WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2012; **12**:445–462
Published online 6 May 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.977

RESEARCH ARTICLE

Decentralized and energy-balanced algorithms for maintaining temporal full-coverage in mobile WSNs

Chih-Yung Chang[1]*, Li-Ling Hung[2], Yun-Jung Lu[1], Sheng-Wen Chang[1] and Yu-Chieh Chen[1]

[1] Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taipei, Taiwan
[2] Department of Computer Science and Information Engineering, Aletheia University, Tamsui, Taipei, Taiwan

ABSTRACT

Coverage is one of the most important issues in Wireless Sensor Networks (WSNs). However, full coverage only can be achieved when surplus mobile sensors contribute a coverage area larger than the hole size. When there is no surplus mobile sensor to cover a big hole, previous studies have utilized mobile sensors by moving the hole from one location to another, therefore achieving temporal full-coverage, where each location on the monitoring region has been ever covered by mobile sensors during a fixed time interval. However, with only some mobile sensors participating in the hole-movement task, this results in an energy-imbalance WSN. This paper considers a mobile WSN that contains a big hole where there exists no redundant mobile sensor to heal the hole. Three distributed algorithms, called *Basic*, *Forward-Only*, and *Any-Direction* movement mechanisms, are proposed to achieve the purpose of temporal full-coverage in a way that the total energy consumption is minimized or that the energy consumption of all mobile sensors that participate in the hole-movement task are balanced. Simulation results reveal that the proposed hole-movement mechanisms enhance the coverage of WSNs and balance the energy consumption of mobile sensor nodes. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS
coverage; energy-balanced; hole-movement; mobile sensor; WSN

***Correspondence**
Chih-Yung Chang, Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taipei, Taiwan.
E-mail: cychang@mail.tku.edu.tw

1. INTRODUCTION

Wireless sensor networks (WSNs) have a wide range of potential applications in environment monitoring, military, smart house, tracking, and remote medical systems. The coverage of the sensing area represents the quality of service (QoS) that is related to the surveillance. Full coverage of a sensing region can be classified into spatial and temporal types. Spatial full-coverage is the one that all sections of the monitoring area can be covered at any given time. The WSN is said to be *temporal full-coverage* if every point in the monitoring region has been ever covered by sensors during a specific period of time. Spatial full-coverage can be reached by dense deployment of static or mobile sensors. Previous works [1] assumed that the WSN is densely deployed with static sensors and addressed an approach that selects a subset of sensor nodes active at any time to reach spatial full-coverage. However, Reference [1] cannot maintain the full coverage as long as the coverage hole appears. In order to remove the drawback of the static sensor, there has been a strong desire to deploy mobile sensors in the WSN in order to reach spatial full-coverage using their mobility support. Researches [2–6] have used mobile sensors to enhance the coverage quality of a monitoring region. This research is differentiated into local and global operations. Researches [2,4] detect their local coverage hole by using *Voronoi diagrams*. Once detected, the mobile sensor locally calculates and moves to the desired position to heal the coverage hole. Studies [5] and [6] used global operations to calculate the size of a big coverage hole and then relocate a group of mobile sensors to heal the hole. However, the above approaches require redundant mobile sensor which is a mobile sensor whose sensing field is overlapped with others over a certain threshold to achieve the purpose of spatial full-coverage.

In addition to spatial full-coverage, other studies [7,8] have discussed how to reach temporal full-coverage in static or mobile WSNs. To prolong network lifetime, Gui and Mohapatra [7] control static sensors' active/sleep mode so that every point in the monitoring region is covered by sensors, thereby reaching temporal full-coverage. However, this algorithm needs enough static sensors to cover the monitoring region. In order to improve this drawback, Reference [8] states that mobile sensors moving continuously

Figure 1. The example of cascaded movement.

can achieve temporal full-coverage. Yet, Reference [8] does not take the power balancing factor into account and would result in an imbalance WSN power consumption.

Energy balancing is an important concern in WSNs [9–11]. How to balance the energy consumption of the network is often regarded as the key issue involved in the protocol design. Wang *et al*. [12] developed a hole-healing mechanism that considers the factor of energy-balancing. A cascaded movement rule is to find some cascading nodes, and use them for relocation to balance the power consumption. As shown in Figure 1, the redundant sensor s_3 intend to heal hole A. Instead of moving to the hole directly, s_1 and s_2 are chosen as cascading nodes. As a result, s_3 moves to replace s_2, s_2 moves to replace s_1, and s_1 moves to the destination. By applying the cascaded movement, sensors s_1, s_2, and s_3 can balance their energy consumptions while the existing hole can be healed. However, it can only heal the hole with a size smaller than a sensor's sensing range and is unable to repair big coverage holes in a WSN. In addition, it also requires redundant mobile sensors to execute the algorithm. In Reference [13], another similar cascaded movement rule is proposed to cover the hole. The sensor that detects the vacant grid finds a trusted node to move into the vacant grid cell by cell. However, the cascaded movement of Reference [13] did not consider energy-balancing since it did not consider the remaining energy of sensors.

This paper observes that moving a mobile sensor in a direction is identical to moving a hole in the opposite direction. We developed three distributed hole-movement strategies for moving an existing big hole in a way that either the total energy consumption is minimized or the power consumption of sensors is balanced. First, the *Basic* hole-movement strategy is proposed to move the hole with the least total power consumption. Second, the *Forward-Only* movement strategy moves the hole by taking both energy-balancing and total power consumption into consideration. Third, the *Any-Direction* movement strategy is proposed by fully considering energy-balancing as its policy. Based on the proposed strategies, whenever a mobile sensor detects a coverage hole within its sensing range, it can locally determine whether or not it should migrate to the hole's location. Finally, a hybrid mechanism that takes full advantages of the proposed hole-movement strategies is proposed so that the hole can be moved in an energy-balanced manner under the given constraint for total energy consumption.

The rest of this paper is organized as follows. Section 2 describes the network environment and problem formulation of this work. Section 3 details the developed hole-movement strategies. Section 4 analyzes the complexity of maximal power consumption of the proposed strategies. The performance evaluation of the proposed mechanism against the existing work is reported in Section 5. Finally, Section 6 concludes the paper.

2. NETWORK ENVIRONMENT AND PROBLEM FORMULATION

2.1. Network environment

The WSN consists of a number of mobile sensors and a sink node. The sink node is placed at the boundary of the monitoring region. All mobile sensors have unique ID respectively and are time synchronized using time-stamp exchange. Each deployed mobile sensor is aware of its location information according to the previous localization studies [14,15] without GPS equipment. Further, each mobile sensor equips the compass which provides the moving direction of the mobile sensor. As a result, the current location of the mobile sensor after moving to a new place can be derived based on the moving direction provided by the compass and the moved distance which can be calculated based on the rotation rate of the mobile sensor's wheels. All mobile sensors will monitor the given region and then transmit their sensing data together with the location information to the sink node through multi-hop communication. According to the received location information, the sink node can identify the location of a coverage hole if there is no sensing information of that location returned by any sensor. Let r_c and r_s respectively denote the communication and sensing ranges of all sensor nodes. In Reference [16], it is proved that ideal deployment is achieved whenever any three neighboring nodes form an equilateral triangle. Under this deployment, the number of sensors required for maintaining full coverage is minimized, and the side of the equilateral triangle becomes $\sqrt{3}rs$. With the optimal deployment, the monitoring region can be treated as a collection of equal-sized hexagon cells, with the center of each cell containing a sensor responsible for monitoring its own cell region. We assume that the communication range r_c is larger than $\sqrt{3}rs$ and all mobile sensors are deployed over a WSN with a hexagonal lattice as shown in Figure 2(a).

The following further explains that deploying all mobile sensors over a WSN with hexagonal lattice is reasonable. Initially, all mobile sensors might be randomly deployed in the WSN. However, in a WSN, the number of holes increases with the working time because more sensors

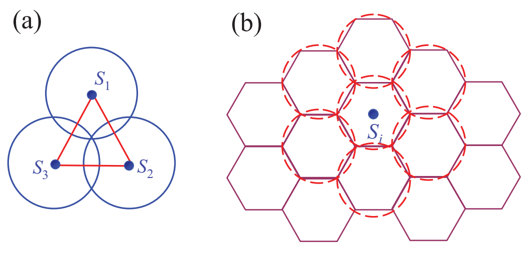


Figure 2. The optimal deployment that minimizes the overlap region but maintains the full coverage.

exhaust their energy for executing sensing and communication tasks. Applying the algorithms proposed in References [3,4] will result in a situation wherein the mobile sensors are closer to each other with similar distances. When the number of mobile sensors equals exactly the minimal number of the sensor nodes that can maintain full coverage of the monitoring area, the distribution of mobile sensors will be like the one shown in Figure 2(b). At this moment, each mobile sensor is located in the center location of each hexagon cell wherein their sensing regions form a hexagonal lattice. Moreover, study [16] showed that hexagon deployment is optimal for covering a plane with circles, since the hexagon arrangement can minimize overlapping areas of sensor nodes. The used network model can also be easily obtained by using a robot to regularly deploy mobile sensors in a monitoring area.

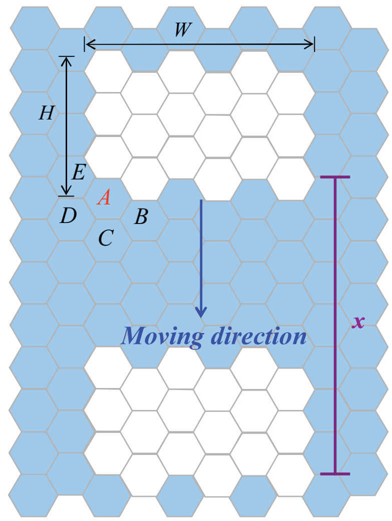


Figure 3. Moving the big hole with size 6 × 3 for a distance *x*.

2.2. Motivation of hole movement

When sensors have operated for a considerable time period, the number of mobile sensors exactly equals the minimal number of sensor nodes that can maintain full coverage of a the monitoring area, and the distribution of mobile sensors will be like the one shown in Figure 2(b). In case a big hole accidentally exists in the WSN at this moment, the algorithms proposed in References [2–4] would not work. The existence of a big hole therefore significantly reduces the performance of sensing and communication tasks.

