Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2007, Article ID 64574, 15 pages
doi:10.1155/2007/64574

Research Article

HUMS: An Autonomous Moving Strategy for Mobile Sinks in Data-Gathering Sensor Networks

Yanzhong Bi,[1,2] Limin Sun,[1] Jian Ma,[3] Na Li,[4] Imran Ali Khan,[4] and Canfeng Chen[3]

[1] *Institute of Software, Chinese Academy of Sciences, Beijing 100080, China*
[2] *Graduate School of Chinese Academy of Sciences, Beijing 100039, China*
[3] *Nokia Research Center, Beijing 100013, China*
[4] *Computer Network Information Center, Chinese Academy of Sciences, Beijing 100080, China*

Received 30 September 2006; Accepted 13 March 2007

Recommended by Lionel M. Ni

Sink mobility has attracted much research interest in recent years because it can improve network performance such as energy efficiency and throughput. An energy-unconscious moving strategy is potentially harmful to the balance of the energy consumption among sensor nodes so as to aggravate the hotspot problem of sensor networks. In this paper, we propose an autonomous moving strategy for the mobile sinks in data-gathering applications. In our solution, a mobile sink approaches the nodes with high residual energy to force them to forward data for other nodes and tries to avoid passing by the nodes with low energy. We performed simulation experiments to compare our solution with other three data-gathering schemes. The simulation results show that our strategy cannot only extend network lifetime notably but also provides scalability and topology adaptability.

1. INTRODUCTION

Wireless sensor networks composed of networked sensors and mobile sinks have the potentiality of providing diverse services to numerous applications, such as surveillance systems and control systems for commercial, industrial, and military scenarios. In those systems, a large amount of inexpensive sensors is deployed in monitoring fields to sense the physical environments, and a few mobile sinks are involved in collecting sensed data, making decisions, and taking actions. Since sensor nodes are expected to be deployed in harsh environments, which cause great difficulty to recharge or change their battery, the lifetime of a wireless sensor network is limited to the battery lifetime of the sensors [1–3].

Many energy-efficient protocols and schemes have been proposed for data-gathering sensor networks in recent years [4–7]. However, if the device involved in collecting data is static, the sensors that are close to the device would become hotspots and die earlier than other sensors because they have to transmit huge amounts of data for other sensors. Many researchers have demonstrated that the mobility of network elements can improve network performance, that is, network throughput, reliability, and energy efficiency [8–22]; therefore, wireless sensor networks with mobile sinks have many advantages over the static sensor networks for data-gathering applications. In particular, employing a mobile device to collect data can reduce the effects of the hotspot problem, balance energy consumption among sensor nodes, and thereby prolong the network lifetime to a great extent [23–25]. However, many moving strategies are not suitable for the mobile sinks in data-gathering networks. For example, a random moving sink [8–10] is unconscious of energy and potentially threatens the energy balance among sensor nodes. In addition, a mobile sink that moves along some tracks or cableways [13–18, 23–25] lacks flexibility and scalability because its moving path always has to be redesigned when the sink is transplanted to other networks. In contrast, autonomous moving strategies, in which a sink makes moving decisions according to the run-time circumstances, can provide reasonable adaptability to various types of network conditions.

We focus on a type of wireless network that consists of many sensors and a mobile sink, which is called *energy mower* and is in charge of collecting sensed data periodically. In this paper, we propose an autonomous moving strategy, in which the energy mower can make moving decisions without the global topology of the network or energy status of all sensor nodes. The aim of this research is to design a strategy for the energy mower to react to the energy distribution of the sensors. If the sensors report their data by multihop, the closer to the energy mower the sensors are, the heavier their traffic burdens are, and the more energy they have to consume. Thus, we drive the energy mower to approach the sensor with the highest residual energy in the network and avoid passing by the sensors with low residual energy. In each data-gathering period, the sensors pack their residual energy information into data packets, so that the energy mower can calculate a new position to move after it collects all the packets. During the sojourn of the energy mower in each position, the sensors report their data packets by multihop. Furthermore, considering the limited speed of moving the energy mower in a real scenario, it is not possible for the energy mower to reach anywhere in the network field by one move. As a whole, the proposed strategy makes the energy mower move autonomously to collect data packets in the monitoring area, along with balancing the energy consumption among all the sensors, alleviating the hotspot problem, and extending the network lifetime.

