That can be observed from Fig. 9 (a). Fig. 9 (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 diffusing event.

###### 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. In addition, eight kinds of different static events are occurred in network simultaneously. The velocity of sink is 2m/s and



the querying range is  $2 \times 2$  grids. The cache mechanism is used to decrease the energy consumption, when multiple sink issued the same query. Fig. 10 (a) shows the average delay time. When the same queries increased, the cache mechanism can decrease the energy consumption for 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 while 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. 10 (b).

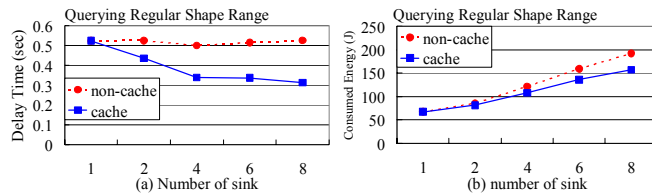


Fig. 10: the impact of cache mechanism

#### 4) Impact of Void Region

We assume 100 sensors are deployed in the network. The density of void grid is varied from 10% to 20% in the network. When the density of void grid is 10%, 10 void grids are in the network. The simulation results show the different percentage of void grids impact. The querying range is  $2 \times 2$  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.

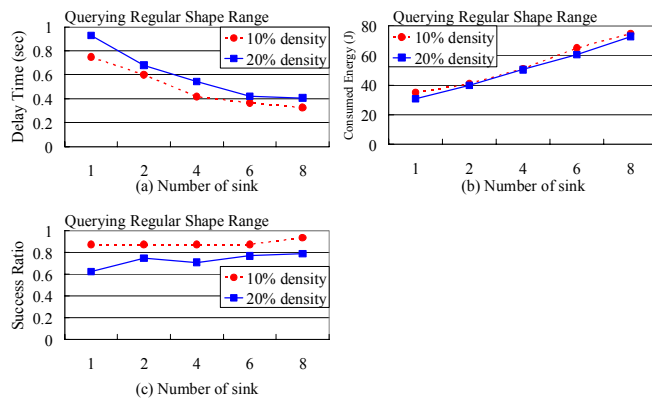


Fig. 11: the impact of void region

When the density of void grid increased, the sink has to wait for the longer time to obtain data as shown in Fig. 11 (a). 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. 11 (b). The success rate in 20% density of void grid is low than that in 10% as shown in Fig. 11 (c). The *success ratio* is the ratio of received data packet to querying. 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 data from the source. The success ratio is decreased while

the density of void grids increased.

## V. CONCLUSIONS

This work presents novel and efficient grid-based approaches to aggregate data for querying range. This work addressed a regular-shape query to collect data on designated rectangle range and an irregular-shape query to monitor and collect data of diffusing event. After performing data dissemination, mobile sink obtains event from grid heads. Mobile sink requests the source to aggregate interested data. Another approach is presented to monitor diffusing event. In addition, cache mechanism is used to solve the same queries issued from multiple sinks. Moreover, face routing is utilized to detect and detour void regions. The simulation results show that the proposed approach not only decreases energy consumption but also increases performances. When the same queries are issued from multiple mobile sinks, the cache mechanism can decrease resource consumption and the sinks can obtain data faster. In addition, the proposed approach handles the routing and data aggregation with void regions.

## REFERENCES

- [1] Y. Hu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. 7th ACM Annual International Conf. Mobile Computing and Networking*, Italy, pp. 70-84, Jul. 2001.
- [2] 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, No. 2, pp. 240-271, Nov. 2005.
- [3] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, "Dissemination: minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks," in *Proc. 1st International Conf. Embedded Networked Sensor Systems*, pp.193-204, Nov. 2003.
- [4] 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 *Proc. 23rd International Conf. Distributed Computing Systems Workshops (ICDCSW'03)*, Providence, Rhode Island, USA. pp. 19-22, May, 2003.
- [5] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. 6th ACM/IEEE International Conf. Mobile Computing and Networking (MobiCom 2000)*, Boston, Massachusetts, pp. 243-254, Aug. 2000.
- [6] VINT Project, "Network Simulator version 2 (NS-2)," Technical report, <http://www.isi.edu/nsnam/ns>, Jun. 2001.
- [7] G. Xing, C. Lu, R. Pless, and Q. Huang, "Impact of sensing coverage on greedy geographic routing algorithms," *IEEE Trans. Parallel and Distributed Systems*, Vol. 17, No. 4, April 2006.
- [8] H. L. Xuan and S. Lee, "CODE: a coordination-based data dissemination protocol for wireless sensor networks," in *Proc. IEEE Intelligent Conf. Sensor Networks and Information*, pp. 14-17, Dec. 2004.
- [9] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "TTDD: two-tier data dissemination in large-scale wireless sensor networks," *Journal of Wireless Networks*, Vol. 11, No. 1-2, Jan. 2005.