In such a circumstance, it would be impossible for the sensing region to reach spatial full-coverage. Consequently, it is an important issue to address how to satisfy temporal full-coverage of a sensing region when the number of mobile sensors is not enough to cover the whole sensing region. This paper motivates the coverage problem of applications that require 'hole moving' when there exists no surplus mobile sensor in the WSN. Two application scenarios below show the motivations of hole-movement in the considered network environment. First, consider the object-tracking application, which is one of the most popular issues discussed in previous researches [17]. When an object is close to a coverage hole, we need to move the hole to another location in order to prevent the object from entering the coverage hole. In this application scenario, hole-movement can increase the chances of detecting the target object. Second, the existence of coverage holes will cause some portions of the region to lose sensing functionality. Hole-movement enables the original uncovered region to become covered and some covered regions to become uncovered, hence balancing the sensing precision of the whole sensing region.

2.3. Problem formulation

The following describes the problem formulation of this paper. We consider a big coverage hole exists in a WSN. There are η mobile sensors in the sensing area and $S = \{s_1, s_2, \ldots, s\eta\}$ denotes the set of η mobile sensors. We assume that the number of sensors is not enough to cover the coverage hole. The big hole in this paper consists of lots of small-sized hexagonal cells. The size of the big hole is measured by the total number of the hexagonal cells. More specifically, we assume that the size of the big hole is $W \times H$, where W is larger than H. We further assume that the big hole requires to be moved a distance x in a direction d. Therefore, the hole-movement task T can be represented by $T = (Hole, direction, distance)$, where the *Hole* is represented by the *Hole.location* and *Hole.size* which denote the central location of the *Hole*, and the hole size, respectively. The movement direction and distance of the task T are d and x, respectively. The motivations for hole-movement have been described in Section 2.2. In Figure 3, a big hole is said to be moved a distance of x if the distance between the bottom of the original big hole and the bottom of the moved big hole is x. The value x is a multiple of $\sqrt{3}r_s$ because the communication range of a sensor is $\sqrt{3}r_s$. As shown in Figure 3, the big hole, sized 6×3, is intended to be moved with a distance of $x/\sqrt{3}r_s = 7$ hops. Our goal is to develop hole-movement strategies that can maintain temporal full-coverage by considering energy-balancing or the least total power consumption.

Here, we clarify the term 'energy-balancing'. One important concept in this paper is that the movement of mobile sensors is identical to the movement of the hole. More specifically, when one mobile sensor moves from cell, say a, to cover a small hole cell, say b, the result will be that the small hole cell b moves to cell a. Moving a big hole from location a to location b can only be achieved by the movement of a number of sensors whose sensing coverage can cover the big hole. To minimize each mobile sensor's power consumption in executing a hole-movement task, a good hole-movement policy is the one that distributes the hole-movement task to as much mobile sensors as possible so that the work load of hole-movement on each mobile sensor is minimized. The 'energy-balancing' means that the energy consumption of each mobile sensor participating in the hole-movement task can be minimized.

An energy-balanced index B, which is defined according to Jain's fairness index, is used to measure the degree of energy balance. Let λ denote the number of mobile sensors participating in a task T, and e_i denote the energy consumption of each mobile sensor s_i for all $1 \le i \le \lambda$, respectively. Formula (1) defines the energy-balanced index B, which is normalized between 0 and 1. In case all e_i have same value, the result of the fairness index becomes equal to 1, which is the optimal value. The energy-balanced index of a mechanism approaching 1 indicates that the mechanism provides better fairness in terms of energy consumption.

$$B = \frac{\left[\sum_{i=1}^{\lambda} e_i\right]^2}{\lambda\left[\sum_{i=1}^{\lambda} (e_i)^2\right]} \tag{1}$$

The problem and goal of this paper is formalized below. Given a monitoring region with size $M \times N$, and a deployed WSN which contains s mobile sensors, where s is smaller than the minimal number of sensors required for maintaining full coverage of the WSN, there exists a big hole with size $W \times H$ that should be moved with a distance x in a direction d. The goal of this paper is to develop energy-balanced hole-movement strategies that maximize the energy-balanced index B.

3. HOLE-MOVEMENT STRATEGIES

This section first defines a set of definitions which are used in presenting the proposed protocols. After, the details of the proposed hole-movement strategies are then described.

3.1. Preliminaries and notations

The mobile sensor that detects holes in its sensing range is referred to as the *detector*. The mobile sensor that participates in the hole-movement task is called the *mover*. The set of all movers for executing the same hole-movement task T is denoted by M_T. Furthermore, notation m_{max} denotes the mover with the maximum power consumption and the notation $E(m_i)$ represents the energy consumption of mover m_i when executing the hole-movement task. In addition, notation $s_i(n_j)$ denotes the jth neighbor of sensor i.

In this paper, a hexagon (instead of a circle) is used to represent the coverage of a sensor node; therefore, any sensor s_i has exactly six neighbors $s_i.UL$, $s_i.Up$, $s_i.UR$, $s_i.DR$, $s_i.Down$, and $s_i.DL$ as shown in Figure 4(a), and each sensor has six moving directions $\{UL, U, UR, DR, D, DL\}$, as shown in Figure 4(b). The direction of node u is said to be *movable* if its neighbor, which corresponds to this direction, does not exist. Notation $u.MD$ denotes node u's

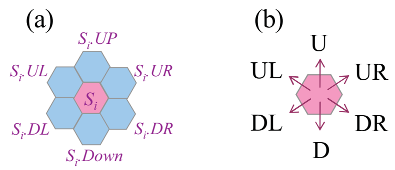


Figure 4. (a) Sensor S_i has six neighbors $S_i.UL$, $S_i.Up$, $S_i.UR$, $S_i.DR$, $S_i.Down$, and $S_i.DL$. (b) Sensor S_i has six moving directions.

movable direction set, which contains all possible moving directions of the detector u.

The detector u finds the directions corresponding to its absent neighbors and records these directions in $u.MD$. Finally, the detector u selects one direction in $u.MD$ to move it one-hop distance. As shown in Figure 3, node A is a detector with six neighbors, $A.UL = E$, $A.Up = hole$, $A.UR = hole$, $A.DR = B$, $A.Down = C$, $A.DL = D$. Consequently, $A.MD$ will be $\{U, UR\}$.

3.2. Energy-balanced hole-movement protocols

The proposed protocol mainly consists of two phases: the *Direction Determination Phase* and the *Hole Moving Phase*. The Direction Determination Phase aims to determine a promising hole-movement direction, which makes sensors consume less energy when moving a hole. We assume that the sink node will decide on a direction d and a distance x of the hole-movement in task. The sink node will apply geographic routing protocol [18] to deliver the hole-moving packet to the boundary nodes of the hole. The hole-moving packet contains the movement direction and distance. The Hole Moving Phase makes the hole move along the direction d and aims to move the hole by minimizing the total power consumption or by balancing the power consumption for all movers in the given WSN.

In the Hole Moving Phase, three distributed hole-movement mechanisms are proposed. First, the *Basic* movement mechanism enables each mover to directly move along the opposite direction of d. Though the *Basic* movement mechanism minimizes total energy consumption, it results in imbalance energy consumption. The other two mechanisms aim to balance energy consumption. They first decompose the original big hole into lots of small-sized hole cells. Then the *Forward-Only* movement mechanism moves each small hole cell forward along direction d, while the *Any-Direction* movement mechanism enables some small hole cells to move some steps backwards of direction d to maximize the value of energy-balanced index B.

3.2.1. Direction determination phase.

The Direction Determination Phase is mainly executed by the sink node. The sink periodically collects the sensing information from each sensor. Since the information returned by each sensor contains its reading and the corresponding location, the sink node can identify a hole's

Wirel. Commun. Mob. Comput. 2012; **12**:445–462 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

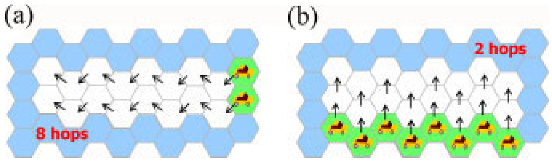


Figure 5. (a) Holes move one hop along the direction which is perpendicular to the shorter side of the coverage hole. (b) Holes move one hop along the direction which is perpendicular to the longer side of the coverage hole.

location if there is no reading of that location returned by any sensor. Therefore, the sink node is aware of the hole information, including the hole size and the hole boundary, without any additional communication overhead. The objective of the Direction Determination Phase is for the sink node to determine a movement distance *x* and a promising movement direction *d*, which can make sensors consume less energy when executing hole-movement tasks.

The sink node will calculate the length of each side of the big hole and choose an appropriate hole-moving direction *d* which is perpendicular to the longer side of the hole. In addition, the sink node also needs to record the location information of the detector, which is located in the middle of the longer side of the coverage hole. Let this location information be the middle location. Finally the sink node broadcasts a hole-moving packet to all sensors in the WSN, which contains the hole-moving direction and the middle location. For example, the hole moving direction shown in Figure 5(b) is better than that of Figure 5(a) since each mobile sensor consumes less energy if the hole moves along the south direction, which is perpendicular to the longer side of the coverage hole.

3.2.2. Hole moving phase.

The Hole Moving phase aims to move the big coverage hole by either minimizing the total energy consumption or by balancing the power consumption of the sensors. There are two goals in the designing of the Hole Moving Phase, both of which have different perspectives.

(1) *Minimize the total power consumption of the hole-movement task. That is,*

$$\min \sum_{mi \in M} E(mi)$$

(2) *Minimize the maximal power consumption of the mover* $m_{\max}$ *in order to balance the power consumptions of all movers. That is,*

$$\min E(m_{\max})$$

In this phase, we present three distributed hole-movement mechanisms, named *Basic*, *Forward-Only* and *Any-Direction* movement mechanisms, for hole-movement.

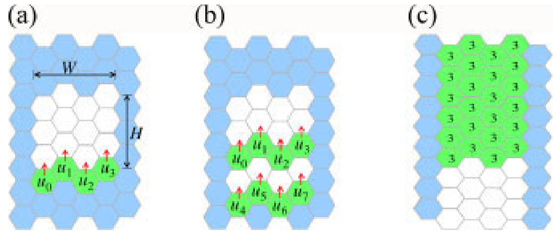


Figure 6. An example of the *Basic* movement mechanism: (a) Detectors nearby the longer side of the big coverage hole play the mover role; (b) after mover u_i, $0 \le i \le 3$, moves, detectors u_i, $0 \le i \le 7$, find the '*Up*' direction in their *MD* and then switches to *moving* status; and (c) the *Basic* mechanism ends when the big coverage hole moves for a distance *x*.