The remainder of the paper is organized as follows: in Section 2, we summarize the related work on utilizing mobility to improve network performance. In Section 3, we describe our data-gathering scheme. In Section 4, we present our moving strategy in detail and provide simulation results in Section 5. In Section 6, we discuss some design details of the moving strategy. Finally, we conclude this paper in Section 7.

2. RELATED WORK

Since we focus on moving strategies of the mobile devices in data-gathering sensor networks, we mainly review some typical related studies in this section.

Wireless sensor networks with mobile devices have drawn more and more attention recently. This type of network can provide flexible services in practical applications, such as in a farming system [26]. The special network is faced with several challenging problems unlike those of the traditional static wireless sensor networks, in particular, on the issues of how to move to the destination and where the mobile devices should be located during the moving procedure. In [27], the authors proposed a practical algorithm based on centroidal voronoi tessellation (CVT) to solve the problem of actuator motion planning to neutralize the pollution. Their moving strategy guarantees that the neutralizing chemicals should be released in such a manner that the diffusion of the pollution is constrained so that the heavily affected area is kept as small as possible. In [28], the authors first set arbitrary initial values of diffusion system parameters, which made a contribution to the optimal trajectories of sensors, and then sensed data were collected during the course of sensor moving. In turn, after analyzing the data collected, the network updates the trajectories of sensors, which are more useful to neutralize the pollution in that scenario. Similarly, in [29], the authors commanded mobile sensors to collect samples of the distribution of interest and then used the samples to predict the distribution of new samples, which have an influential effect on the moving strategy. These studies paid more attention to the original data of the sensors than their energy consumption, which is a key factor in the periodical data-gathering sensor networks.

Much work has been conducted on the data-gathering sensor networks where mobile devices move along fixed paths. The authors of [16, 17] set up a network system, in which the path traversed by their mobile router is fixed, and they proposed a self-adaptive protocol based on wireless communication quality to control the mobility of the mobile router. Their mobile router can adjust its speeds dynamically in response to the run-time environment of the network. In [23, 24], a path planning for a mobile device was formulated as the mobile element scheduling (MES) problem based on the assumption that a mobile element visits each sensor node to collect data. Although the strategies in which a mobile device visits each sensor node or awakes one-hop neighbor nodes to collect sensed data can save the most energy, due to the limited moving speed of an actual mobile device, sensed data will suffer from enormous latency when the network size scales up. The authors of [25] have theoretically proved that, under the conditions of a short path routing and a round network region, moving along network periphery is the optimum strategy for a mobile sink. Their analysis was based on an ideal load-balanced short path routing protocol and the simulations were performed without consideration of MAC effects. In addition, linear programming methods were adopted to determine the optimal positions of the sinks in [14, 15]; deployment problems for static sinks were considered in [30, 31]. However, fixed-track moving strategies lack adaptability to different networks and have to be redesigned when the network devices are deployed in various circumstances.

Recently, several researchers have investigated the autonomous moving strategies for mobile sinks. In [32], the authors pointed out that selecting the optimal moving positions for mobile sinks was an NP-hard problem and proposed a heuristic algorithm to determine the moving directions and distances. In the algorithm proposed in [32], a sink moves towards the nodes that generate the most number of data packets, but it moves only when it detects an unacceptable performance. Therefore, the algorithm is more suitable for event-driven applications, such as detecting targets, rather than data-gathering applications where all nodes report sensed data periodically; otherwise the sink will hover in a small area when it stands at the center of the network because the data amount in each direction is nearly equal. The authors of [33] proposed two strategies to move the sink adaptively to react to dynamic events that followed a correlated random walk mobility model, impracticable to the mobile devices that gather data periodically from all sensor nodes.

3. DATA-GATHERING SCHEME EMPLOYED

We assume that a wireless sensor network, which serves data-gathering applications, consists of a high-powered mobile sink and a large number of battery-powered static sensors. Both the sink and the sensors know their locations by either

GPS services or self-configured localization techniques. Each sensor node sends a fixed-length data packet to the sink in each data-gathering period.

In our data-gathering scheme, before gathering sensed data, the network will carry out a neighbor discovery process first. The discovery process is used to help sensor nodes to set up their neighbor lists. Each sensor node will broadcast several HELLO messages to notify its one-hop neighbors of its own ID and position. The HELLO messages will be sent with different random delays to reduce local collisions. After sending each HELLO message, a sensor node will listen and receive messages from its neighbor nodes. During the neighbor discovery process, the sink does not move, receive, or send messages.

After the execution of the neighbor discovery process, the network starts gathering sensed data periodically. In each data-gathering period, the sensor nodes will send their data to the sink through multihop communication paths. Considering that the sensor nodes near the sink are inclined to become hotspots with the multihop routing protocols, we suggest that the sink should move proactively to shift the hotspot area to different places of the network. We can take advantage of the proactive movement of the sink to balance the energy consumption among the sensor nodes and extend the network lifetime. Our data-gathering scheme aims to provide a feasible framework for this type of sensor network.

In this scheme, each data-gathering period consists of three phases. In the first phase, the sink broadcasts a notification message to inform the sensor nodes of its position. Because of the speed constraint of the sink, it is unnecessary to inform all sensor nodes of its each movement. If the sink does not move far, only the sensor nodes in its vicinity have to be informed of the movement, and the nodes far from it do not have to change their directions of sending data. The sink can control the spreading range of the notification message by adjusting the value of the *Time-to-Live* field in the message. In addition, if the network is not very large, state-of-the-art communication techniques can provide the sink with the capability of sending the notification message with a large communication radius to inform the whole network. In this case, all the sensor nodes only need to receive one message to obtain the new position of the sink.

In the second phase of the data-gathering period, the sensor nodes report their data to the sink in a multihop manner. As the sensor nodes know the positions of the sink and their one-hop neighbors, they can determine their next-hop nodes using a location-based routing algorithm. During this phase, the sink stays on and gathers data from all the nodes in the network, which is beneficial to routing, thus many existing energy-efficient protocols designed for static networks can be applicable.

In the third phase, in response to the residual energy status of the network, the sink determines and arrives at the new position before the next data-gathering period begins. Since the sensor nodes do not need to receive or send data in this phase, they can switch into sleep mode to preserve energy.

In summary, the scheme divides a data-gathering period into three separate phases according to different operations of the sink, which involve movement, position notification, and data collection. Therefore, the scheme can be used with diverse moving strategies for sinks and routing protocols for sensors, which makes the whole system scalable and flexible.

4. AUTONOMOUS MOVING STRATEGY

In this section, we present a *half-quadrant-based moving strategy* (HUMS), which incorporates with our data-gathering scheme, for the mobile sink. Unlike the strategy proposed in [32, 33], our strategy makes a sink move proactively towards the node that has the most residual energy to balance energy consumption among all sensor nodes in the network. It seems that the sink regards the residual energy of the sensor nodes as an uneven grassy lawn and tries to make it smooth by cutting the tallest grass. Therefore, we call the sink that employs our moving strategy as an *energy mower*.

4.1. Preparation for making moving decisions

To make moving decisions with HUMS, the energy mower requires each data packet reported by the sensors contain two groups of information besides sensed data: one consists of the residual energy and the location of the sensor node that has the highest residual energy among the nodes experienced by the packet, and the other is composed of the residual energy and the location of the node that has the lowest residual energy. Sensor nodes on the delivery path of the packet can update the information of either of the two groups according to the comparison results between their own residual energy and that recorded in the packet. If their residual energy is higher than the record of the highest energy in the first group, they will replace the location and energy information in the first group with their own. Similarly, they will replace the information in the second group if their residual energy is lower than the record of the lowest energy.