3.3. Basic movement mechanism

Recall that the sink will determine a direction *d* for hole-movement in the Direction Determination Phase. Let $\overline{d}$ denote the opposite direction of *d*. The basic idea of the *Basic* movement mechanism is quite straightforward. It enables the sensors nearby the longer side of the big coverage hole to play the mover role and move along the direction $\overline{d}$. After all of the sensors receive the hole-moving request from the sink node, the *Basic* movement mechanism will then be initiated. During the execution of the task *T*, each detector has two statuses. One is a '*detecting*' status and the other is a '*moving*' status. When a sensor becomes a detector, the detector stays in the detecting status and checks whether the direction $\overline{d}$ is in its *MD*. If yes, the detector switches to moving status and then moves toward the direction $\overline{d}$. Figure 5 illustrates how the *Basic* movement mechanism works. A big hole with size 4×3 exists in the WSN. The big hole will move toward the '*Down*' direction according to the decision made in the Direction Determination Phase. Therefore, the detector, whose *MD* has the '*Up*' direction will change its status to moving and move one hop distance along the '*Up*' direction to help the hole cell migrate toward the '*Down*' direction. As shown in Figure 6(a), detectors u_i, $0 \le i \le 3$, will switch from detecting to moving status since their $u_i.MD$ contains the '*Up*' direction. For example, we have $u_1.MD = \{\text{UL, U, UR}\}$. Therefore detector u_1 will move one hop distance along the '*Up*' direction to help the hole cell migrate toward the '*Down*' direction. After the movers move upward one hop, the big coverage hole will then be decomposed. After that, detectors u_i, $0 \le i \le 7$, will discover that their *MD*s contain the '*Up*' direction; hence, they will switch to moving status and subsequently move for a distance of one hop, as shown in Figure 6(b). The same operation will execute repeatedly until the big coverage hole moves for a distance *x*, as shown in Figure 6(c). The number labeled on the sensor represents how many hops the mover has walked. All movers that participated in the hole-movement task using the *Basic* movement mechanism have moved three hops.

The *Basic* movement mechanism can be effective in minimizing the total power consumption of all movers because

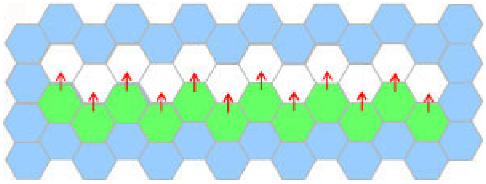


Figure 7. An example of the *Basic* movement mechanism with a 12×1 size.

the decomposed small hole cells always directly move toward the direction d. However, the power consumption of each mover increases along with the hole height. An obvious drawback here is that each mover will consume significant energy if the height of hole is large and the value of the energy-balanced index B is decreased accordingly, which would indicate that the power consumption of all sensor nodes in the WSN is imbalance.

We noticed that the movers can consume less energy in a hole-movement task as long as the height H of the coverage hole is minimized. Figure 6 shows an example that verifies this argument. We consider a big hole with a 12×1 size, which equals the size of the hole shown in Figure 7. Using the *Basic* movement mechanism, each mover only moves one hop when executing a hole-movement task, while the total power consumption of all movers is identical to that in Figure 7. Using Formula 1, we conclude that, under the fixed total power consumption of movers, the more movers there are that participate in the hole-movement task, the more energy balanced all sensors in the WSN become. Based on this, we are motivated to change the hole shape into a belt to enable more movers to share the workload of the hole-movement task. Two mechanisms, called *Forward-Only* and *Any-Direction* movement mechanisms, are proposed to change the hole shape into a belt during the execution of a hole-movement task.

3.4. Forward-only movement mechanism

The *Forward-Only* movement mechanism moves a hole by taking both energy-balancing and total power consumption into consideration. Changing the hole shape into a belt enables more movers to share the workload of the hole-movement task and thus minimizes the maximal power consumption of each mover. The *Forward-Only* movement mechanism aims to simultaneously move a big hole forward along direction d and change the hole shape into a belt. This mechanism decomposes the original big hole into lots of smaller hole cells, with each hole cell moving forward along direction d. Sensors in the WSN can only move along one of the three directions that are closest to direction $\overline{d}$. For example, if the sink determines that the hole-moving direction d is '*Down*', each sensor in the WSN can only move along the '*UL*', '*Up*', or '*UR*' directions which are closest to the '*Up*' direction. Figure 8 depicts the moving trajectories of all hole cells by applying the *Forward-Only* movement mechanism on the big hole sized 4×3.

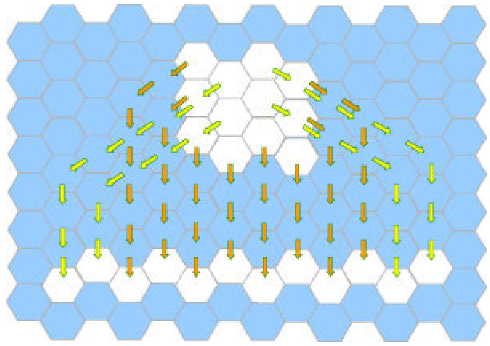


Figure 8. The moving trajectory by applying the *Forward-Only* movement mechanism on the big hole with a 4×3 size.

In order to simultaneously move the big hole and change the hole shape into a belt, the developed movement strategies must satisfy the following three properties:***P1*:** Distributing the hole cells into different columns.***P2*:** The detector that has the maximal remaining energy moves first.***P3*:** A hole is moved only forward along the three directions closest to direction d.

A hole with a belt shape will be good at balancing movers' power consumption. To change the shape of the big hole into a belt, property *P1* is proposed to distribute the hole cells to different columns. To implement property *P2*, each mover maintains a hop value that represents how many hops the mover has moved. The detectors, which have smaller hop values, then help the hole cells migrate in the WSN to minimize the maximal power consumption of movers and obtain a better energy-balanced index B. Property *P3* takes the total power consumption into consideration.

The *Forward-Only* movement mechanism is executed round by round. Here, a one-time computation and one-time movement are applied to relocate the movers' one hop. As shown in Figure 9, detectors perform three actions in each round. Once a sensor becomes a detector, it exchanges its hop information with its six neighbors in the Exchange Hop Information (EHI) section. In the EHI section, each detector is aware of the relative locations of the hole cells and obtains its neighboring detectors' hop information. Next, each detector checks the following rules from *Rule 1* to *Rule 2* and then determines a movement direction in the Determine Movement Direction (DMD) section. If none of these rules are satisfied, the detector will remain stable this round. Otherwise, the detector will play the mover role and move toward its desired direction for a distance of $\sqrt{3}r_s$ in the Moving section. After that, each mover exchanges location information including its own location and the locations of neighboring hole cells. Once a mover has moved toward its desired direction, it increases the hop value by one. The

Figure 9. A round consists of three sections: (1) exchange hop information, (2) determine moving direction, and (3) moving.

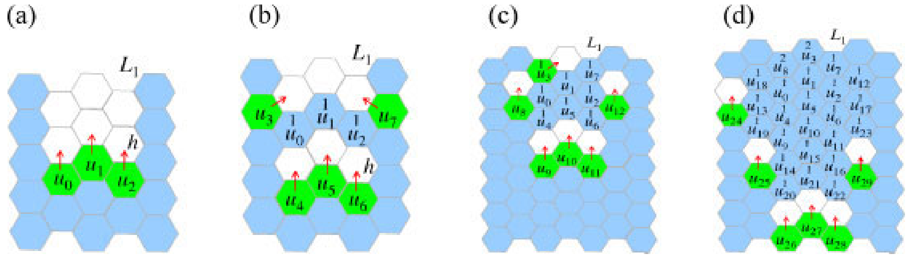


Figure 10. An example for illustrating the *Forward-Only* movement mechanism.

round ends at the end of the Moving section. When one round ends, the next round starts and all detectors refresh and again determine whether it is a detector on this round.

The following proposes two checking rules so that detectors can determine whether or not it should change its status from a detector to a mover and which direction it should move. For simplicity, we assume the big hole will move toward the '*Down*' direction. Each hole cell therefore only has three candidate directions: '*DL*', '*Down*', or '*DR*'. To avoid two movers moving to the same hole cell at the same time, the detector, say *s*, which is located below the hole cell, will be the arbitrator for deciding the movement direction of the hole cell in this round. In other words, the arbitrator needs to choose one mover from its neighboring detectors *s.UL*, *a* and *s.UR* to heal the hole cell that is located above its location.

in the left side while lines 6–10 discuss the circumstance when the arbitrator is in the right side. Once an arbitrator *s* detects any hole cell in its '*Up*' direction, it checks whether its location is in the left side or the right side of the big hole. If *s* is in the left side, it compares its hop value with the neighboring detector *s.UL* and then chooses a detector which has a smaller hop value to move, as shown in lines 1–5 of the *Rule 2* algorithm. Conversely, when the arbitrator is in the right side, it selects a detector which has a smaller hop value as the mover from neighboring detector *s.UR* and itself. Figure 7 indicates that the moving trajectory of hole cells can be seen by applying the *Forward-Only* movement mechanism. The number of the hole cells that walk to the left and right is balanced.

Below is a series of four diagrams illustrating how the *Forward-Only* movement mechanism works by applying the two rules. For simplicity, we use a hole sized 3×2 as an example. In Figure 10(a) u_0, u_1, and u_2 employ *Rule 1* to determine their *moving direction* = '*UP*'. After the movement, the hop value of sensor u_2 becomes 1. When the sensor u_2 detects another hole cell in its '*Up*' direction again, it applies *Rule 2* to choose its neighbor $u_2.UR = u_7$ to heal that hole cell. After u_7 moves according to *Rule 2*, it carries a hop value $u_7.hop = 1$ and locates itself on the column L_1 where the sensor u_2 located. Then, u_1 applies *Rule 2* and selects u_3 to move toward '*UR*', as shown in Figure 10(c). Other movers, which are labeled by the arrowhead symbol, use *Rule 1* to move upward directly. Consequently, as shown in Figure 10(d), all hole cells that constitute a big hole will

Rule 1:	**Rule 2:**
Let *s* be the arbitrator 1. **if** (*s.Up==hole&& s.hop== 0*) 2. moving direction = '*Up*' 3. **end if**	Let *s* be the arbitrator **Left side rule /*arbitrator located at the left side of the hole */** 1. **if** (*s.hop >= s.UL.hop*) 2. *s.UL.* moving direction = '*UR*' 3. **else** 4. *s.* moving direction = '*Up*' 5. **end if** **Right side rule/*arbitrator located at the right side of the hole */** 6. **if** (*s.hop >= s.UR.hop*) 7. *s.UR.* moving direction = '*UL*' 8. **else** 9. *s.* moving direction = '*Up*' 10. **end if**

Figure: Pseudo codes of rules 1 and 2 of the *Forwarding-Only* movement mechanism.