Since the sensed data of the whole network will arrive at the energy mower along different paths, the energy mower will know the locations of some sensor nodes with comparative high or low residual energy in the network after it receives all the data packets in each data-gathering period. The energy mower selects the node with the highest residual energy and regards its location as the moving destination (called *MoveDest* for short) of the current data-gathering period. If there is more than one node with the same highest residual energy, the energy mower will choose the nearest one to be MoveDest. All the nodes that are reported as having the lowest residual energy form a set of *quasi-hotspots*, which are in danger of exhausting their energy. The size of the quasi-hotspots set is usually no more than the number of the one-hop neighbors of the energy mower because many delivery paths will overlap each other and converge near the energy mower. In each data-gathering period, the energy mower will reselect MoveDest and the set of quasi-hotspots to make a new moving decision according to their energy distributions. However, along with MoveDest's rotating from one sensor to another frequently, the energy mower has to alter its moving direction towards different sensors

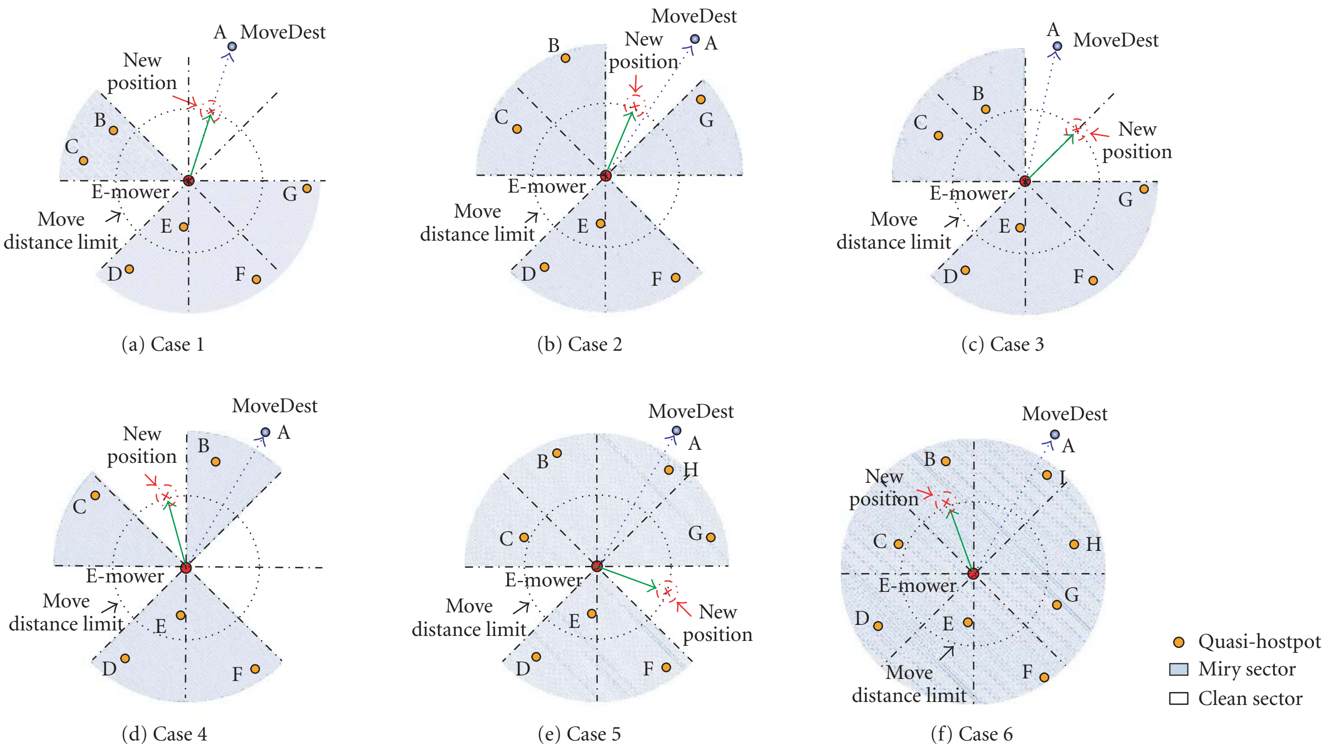

Figure 1: Six decision cases of the half-quadrant-based moving strategy. In each case, the blue node A is MoveDest, and the orange nodes are quasi-hotspots.

continually, like a ping-pong effect. In such case, due to the speed constraint of the energy mower, it may traverse in a small area without reaching any destinations. Furthermore, since the energy mower gathers the sensed data after each movement, the ping-pong effect may consume excessive energy of the sensors around the mower. To handle this problem, we grade the energy of a sensor node with *energy levels*, which may include, for example, 100 levels, and mark a full energy with the highest level. We restrict that the energy mower can select a different node as a new MoveDest only when the residual energy of the node exceeds that of the current MoveDest by at least one energy level. This mechanism provides the energy mower more chances to keep a stable moving direction for a few data-gathering periods and gradually get close to MoveDest.

In data-gathering applications, the sensor nodes near the energy mower have to consume more energy to forward data than the nodes far from the energy mower in multihop routing protocols. Therefore, the energy mower should always try to approach MoveDest to force it to expend much energy on forwarding data for other nodes. On the other hand, on getting close to MoveDest, the energy mower has to avoid passing by the quasi-hotspots, which is beneficial to reduce the energy consumption of the low-energy nodes. Considering that an actual mobile device can only move at a limited speed, we restrict the distance spanned by the energy mower in a data-gathering period to a constant distance depending on the actual mobile device. In other words, it seems like that the energy mower *jumps* towards MoveDest step by step and it jumps only one hop in each data-gathering period. For simplicity, in the following description of the proposed algorithm, we assume the distance to be the same length as the communication distance of a sensor. Further discussion for the selection of the move distance limit is given in Section 5.2.

In HUMS, to make a moving decision, the energy mower sets up a coordinate system that takes its current position as the origin and divides the coordinate system into eight half-quadrants, as shown in Figure 1. Assuming the energy mower knows the location of the network periphery, it can mark the half-quadrants out of the network region as *invalid* ones. Among the other *valid* half-quadrants, the energy mower regards the half-quadrants that do not cover any quasi-hotspots as *clean sectors* and regards those that cover at least one quasi-hotspot as *miry sectors*. In addition, the energy mower assigns an *energy token* to each valid half-quadrant. If there are some quasi-hotspots in a half-quadrant, the energy token of the half-quadrant is set to the average energy of the quasi-hotspots in it. On the other hand, if a half-quadrant does not cover any quasi-hotspots, its energy token is set to a high value, for example, the maximum initial energy of a sensor node. Since the energy mower knows the locations of MoveDest and the quasi-hotspots, it marks the half-quadrant where MoveDest is located as *DestSector*, and both the left and right half-quadrants of DestSector as *forward sectors*. In each data-gathering period, the energy mower is inclined to approach MoveDest through clean sectors; moreover, due to the expectation of approaching

MoveDest as soon as possible, the energy mower prefers to move through DestSector and the forward sectors.

The process of the energy mower approaching MoveDest involves two cases. In one case, when the energy mower is far away from MoveDest, it has to move towards MoveDest. If another sensor node becomes a new MoveDest before the energy mower arrives at the old one, the energy mower will adjust its moving direction and start to approach the new MoveDest. In the other case, when the energy mower is close to MoveDest, it tries to determine a sojourn position around it to consume the energy of MoveDest as much as possible in a short time. Considering that consuming the energy of MoveDest inefficiently can threaten the sensor nodes around MoveDest, which contain little residual energy, we suggest a simple mechanism to help the energy mower find a proper position to sojourn. We describe the mechanisms proposed for the two cases in the following two subsections, respectively.

4.2. Case 1: far from MoveDest

In each data-gathering period of this stage, the energy mower selects a sector to move into by using a half-quadrant-based algorithm and determines a certain sojourn position in that sector by using an algorithm called *minimum-influence position selection* (MIPS) algorithm if needed.

4.2.1. Half-quadrant-based algorithm

The half-quadrant-based algorithm is aimed at selecting one out of the eight half-quadrants to be the destination sector for the energy mower in each data-gathering period. The basic principle of the algorithm is trying to avoid leading the energy mower close to quasi-hotspots while moving towards MoveDest. The scenarios that the algorithm may involve can be classified into six cases according to the distribution of MoveDest and the quasi-hotspots over the eight half-quadrants, which are described as follows.