Rule 1 is mainly designed for the arbitrator that has never moved, whereas *Rule 2* is designed for the arbitrator that has moved. Since the most efficient movement is to directly move the hole toward the direction *d*, *Rule 1* is designed accordingly. By applying *Rule 1*, detector *s*, which has never moved and detects *s.Up* = hole, will select the '*Up*' direction as its moving direction. *Rule 2* is mainly designed for the arbitrator who has moved. The detector, which has moved and detects *s.Up* = *hole*, will choose an appropriate neighboring detector to move based on the number of movement hops. Detector *s* would send a packet to inform the selected neighbor about its decision. Lines 1–5 of the *Rule 2* algorithm discuss the circumstance when the arbitrator is

```
Protocols: Forward-Only movement
// executed in each round
// computation for moving direction
If ( s is a detector )   /* detects holes in its sensing range */
        exchange hop information with neighbors
        Rule 1 is applied  /* try 'up' direction if s has never moved */
         If (s.moving direction = NULL)
            Rule 2 is applied /* for those detector who has moved */
            If (s.moving direction = NULL)
                sends packet to the selected neighboring detector
            end If
         end If
// actually moving
        If(s.moving direction ≠ NULL)
             move along the moving direction
         end If
    end If
Figure 11: Pseudo code of the Forward-Only movement mechanism.
```

Figure 11. Pseudo code of the *Forward-Only* movement mechanism.

be distributed into different columns, and there would only be one hole cell in each column. At this time, the shape of the big hole would be like a belt. From now on, the big hole will be migrated straight along the '*Down*' direction in an optimal energy-balanced manner.

Figure 11 is a pseudo code for summarizing the rules developed for the *Forward-Only* movement mechanism. *Rule 1* has a higher priority than *Rule 2*. Once a detector determines its moving direction in *Rule 1*, it will not change its decision even if it receives another detector's informed packet later.

3.5. Any-direction movement mechanism

As the name indicates, each hole cell can move along six different directions, which include '*UL*', '*Up*', '*UR*', '*DR*', '*Down*' or '*DL*' in the *Any-Direction* movement mechanism. In this mechanism, we only extremely take the energy-balancing into consideration for hole-movement. Similar to the *Forward-Only* movement mechanism, the *Any-Direction* movement mechanism aims to simultaneously move the big hole forward along the direction *d* and change the hole shape into a belt. The *Any-Direction* movement mechanism also decomposes the original big hole into lots of smaller hole cells, with each hole cell moving along any direction, though preferring direction *d*. This mechanism enables some small hole cells to move some steps back toward the direction *d* and therefore balances the power consumption of all movers. Figure 12 depicts the movement trajectories that apply the *Any-Direction* movement mechanism to the hole-movement task for a big hole sized 4×3.

To simultaneously move the big hole and change the hole shape into a belt, the developed movement strategies must satisfy properties *P1* and *P2, which are* designed in the *Forward-Only* movement mechanism. Notice that the *Any-Direction* movement mechanism does not adopt property *P3* designed in the *Forward-Only* movement mechanism. Because the *Any-Direction* movement mechanism allows some hole cells to move some steps back toward direction *d*, the maximal power consumption of a mover can be further reduced, since more movers participate in the hole-movement task. However, the total power consumption is increased, since some hole cells move back toward the direction *d*.

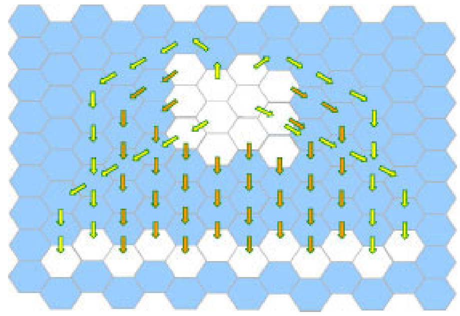


Figure 12. Movement trajectory that applies the *Any-Direction* movement mechanism on a big hole sized 4×3.

The *Any-Direction* movement mechanism is also executed in rounds. A one-time computation and one-time movement policy is applied to relocate the sensors. As shown in Figure 9, detectors perform three actions each round. The main difference from the *Forward-Only* movement mechanism lies in the checking rules in the Determine Movement Direction (DMD) section.

The following proposes five checking rules for detectors to determine whether or not a detector should change its status from a detector to a mover, and which direction it should move. Here, we assume the big hole will move toward the '*Down*' direction. Similar to the *Forward-Only* movement mechanism, in the procedure of changing the hole shape into a belt, the *Any-Direction* movement tries to make the hole cells in the left and right parts of the big hole move toward the left and right, respectively, to balance the number of hole cells walking to the left and right as shown in Figure 12. Next, we define the priority of hole cell movement direction at first and then illustrate the five rules and, after, make them match the priority.

Consider a hole cell located at the left side of the big hole where there exists a moved sensor below the hole cell. Figure 13 shows the moving direction priority of the hole cell. The number labeled on the detector represents the priority. They are '*h.DL*', '*h.UL*', '*h.Down*', and '*h.Up*' in order. Based on *P1*, we prefer to distribute the hole cells of the same column into different columns. Since the movement direction *d* is '*Down*', we make '*DL*' direction as the first movement priority for the hole cell. The '*UL*' direction is the second priority because when the hole cell moves toward '*UL*', it can migrate to a different column. Since the hole cell moving along the '*Up*' direction increases the total power consumption of a hole-moving task, the '*Up*' direction has last priority in the hole cell movement. Fortunately, because the detector's moving direction is opposite to the hole cell moving direction, the detectors that correspond to these four moving direction priority for the hole cell *h* will become the four candidates of the mover for this hole cell.

Wirel. Commun. Mob. Comput. 2012; **12**:445–462 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

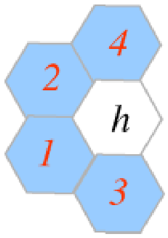


Figure 13. The movement priority of the hole cell *h*, which belongs to the left side of the big hole, is labeled on each detector.

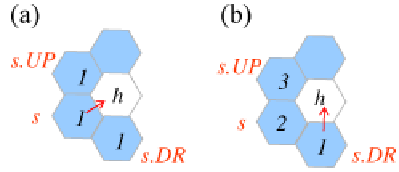


Figure 14. (a) A case where the detector *s* moves toward the '*UR*' direction. (b) A case where the detector *s.DR* moves toward the '*Up*' direction. This is because it has the minimal hop value among *s*' neighboring candidates.

The movement priority for the hole cells on the right side of the coverage hole is similar to that of hole cells on the left side. Therefore, the movement priority order of hole cells on the right side is '*h.DR*', '*h.UR*', '*h.Down*', and '*h.Up*', in this order.

In the *Any-Direction* movement mechanism, a detector that has the maximal remaining power should help the hole cell migrate, even if the detector is located above the hole cell. To avoid more than one mover migrating to the same hole cell, the movement direction of a hole cell is determined by the negotiation among the detectors located around the same hole. For example, in Figure 13, the four detectors negotiate and determine a mover by themselves. The five rules designed in the *Any-Direction* movement mechanism implement the priority and the process of negotiation mentioned above.

Rule 1 is designed for the detector that is located below a hole and that has never moved. Since the most efficient movement is to directly move the hole toward the direction *d*, *Rule 1* is designed accordingly.

Rules 2 is designed for the detector that is located above the hole cell. When a detector *s* above a hole cell *h* simultaneously has the '*DL*', '*Down*' and '*DR*' direction in *s.MD*, it indicates that none of its neighboring detectors can help the hole cell *h* migrate. Therefore we enable the detector *s* to move downward, helping the hole cell *h* move along the upward direction. In the hole-movement procedure, we prefer that the hole cells moves '*Down*'. For this reason, the mechanism allows detectors belonging to the down side of the big hole to have one more row to move upward compared to detectors belong to the up side of the big hole. This is why we set the detectors can only move downward when its hop value is less than half of the height of the big hole minus one.

Rule 3 — Left side rule	Right side rule
if (*s.UR==hole*)	**if** (*s.UL==hole*)
if (*s.hop* <= *s.Up.hop* or *s.Up is absent*)	**if** (*s.hop* <= *s.Up.hop* or *s.Up is absent*)
if (*s.hop* <= *s.DR.hop* && *s.DR.hop* ≠ 0 **or** *s.DR is absent*)	**if** (*s.hop* <= *s.DL.hop* && *s.DL.hop* ≠ 0 **or** *s.DL is absent*)
s. moving direction = '*UR*'	*s*. moving direction = '*UL*'
end if	**end if**
if(*s.hop* > *s.DR.hop*)	**if**(*s.hop* > *s.DL.hop*)
s.DR. moving direction = '*Up*'	*s.DL*. moving direction = '*Up*'
end if	**end if**
end if	**end if**
end if	**end if**

Figure: The pseudo codes of rule 3 (including left and right side rules) of the *Any-Direction* movement mechanism.

Rule 3 is designed for the detector that is located on the down-left or down-right locations of the hole cell *h*. Consider that detector *s* is located on the left side of the big hole. As shown in Figure 14(a), if the detector *s* has the '*UR*' direction in *s.MD*, it has the highest priority for helping the hole cell *h* move '*DL*'. Therefore, when its hop value is less than the neighboring detector *s.Up*, which is above *s*, it indicates that either the detector *s* or the detector *s.DR*,

Rule 1	Rule 2
if (*s.Up==hole* && *s.hop== 0*)	**if** (*s.Down==hole*)
s.moving direction = '*Up*'	**if** (*s.hop* < $\lceil width/2 \rceil - 1$ && *s.DL==hole* && *s.DR==hole*)
end if	*s*.moving direction = '*Down*'
	end if
	end if

Figure: The pseudo codes of rules 1 and 2 of the *Any-Direction* movement mechanism.

which is located below the hole cell *h*, has to move to the location of hole cell *h*. At this time, the detector compares its hop value with the hop value of the neighboring detector *s.DR*. The one which has the smaller hop value is selected to be the mover.

Consider line 3 of *Rule 3*. When the hop value of the neighboring detector *s.DR.hop* is equal to zero, the detector *s* would not be able to move this round. That is because the neighboring detector *s.DR* has decided its moving direction in *Rule 1*.

Figure 14 shows two examples that illustrate how *Rule 3* works. Here we assume that the hole cell *h* is in the left side of the coverage hole in Figure 14. By applying *Rule 3*, when $s.hop <= s.Up.hop$, the detector *s* decides whether its neighboring detector *s.DR* or itself will move to the hole cell's location. As shown in Figure 14(a), detectors *s* and *s.DR* have the same hop value. Therefore, the detector *s* becomes the mover, and migrates to the hole cell's location. Conversely, if *s.DR* has a smaller hop value compared to *s*, the detector *s.DR* becomes the mover to heal the hole cell *h*.

Rule 4 is designed for a detector that is located on the left-up or right-up locations of the hole cell *h*. Consider a detector *s* located at the left side of a big hole. As shown in Figure 15(a), if the detector *s* has '*DR*' in *s.MD*, it has the second priority to help the hole cell *h* move along the '*UL*' direction. Therefore, when its hop value is less than the neighboring detector *s.Down*, which is below *s*, this indicates that either the detector *s* or the neighboring detector *s.UR*, which is located above the hole cell *h,* can move to the location of hole cell *h*. In this case, the detector *s* compares its hop value with that of the neighboring detector *s.UR*, and then selects the one that has a smaller value to be the mover.

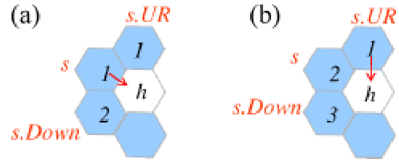


Figure 15. (a) A case where the detector *s* moves toward the '*DR*' direction. (b) A case that the detector *s.UR* moves toward the '*Down*' direction. This is because that it has the smallest hop value among *s*' neighboring candidate.

Figure 15 shows two examples that illustrate how *Rule 4* works. We assume that the hole cell *h* is in the left side of the coverage hole. By applying *Rule 4*, when $s.hop < s.Down.hop$, the detector *s* decides whether its neighboring detector *s.UR* or itself will move to the hole cell's location. As shown in Figure 15(a), detectors *s* and *s.UR* have the same hop value. Therefore, the detector *s* becomes the mover and then moves along the '*DR*' direction. Conversely, if detector *s.UR* has a smaller hop value than *s*, the detector *s.UR* becomes the mover and then migrates to the hole cell *h*' location, as shown in Figure 15(b).

Rule 5 is mainly designed to prevent the hole from doing the back-and-forth movement. Since the hole cell and the detector can move along six directions in the *Any-Direction* movement mechanism, *Rules 1–4* may cause the hole cell to move back and forth. The following lists the forbidden movement directions associated with the last moving direction.

Rule 5: Detector *s*, which has moved, before is forbidden to move by the directions listed in Table I this round, according to the last moving direction.

```
Rule 4
Left side sub-rule
1. if ( s.DR==hole )
2.     if (s.hop < s.Down.hop or
         (s.Down is absent && s.hop==0))
3.       if (s.hop <= s.UR.hop && s.UR.
            moving direction ≠ 'Down' or
            s.UR is absent)
4.          s. moving direction = 'DR'
5.       end if
6.       if (s.hop > s.UR.hop)
7.          s.UR. moving direction = 'Down'
8.       end if
9.     end if
10. end if

Right side sub-rule
11.  if ( s.DL==hole )
12.     if (s.hop < s.Down.hop or
           (s.Down is absent && s.hop==0))
13.       if (s.hop <= s.UL.hop && s.UL.
             moving direction ≠ 'Down'
             or s.UL is absent)
14.          s. moving direction = 'DL'
15.       end if
16.      if (s.hop > s.UL.hop)
17.         s.UL. moving direction = 'Dow'
18.      end if
19.    end if
20.  end if
```

Figure: The pseudo codes of rule 4 (including left and right side sub-rules) of the *Any-Direction* movement mechanism.

Consider line 2 of *Rule 4*. When the neighboring detector *s.Down* does not exist, the detector *s* can only move when its hop value is zero. This is because that the absence of *s.Down* results in the detector *s* being unable to know whether or not the detector below the hole cell *h* also intends to heal the hole *h*.

Figure 16 is used as an example to illustrate how the *Any-Direction* movement mechanism works by applying the proposed five rules. For simplicity, we use a hole sized 3×2 as an example. In the first step, detectors u_0, u_1, and u_2 apply *Rule 1* to determine their *moving direction* = '*Up*' and the detector u_3 applies *Rule 2* to decide its *moving*

Table I. The forbidden movement direction associated with the last moving direction.

The last moving direction	The forbidden movement directions
Up	Down
up-left	Up-right and down-right
up-right	Up-left and down-left
down-left	Up-right and down-right
down-right	Up-left and down-left
Down	Up-left, up and up-right

direction = '*Down*'. Therefore, detectors u_0, u_1, and u_2 move upward and detector u_3 moves downward on the first step, as shown in Figure 16(a).

After, u_5 and u_{10} apply *Rules 3* and *Rule 4* to determine their *moving directions*, '*UR*' and '*DL*', respectively, as shown in Figure 16(b). Here we find that, if the detector u_1 applies *Rule 2*, it can make the decision of *moving direction* = '*Down*'. However, the detector u_1 remains unmoved this round due to the restriction of *Rule 5*. Since u_1 moved upward in the first round, it can't move downward this round. Executing the algorithm repeatedly, the big hole will finally be decomposed into six hole cells that are distributed to different columns, as shown in Figure 16(c). At this step, the shape of the big hole is now like a belt. The big hole would now migrate along the 'Down' direction until the movement distance *d* is reached. Figure 17 gives a pseudo code which summarizes the rules designed for the *Any-Direction* movement mechanism.

```
Protocols: Any-Direction movement
// This protocol is executed in each round
// Computation
if ( s is a detector )
    exchange hop information with neighbors
    Rule 1 is used
    if (s.moving direction = NULL)
        Rule 2 is used
        if (s.moving direction = NULL)
            Rule 3 is used
            sends packet to selected
             neighboring detector
        end if
        if (s.moving direction = NULL)
            Rule 4 is used
            sends packet to selected
            neighboring detector
        end if
    end if
    if (s.moving direction ≠ NULL)
        Rule 5 is used
     // Actually Moving
        if(s.moving direction ≠ NULL)
            move along the moving direction
        end if
    end if
end if
```

Figure 17. Pseudo code of the *Any-Direction* movement mechanism.

4. COMPLEXITY ANALYSIS OF MAXIMAL POWER CONSUMPTION

This section analyzes the maximal power consumption of movers and the total power consumption for executing a hole-movement task. For simplicity and for not having loss in generation, we consider that the big hole is regular and should be moved along the direction *d* for a distance *x*.

In the *Basic* movement mechanism, the maximal power consumption of all movers is quite related to the height of the coverage hole. If we intend to move each hole cell in the big hole by one hop, the movers that below the bottom of the big hole need to move $Height/\sqrt{3}r_s$ hops. After the big hole moves *x* hops, each mover that participated in the hole-movement task moves $Height/\sqrt{3}r_s$ hops, as shown in Figure 6. As a result, by applying the *Basic* movement mechanism, the maximal power consumption of movers can be evaluated by Formula (2).

$$E(m_{\max}) = Height/\sqrt{3}r_s \times P \tag{2}$$

In the *Forward-Only* movement mechanism, the trajectories of the holes are characterized by the following facts. First, the hole cells in the bottom row of the big hole move downward according to the application of *Rule 1*. By applying *Rule 2*, some hole cells that belong to the left side of the big hole move along the '*DL*' or '*Down*' directions, while the hole cells that belong to the right side of the big hole move along the '*DR*' or '*Down*' directions. By applying the *Forward-Only* movement mechanism, the migration order of the hole cells leaving from their initial location accord to the *V-layer*, as shown in Figure 18. In Figure 18, when the mechanism is applied in the first round, hole cells that are located on the bottom of the big hole are moved first. In the second round, the hole cells lying on the *V-layer*, labeled '2nd', are moved accordingly. In the third round, the hole cell lying on the *V-layer*, labeled '3rd', are moved subsequently. The rest may be deduced by analogy. Let h_{last} denote the last moved hole. Finally, the h_{last} is located on the middle of the top of the big hole, which is labeled h_{last}. From this observation, we see that the closer a detector is to the bottom side of the big hole, the earlier it can help the hole cell move. After the big hole is moved one hop, the big

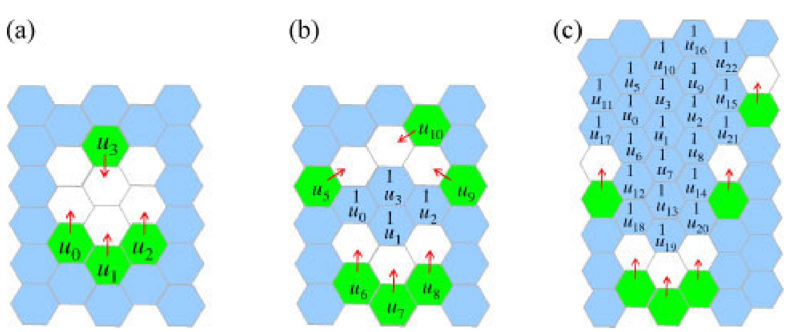


Figure 16. An example of the *Any-Direction* movement mechanism.

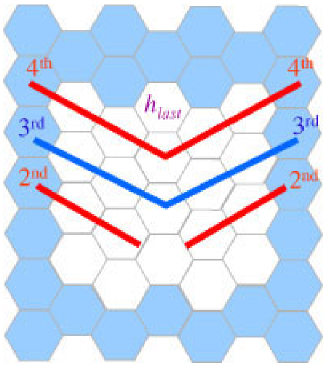


Figure 18. The migrating order of hole cells when they leave from their initial locations of a big hole sized 5 × 4.

hole is also divided into *Top* and *Down* parts. The detectors that are closer to the bottom-left or bottom-right sides of the *Top* part also help the hole cell move early. Based on this idea, the hole cell that is moved last will be moved by the detector that is located on the bottom side of the big hole or the bottom-left/right side of the *Top* part, and this detector will consume the maximal power among all the movers.

We analyze the moving trajectory of the sensors to find one that helps the last moved hole cell migrate. Since the hole cell in the *Forward-Only* movement mechanism can only move along the '*DL*', '*Down*' or '*DR*' directions, the last moved hole cell may be migrated by the detector that is at the bottom-left/right side of the *Top* part or the bottom side of the big hole. For simplicity, we only discuss the detector located on the left side of the big hole. Therefore, the last moved hole cell may be moved by either the detector that located on the bottom-left side of the *Top* part or the detector that lies on the bottom side of the big hole. Here, two cases are given to represent this argument. Figure 19(a) and (b) illustrate that, when the big hole's *Width/2* < *Height*, the last moved hole cell is healed by the detector, *B* which is located on the bottom-left side of the *Top* part of the big hole. On the other hand, Figure 19(c) and (d) illustrate that, when the big hole's *Width/2* > *Height*, the last moved hole cell is migrated by the detector *E*, which is located on the bottom side of the big hole.