Case 1. As shown in Figure 1(a), if DestSector and both forward sectors do not cover any quasi-hotspots, that is, they are clean, the energy mower will move in the direction of MoveDest. Since the energy mower has limited moving ability during one data-reporting period, which is illustrated by the dotted circle in Figure 1(a), it will move to the intersection of the line towards MoveDest and the dotted circle. In this way, the energy mower approaches MoveDest directly, without fear of drastically exhausting the energy of the quasi-hotspots.

Case 2. If DestSector is clean, but both forward sectors are miry, the energy mower will move into DestSector, as shown in Figure 1(b). Because the energy mower wants to keep far from the quasi-hotspots in both forward sectors, it calculates the precise sojourn position to arrive at by using the MIPS algorithm.

Case 3. As Figure 1(c) shows, if DestSector and a forward sector are clean, the energy mower will move to the intersection of the dotted circle and the boundary between DestSector and the clean forward sector. This position is beneficial to both requirements of approaching MoveDest as soon as possible and keeping away from quasi-hotspots as far as possible.

Case 4. As shown in Figure 1(d), if DestSector is miry, and meanwhile, at least one of the two forward sectors is clean, the energy mower will move into a clean forward sector rather than DestSector. When only one forward sector is clean, the energy mower will move into it. On the other hand, when both forward sectors are clean, the energy mower will calculate the sum of the energy tokens of the left and right sectors of each forward sector, respectively, and choose the forward sector with a higher sum to move into. Similarly, the energy mower will calculate the precise sojourn position by using the MIPS algorithm.

Case 5. If DestSector and forward sectors are all miry, and at least one of the other sectors is clean, the energy mower will give up moving towards MoveDest temporarily and will move along a roundabout route. It will determine the sojourn position in the similar way as that in Case 4.

Case 6. As Figure 1(f) shows, if all the eight sectors are miry, the energy mower will select the sector with the highest energy token to move into and calculate the precise sojourn position with MIPS.

4.2.2. MIPS: minimum-influence position selection algorithm

Every quasi-hotspot hopes to stay away from the energy mower to reduce the energy consumption of forwarding data. The main idea behind the MIPS algorithm is that it is necessary for the energy mower to take account of the position distribution of some near quasi-hotspots when determining a sojourn position in the sector selected by the half-quadrant-based algorithm. The energy mower uniformly selects several points (e.g., four) on the dotted arc spanning the selected sector, which is a section of the circle of the move distance radius, as shown in Figure 2, and regards these points as candidates for the sojourn position. In MIPS, we define a type of *influence force* between a quasi-hotspot and a candidate for the sojourn position according to the residual energy and the position of the quasi-hotspot. The energy mower calculates the *composite influence force* from all the quasi-hotspots on each position candidate and chooses the candidate that has the minimum composite force as the sojourn position of the current data-gathering period.

Assuming the traffic burden of a sensor node is proportional to the square of the distance from the node to the edge of the network when a short-path-like routing protocol is employed [25], we define the strength of the influence force between a quasi-hotspot q to a position candidate c as

$$\left|\vec{f_c^q}\right| = k \cdot \frac{D_q^2}{E_q}, \quad (1)$$

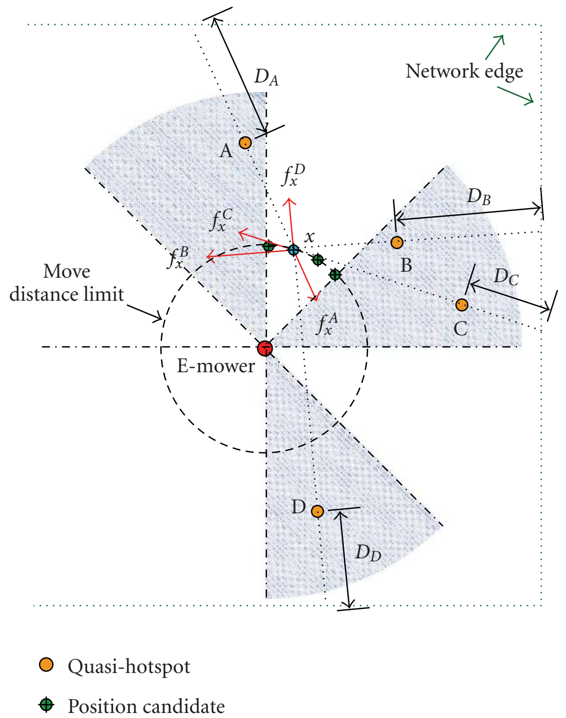

FIGURE 2: Influence forces acting on a position candidate x from all the four quasi-hotspots (nodes A, B, C, and D) in the network.