Now we aim to compute the maximal power consumption of the movers in the *Forward-Only* movement mechanism. Using the above analysis, we can derive that the mover that consumes the maximal power consumption among all movers lies on the bottom side of the big hole or the bottom-left/right side of the *Top* part with a different ratio of *Height* to *Width*. To get the maximal power consumption of the

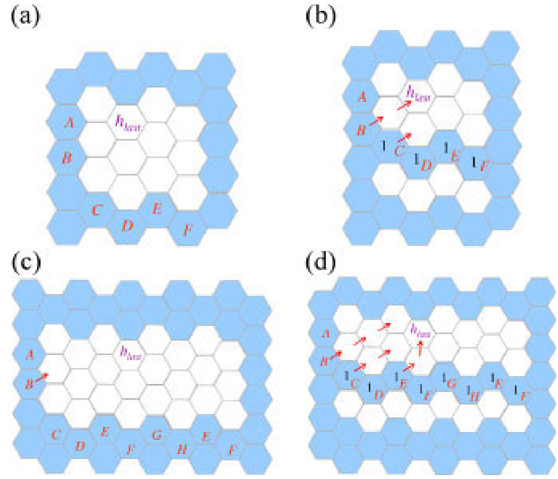


Figure 19. Two examples showing which detectors would help the last moved hole cell migrate.

movers, we only have to evaluate the number of hops that the detector walks from the bottom side of the big hole, or the bottom-left/right side of the *Top* part to the location of the last moved hole cell h_{last}. The detector that lies on the bottom left side, walking to the location of the last moved hole cell h_{last}, would have spent $\lceil Height/2 \rceil / \sqrt{3} r_s$ hops on moving. On other hand, the detector that lies on the bottom side, walking to the location of the last moved hole cell, would have moved $Height/\sqrt{3} r_s$ hops. Therefore, after the big hole moves x hop, the value of the maximal power consumption of movers that apply the *Forward-Only* movement mechanism is given in Formula (3).

$$E(m_{\max}) = \begin{cases} \left(\left\lceil {}^{Width}\!/_{2} \right\rceil / \sqrt{3} r_s \right) \times P, & \text{if } (Width/2 < Height) \\ (Height/\sqrt{3} r_s) \times P, & \text{if } (Width/2 > Height) \end{cases} \quad (3)$$

Similar argument for deriving the maximal power consumption can be applied to the *Any-Direction* movement mechanism. In the *Any-Direction* movement mechanism, the hole cells belongs to the left side of the big hole will move toward the left and the hole cells belong to the right side of the big hole will move toward the right. In the *Any-Direction* movement mechanism, we limit the moving hops of the mover that intends to move along direction $\overline{d}$. This constraint helps prevent many hole cells from moving backward, which violates the goal of minimizing the total power consumption. Therefore, the last moved hole cell is located at the center of the big hole. The detector that spends the most power consumption on moving will locate on the bottom side of the big hole or the bottom-left/right side of the *Top* part, and the detector only has to move to the location at the center of the big hole. Hence, the maximal power consumption of all movers that apply the *Any-Direction* movement mechanism is given in Formula (4).

$$E(m_{\max}) = \begin{cases} \left(\left\lceil {Width}/{2} \right\rceil / \sqrt{3} r_s\right) \times P, & \text{if } (Width/2 < Height) \\ \left\{\left[Height - \left(\left\lceil {Width}/{2} \right\rceil - 1\right)\right] / \sqrt{3} r_s\right\} \times P, & \text{if } (Width/2 > Height) \end{cases} \quad (4)$$

Let $diff_{Any,Basic}$ denote the difference between the *Any-Direction* and *Basic* movement mechanisms in terms of maximal energy consumption. According to Expressions (2) and (4), the evaluation of $diff_{Any,Basic}$ can be derived as shown in Expression (5).

$$diff_{Any,\ Basic} = (2) - (4) = \begin{cases} \left\lceil \left(Height - \left\lceil {Width}/{2} \right\rceil\right) / \sqrt{3} r_s \right\rceil \times P, & \text{if } (Width/2 < Height) \\ \left\lceil \left(\left\lceil {Width}/{2} \right\rceil + 1\right) / \sqrt{3} r_s \right\rceil \times P, & \text{if } (Width/2 > Height) \end{cases} \quad (5)$$

The expression $diff_{Any,Basic} > 0$ indicates that the *Any-Direction* movement mechanism performs better in terms of energy-balancing. Similarly, let $diff_{Any,Forward}$ denote the difference between *Any-Direction* and *Forward-Only* movement mechanisms in terms of maximal energy consumption. Similar to (5), the $diff_{Any,Forward}$ can be derived according to Expressions (3) and (4), as shown in Expression (6). The expression $diff_{Any,Forward} > 0$ further indicates that the *Any-Direction* movement mechanism works better in terms of energy-balancing. As a result, *Any-Direction* movement mechanism outperforms the *Basic* and *Forward-Only* movement mechanisms in terms of energy-balancing.

$$diff_{Any,\ Forward} = (3) - (4) = \begin{cases} 0, & \text{if } (Width/2 < Height) \\ \left\lceil \left(\left\lceil {Width}/{2} \right\rceil + 1\right) / \sqrt{3} r_s \right\rceil \times P, & \text{if } (Width/2 > Height) \end{cases} \quad (6)$$

5. PERFORMANCE EVALUATION

This section compares the performances of the developed hole-movement strategies and a previous work [13] (*TCSC* in short). The WSN is initially randomly deployed with a large amount of sensors and then applied VFA algorithm [3] to heal the hole. The VFA algorithm is executed repeatedly to maintain spatial full-coverage until the whole sensing region reaches full coverage without any extra sensor. Therefore, at this moment, the lattice of all sensors would be hexagonal. Then we arbitrarily generate a big hole that contains several hole cells. The three proposed hole-movement mechanisms, namely *Basic*, *Forward-Only*, and *Any-Direction*, are then applied to execute the hole-movement task which aims to move the big hole along a specific direction d for a distance x. Performance comparisons of the three proposed hole-movement mechanisms and previous work *TCSC* are examined in terms of maximal power consumption, total power consumption, energy-balanced index and temporal full-coverage.

The simulation environment is described below. The network size is $1500\text{m} \times 1500\text{m}$. A big hole with various sizes may exist in any location of the network. The parameters used in our simulation are listed in Table II. The movement cost in energy consumption is set at 8.267 J/m (286.38 J/hop), which refers to the parameters in Reference [19]. Each simulation result is obtained from the average of 10 independent runs. To simplify the discussion of the performance results, the following arguments are given based on the assumption that the hole-movement task intends to move a big hole along a downward direction. The following depicts the results of our performance study.

5.1. The maximal power consumption

The maximal power consumption of movers is used to evaluate the degree of power balancing. Figure 20 depicts the maximal power consumption of movers by applying the three proposed hole-movement mechanisms and the *TCSC*. The hole-movement task intends to move the big hole for a distance of $x = 70$ hops $\times 20\sqrt{3}$ m/hop $= 1400\sqrt{3}$ m. The size of the big hole varies from 2×2 to 7×7. In general, the maximal power consumptions of the four mechanisms increase along with the size of the big hole. However, compared to the other three mechanisms, the maximal power consumption of the *Basic* movement mechanism is significant. By applying the *Basic* movement mechanism, the maximal power consumption totally depends on the height of the big hole. In the *Forward-Only* movement mechanism, detectors that are located at the bottom, left and right sides of the big hole participate in the hole-movement task and share the task workload. Therefore, the maximal power consumption of the *Forward-Only* movement mechanism is smaller than that of the *Basic* movement mechanism. Compared with the other three mechanisms,

Table II. Simulation parameters.

Parameter	Value
Communication range	40 m
Sensing range	20 m
Packet transmission cost	0.075 J/s
Packet reception cost	0.030 J/s
Idle cost	0.025 J/s
Maximum energy consumption in motes	324 J/h
Total initial energy	324000 J (1000 h)

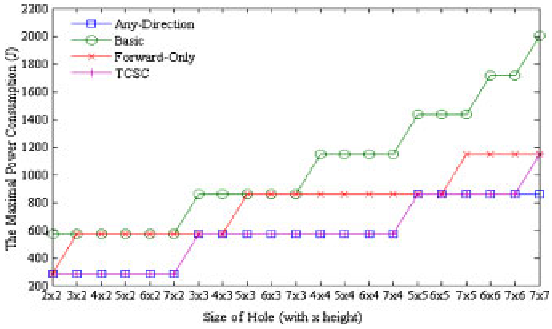


Figure 20. The maximal power consumption of movers with different big hole sizes by moving a fixed distance.

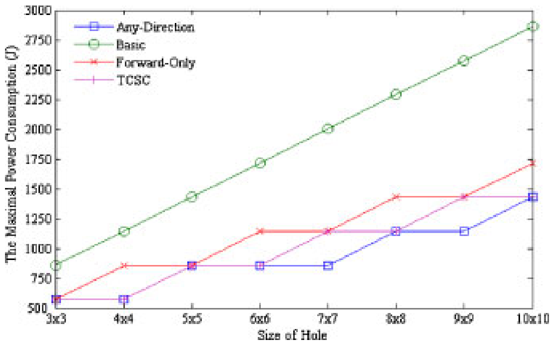


Figure 21. Performance evaluation of the three proposed movement mechanisms and the *TCSC* in terms of the maximal power consumption of movers.

the *Any-Direction* movement mechanism involves detectors, which are located at the top side of the hole, to participate in the hole-movement task. Consequently, the *Any-Direction* movement mechanism outperforms the other three mechanisms in terms of maximal power consumption.

Figure 21 compares the three proposed movement mechanisms and the *TCSC* in terms of the maximal power consumption. Here, the hole size varies from 3×3 to 10×10 to observe the change in the maximal power consumption between the four compared mechanisms. As shown in Figure 21, the *Any-Direction* movement mechanism outperforms the other three mechanisms since it has more movers that participate in the hole-movement task and thus reduces the maximal power consumption of the movers.

5.2. The total power consumption

Figure 22 investigates the total power consumption of the three proposed movement mechanisms. The hole-movement task intends to move the big hole for a distance of $x = 70$ hops $\times 20\sqrt{3}$ m/hop $= 1400\sqrt{3}$. In other words, each hole cell has to move for a distance of $1400\sqrt{3}$ to its destination. The size of the big hole varies from 2×2 to 7×7. In the *Basic* movement mechanism, each hole cell walks

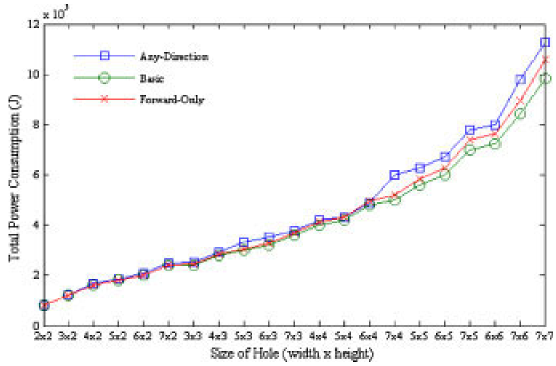


Figure 22. Performance evaluation of the three proposed movement mechanisms in terms of total power consumption of movers with a fixed moving distance on a big hole.

along the direction d directly. However, the *Forward-Only* and *Any-Direction* movement mechanisms aim to change the hole shape into a belt. They distribute the hole cells into different columns, which cause some hole cells to move along the directions other than direction d. Therefore, the total power consumption of the *Basic* movement mechanism is less than the other two movement mechanisms. In addition, Figure 22 also shows that the *Any-Direction* movement mechanism consumes the largest total energy since the mechanism allows some hole cells moving along direction $\overline{d}$, resulting in the increase of total power consumption.

5.3. The energy-balanced index

In the experiments, the energy-balanced index is measured by the average power consumption of all movers. Therefore a larger value in the energy-balanced index indicates that the energy consumption of all sensors is balanced. In Figure 23, we compare the three proposed mechanisms and the *TCSC* in terms of their energy-balanced index. Here, the hole size varies from 2×2 to 7×7. The hole-movement task intends to move the big hole for a distance of $x = 70$ hops $\times 20\sqrt{3}$ m/hop $= 1400\sqrt{3}$ m. Figure 23 shows that the

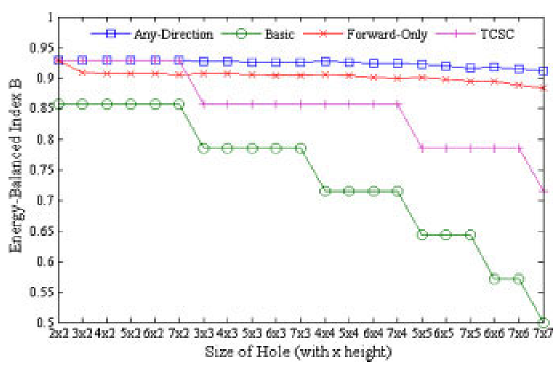


Figure 23. Performance evaluation of the three proposed movement mechanisms and the *TCSC* in terms of their energy-balanced index with a fixed moving distance on a big hole.

Wirel. Commun. Mob. Comput. 2012; **12**:445–462 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

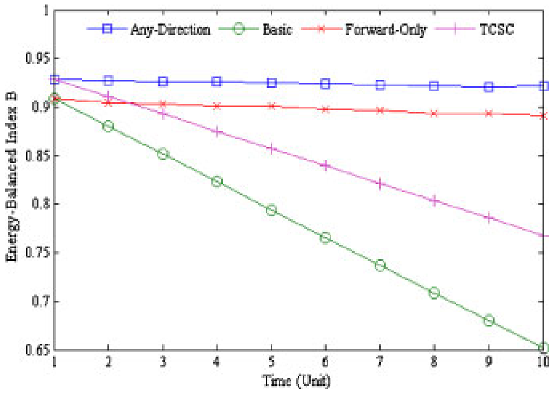


Figure 24. Performance evaluation of four different movement mechanisms in terms of their energy-balanced index.

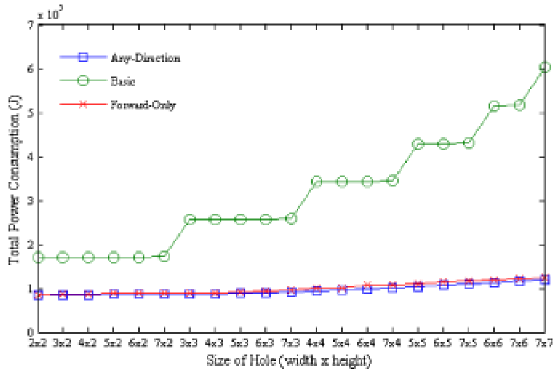


Figure 25. Performance evaluation of the three proposed movement mechanisms in terms of total power consumption with a fixed hole sweep area.

energy-balanced indices of the *Basic* and *TCSC* movement mechanisms decrease with the size of the big hole because the power consumption of each mover highly depends on the height of the hole. However, *TCSC* has a stronger height constraint than *Basic* movement mechanism in terms of power consumption of the mobile sensor. Therefore, the energy-balance index of the *TCSC* is better than that of the *Basic* movement mechanism. As shown in Figure 23, the energy-balanced indices of the *Forward-Only* and *Any-Direction* movement mechanisms are close to the optimal value of the energy-balanced index since they enable more movers to share the workload of the hole-movement task and therefore result in a good energy-balanced index.

Figure 24 compares the energy-balanced index of the three hole-movement mechanisms and the *TCSC*. Here, the hole size is set to 7×7, and the hole-movement task intends to move the big hole for a distance of $x = 70\,\text{hops} \times 20\sqrt{3}\,\text{m/hop} = 1400\sqrt{3}\,\text{m}$. As shown in Figure 24, the energy-balanced index of the *Basic* and *TCSC* movement mechanisms decrease along with the time since the power consumption of each mover highly depends on the height of the hole. However, the energy consumption of movers in *TCSC* does not exceed the constraint of [*Height/2*]. Therefore, the energy-balance index of the *TCSC* is better than that of the *Basic* movement mechanism. On the other hand, the *Forward-Only* and *Any-Direction* movement mechanisms have good energy-balanced index since more sensors participate in the hole-movement task, and thus the power consumptions of all the sensors in the WSN are more energy-balanced.

5.4. Temporal full-coverage

To achieve temporal full-coverage over a period of time, the mover should walk over the network to replace holes to timely increase the coverage degree. In addition, the total power consumption is another index for estimating the network lifetime. Therefore, this experiment intends to investigate the total power consumption for executing a task that sweeps a hole across a fixed area A. The three proposed mechanisms are applied to execute the abovementioned task. After, their total power consumptions are compared under the environments of a big hole with different sizes and shapes. Here, the hole size varies from 2×2 to 7×7. The fixed coverage area A is set at $1040\,\text{m}^2$.

As shown in Figure 25, by applying the *Basic* movement mechanism, the total power consumption of the movers increases significantly with the hole size. In fact, the trajectory of each mover in the *Basic* movement mechanism is similar since sensors that are located on the same column have the same movement trajectory. Therefore, the more area that they have swept will be overlapped which results in the *Basic* movement mechanism walking for more hops to reach the fixed coverage area A. The *Forward-Only* and *Any-Direction* movement mechanisms outperform the *Basic* movement mechanism in terms of total power consumption required for sweeping a fixed area size. This is because the two mechanisms distribute the big hole into different areas, and the size of the overlap region swept by their movers is smaller compared to the movers that apply the *Basic* movement mechanism. The performance of the *Any-Direction* movement mechanism is better than the *Forward-Only* movement mechanism. However, since the measurement base is 10^5, these two lines are close to each other.

5.5. The total power consumption of a hybrid mechanism

Given a constraint in the maximal power consumption of each mover, we intend to find out the best movement that minimizes the total power consumption. Recall that the *Basic* movement mechanism minimizes total power consumption but results in a situation wherein each mover consumes significant power and thus imbalance power. Therefore, given the constraint in the maximal power consumption for each mover, the *Basic* movement mechanism might be a failure since each mover that applies the *Basic* movement mechanism might consume more energy than

DOI: 10.1002/wcm

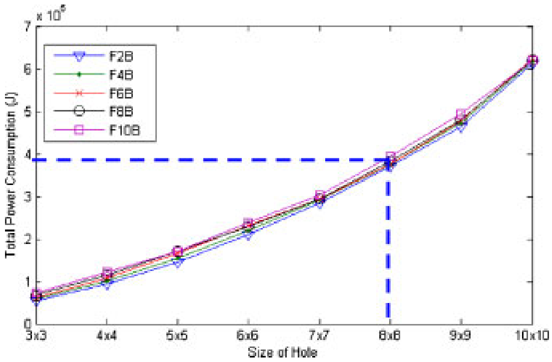


Figure 26. Performance evaluation of the five different hybrid movement strategies where the *Forward-Only* movement mechanism is applied first. The evaluation compares the total power consumption of the five strategies.

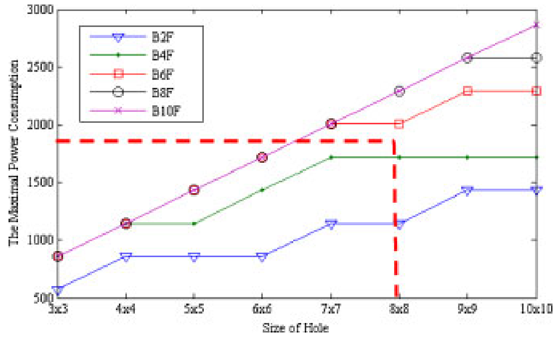


Figure 28. Performance evaluation of the five different hybrid movement strategies that apply the *Basic* movement mechanism first. The measurement compares the maximal power consumption of all movers for the five strategies.

the given constraint to complete the movement task. The *Forward-Only* movement mechanism takes both total power consumption and power balancing issues into consideration. Therefore, applying the *Forward-Only* movement mechanism can distribute some workload of the hole-movement task to the other movers such that each mover's power consumption can be reduced while the total power consumption is not significantly increased. Therefore, the combining *Basic* and *Forward-Only* movement mechanisms can meet the requirement where each mover's power consumption is not larger than the given constraint, and in which the total power consumption for hole-movement task can be minimized.

This subsection presents the experimental results that combine the *Basic* movement mechanism and *Forward-Only* movement mechanisms, as shown in Figures 26 and 27. Here the hole-movement task intends to move the big hole for a distance of $x = 70$ hops $\times 20\sqrt{3}$ m/hop $= 1400\sqrt{3}$ m with the hole size varying from 3×3 to 10×10. In Figure 26, the hybrid mechanism applies the *Forward-Only* movement mechanism first and, after, the *Basic* movement mechanism. The hybrid hole-movement mechanisms in Figure 26 are *F2B, F4B, F6B, F8B,* and *F10B* where *F2B* shows that the hybrid mechanism first applies the *Forward-Only* movement mechanism for two hops, and then applies the *Basic* movement mechanism for 38 hops as its strategy. The other hybrid strategies in Figure 26 may be deduced by analogy. Figure 27 shows the performance results of a hybrid mechanism which applies the *Basic* movement mechanism first, and, after, the *Forward-Only* movement mechanism, as its strategy. These hybrid strategies in Figure 27 are *B2F, B4F, B6F, B8F and B10F*. Similar to the hybrid mechanism in Figure 26, the *B2F* indicates that the hybrid mechanism first applies the *Basic* movement mechanism for two hops and then applies the *Forward-Only* movement mechanism for 38 hops as its strategy.

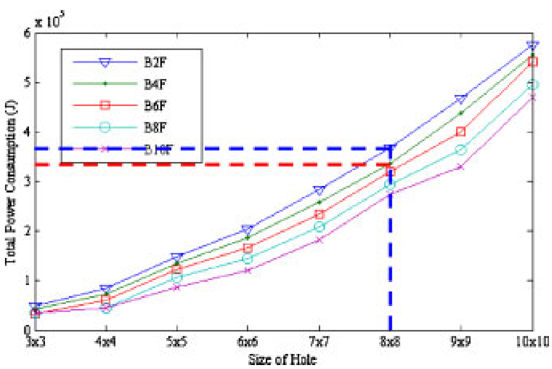


Figure 27. Performance evaluation of the five different hybrid movement strategies where the *Basic* movement mechanism is applied first. The evaluation compares the total power consumption of the five strategies.

In these two experiments, it can be observed that the total power consumption for the hybrid mechanism that applies the *Basic* movement mechanism first is better than the one that applies the *Forward-Only* movement mechanism first. In Figure 28, we show the maximal power consumption of movers with different hybrid strategies which apply the *Basic* movement mechanism first. Knowing the maximal power consumption of movers and the total power consumption, we obtain an optimal movement that minimizes the total power consumption for a given constraint in the maximal power consumption of each mover. Take a simple example. We restrict the maximal power consumption of each mover by 2000 J at most, and we intend to find the best hybrid strategy that minimizes the total power consumption for moving a big hole, sized 8×8. As shown in Figure 28, the two strategies, *B2F* and *B4F*, satisfy this constraint. Next, we check the two mechanisms in Figure 27. The *B4F* hybrid strategy has smaller total power consumption compared to the *B2F* strategy. Therefore, it is suggested that the hole-movement task should apply the *Basic* movement mechanism for four hops first and then apply the *Forward-Only* movement mechanism for 36 hops to obtain optimal total power consumption under the given constraint.

 Wirel. Commun. Mob. Comput. 2012; **12**:445–462 © 2010 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

6. CONCLUSIONS AND FUTURE WORK

Given a mobile WSN that contains a hole with a number of mobile sensors that are insufficient to cover it, the data collected from the sensors would be biased since only the sensing information of a certain region can be collected. The proposed hole-movement mechanisms can prevent data collection from being biased and achieve the purpose of temporal full-coverage. This paper observes that moving a mobile sensor in a direction is identical of moving a hole in the opposite direction. We developed three distributed hole-movement strategies that move the existing big hole in a way that either the total energy consumption is minimized or the power consumption of the sensors is balanced. First, the *Basic* movement strategy is proposed to move a hole with minimal total power consumption. Second, the *Forward-Only* movement strategy moves a hole by taking both energy-balancing and total power consumption into consideration. Finally, the *Any-Direction* movement strategy is proposed by fully considering the energy-balancing as its policy. Based on the proposed strategies, whenever a mobile sensor detects a coverage hole within its sensing range, it can locally determine whether or not it should migrate to the hole cell to achieve temporal full-coverage. Simulation results reveal that the proposed hole-movement mechanisms enhance the coverage of WSN temporally and balance the power consumption of mobile sensor nodes. Future work will investigate the integration of the proposed mechanisms, develop a hybrid mechanism which takes full advantages of the three mechanisms and extend their usages to randomly deployed WSNs.

ACKNOWLEDGEMENTS

The author would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 97-2221-E-032-049-MY3.

REFERENCES

1. Slijepcevic S, Potkonjak M. Power efficient organization of wireless sensor networks. *IEEE Pervasive Computing and Communications Workshop (PerCom)*, March 2005; 406–410.
2. Wang G, Cao G, Berman P, La Porta T. Bidding protocols for deploying mobile sensors. *IEEE Transactions on Mobile Computing* 2007; **6**(5): 515–528.
3. Zou Y, Chakrabarty K. Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)* 2004; **3**: 61–91.
4. Wang G, Cao G, La Porta T. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing* 2006; **5**(6): 640–652.
5. Shan X, Tan J. Mobile sensor deployment for a dynamic cluster-based target tracking sensor network. *IEEE International Conference on Intelligent Robots and Systems*, August 2005; 1452–1457 .
6. Zhang M, Du X, Nygard K. Improving coverage performance in sensor networks by using mobile sensors. *IEEE MILCOM*, October 2005; 17–20.
7. Gui C, Mohapatra P. Virtual patrol: a new power conservation design for surveillance using sensor networks. *Information Processing in Sensor Networks (IPSN)*, April 2005; 246–253.
8. Liu B, Brass P, Dousse O. Mobility improves coverage of sensor networks. *International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, 2005; 300–308.
9. Meghanathan N. Use of tree traversal algorithms for chain formation in the PEGASIS data gathering protocol for wireless sensor networks. *KSII Transactions on Internet and Information Systems* 2009; **3**(6): 612–627.
10. Lee D, Kim J. Analysis of optimized aggregation timing in wireless sensor networks. *KSII Transactions on Internet and Information Systems* 2009; **3**(2): 209–218.
11. Yeo M, Seo D, Yoo J. Data correlation-based clustering algorithm in wireless sensor networks. *KSII Transactions on Internet and Information Systems* 2009; **3**(2): 331–343.
12. Wang G, Cao G, Porta TL, Zhang W. Sensor relocation in mobile sensor networks. *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Vol. 4, 2005; 2302–2312.
13. Jiang Z, Wu J, Agah A, Lu B. Topology control for secured coverage in wireless sensor networks. *Proceedings of the 3rd IEEE International Workshop on Wireless and Sensor Networks Security (IEEE WSNS)*, October 2007.
14. Hu L, Evans D. Localization for mobile sensor networks. *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, October 2004; 45–57.
15. Rudafshani M, Datta S. Localization in wireless sensor networks. *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*, April 2007; 51–60.
16. Bai X, Kumar S, Yun Z, Xuan D, Lai TH. Deploying wireless sensors to achieve both coverage and connectivity. *Proceedings of the 7th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2006.
17. Chen WP, Hou JC, Sha L. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing* 2004; **3**: 258–271.
18. Chang CY, Chang CT, Tu SC. Obstacle-free geocasting routing protocol for ad-hoc wireless networks.

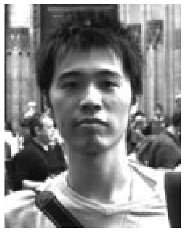
ACM/Baltzer Journal of Wireless Networks 2003; **9**(2): 143–155.

19. Ganeriwal S, Kansal A, Srivastava MB. Self aware actuation for fault repair in sensor networks. *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

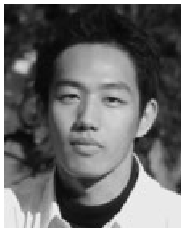
AUTHORS' BIOGRAPHIES

Chih-Yung Chang received the Ph.D. degree in Computer Science and Information Engineering from National Central University, Taiwan, in 1995. He joined the faculty of the Department of Computer and Information Science at Aletheia University, Taiwan, as an Assistant Professor in 1997. He was the Chair of the Department of Computer and Information Science, Aletheia University, from August 2000 to July 2002. He is currently a Full Professor of Department of Computer Science and Information Engineering at Tamkang University, Taiwan. Dr Chang served as an Associate Guest Editor of Journal of Internet Technology (JIT, 2004 and 2008), Journal of Mobile Multimedia (JMM, 2005), International Journal of Wireless and Mobile Computing, Journal of Information Science and Engineering (JISE, 2009) and a member of Editorial Board of International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC, 2009), Journal of Information Technology and Applications (JITA, 2008–2009), Tamsui Oxford Journal of Mathematical Sciences (2001–2009). He was an Area Chair of IEEE AINA 2005, Vice Chair of IEEE WisCom 2005, EUC 2005, and IEEE AINA 2008, Track Chair (Learning Technology in Education Track) of IEEE ITRE 2005, Program Co-Chair of IEEE MNSAT 2005, UbiLearn 2006, WASN 2007, ACM SAMnet 2008, IEEE AHUC 2008, Workshop Co-Chair of IEEE INA 2005, MSEAT 2003, MSEAT 2004, ICS 2008, and Publication Chair of MSEAT 2005, SCORM 2006, and FGCN 2007. Dr. Chang is a member of the IEEE Computer Society, Communication Society and IEICE society. His current research interests include vehicular ad hoc networks, wireless sensor networks, WiMax broadband networks, Bluetooth radio networks, Ad Hoc wireless networks, and mobile computing.

Li-Ling Hung received her B.S. degree in computer science from Tunghai University, Taichung, Taiwan in 1993. She received her MS degree in computer science and engineering from Yuan-Ze University, Chungli, Taiwan in 1995. She received her PhD degree in computer science and information engineering at National Taiwan University of Science and Technology, Taipei in 2008. Since February 2009 she has been an assistant professor in Department of Computer Science and Information Engineering, Aletheia University, Tamsui, Taiwan. Her research interests include wireless sensor networks, vehicular Ad Hoc networks, WiMax, and parallel computing.



Yun-Jung Lu received the B.S. degree in Computer Science from Alethiea University, Taiwan, in 2006, and the M.S. degree in 2008 from Computer Science and Information engineering of Tamkang University, Taiwan, where he is working toward his Ph.D degree. His research interests include WiFi and mobile computing, ad hoc and sensor network and networking planning.



Sheng-Wen Chang received the B.S. degree in Computer Science and Information Engineering from Tamkang University, Taiwan, in 2004, the Ph.D. degree in 2009. Mr. Chang was a student member of the IEEE Computer Society, Communication Society and IEICE society. His research interests include LTE, WiMAX, Bluetooth radio networks, wireless sensor networks and ad hoc wireless networks. He currently is performing his compulsory military service for his country.

Yu-Chieh Chen received the B.S. degree in Computer Science and Information Engineering from Ming Chuan University, Taiwan, in 2005, and the M.S. degree in Computer Science and Information Engineering from Tamkang University, Taiwan, in 2007. Since 2007, he is a Ph.D. degree candidate in Department of Computer Science and Information Engineering at Tamkang University. Mr. Chen won lots of scholarships in Taiwan and participated in many Wireless Sensor Networking projects. His research domains include wireless sensor networks, Ad-Hoc wireless networks, mobile/wireless computing, and WiMAX.