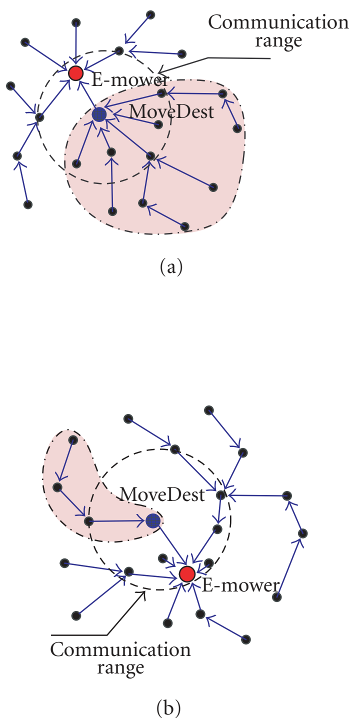

FIGURE 3: Different sojourn positions of the energy mower cause different forwarding workloads of MoveDest.

where k is a constant, E_q is the residual energy of the quasi-hotspot q, and D_q is an estimate distance from q to the edge of the network, which is used to estimate the forwarding workload of the quasi-hotspot q if the energy mower stays at the position c. The direction of the influence force lies in the same direction from q to c, which is illustrated in Figure 2. Equation (1) indicates that if the quasi-hotspot has less energy and reckons that it will have more workload for a certain position candidate, it will generate a stronger influence force on the candidate.

Let C denote the set of the candidates for the sojourn position, and let Q denote the set of the quasi-hotspots. To calculate the composite influence force on a position candidate c $(c \in C)$, the energy mower sets up a coordinate system with the position candidate as the origin and calculates the influence force from each quasi-hotspot q $(q \in Q)$, $\vec{f_c^q}$, according to (1). Suppose the coordinates of c and q are (x_c, y_c) and (x_q, y_q), respectively. The strength of the component forces of f_c^q along the x-axis and the y-axis of the coordinate system can be written as

$$\left(\vec{f_c^q}\right)_X = \left|\vec{f_c^q}\right| \cdot \frac{x_q - x_c}{\sqrt{(y_q - y_c)^2 + (x_q - x_c)^2}},$$
$$\left(\vec{f_c^q}\right)_Y = \left|\vec{f_c^q}\right| \cdot \frac{y_q - y_c}{\sqrt{(y_q - y_c)^2 + (x_q - x_c)^2}}. \quad (2)$$

Therefore, the energy mower can calculate the strength of the composite influence force on the candidate c according to the following equation:

$$\left|\vec{F_c}\right| = \sqrt{\left[\sum_{q\in Q}\left(\vec{f_c^q}\right)_X\right]^2 + \left[\sum_{q\in Q}\left(\vec{f_c^q}\right)_Y\right]^2}. \quad (3)$$

After calculating the composite influence forces for all the position candidates, the energy mower will select the candidate with the minimum value of $|\vec{F_c}|$ as the sojourn position of the current data-gathering period.

4.3. *Case 2: beside MoveDest*

When MoveDest does not change to another sensor node during several data-gathering periods, the energy mower has a chance to arrive at a location close to MoveDest. After entering the communication range of MoveDest, the energy mower expects to find a sojourn position around MoveDest to force MoveDest to forward data for other nodes and consume much energy until it no longer has the highest energy among all the nodes in the network. The energy mower should not be too close to MoveDest or else it would become a *stand-in* for MoveDest and take on most of the reception workload of MoveDest. Therefore, the energy mower should keep a distance of about one hop from MoveDest.

If MoveDest is close to the edge of the network or the nodes near MoveDest are deployed asymmetrically, the

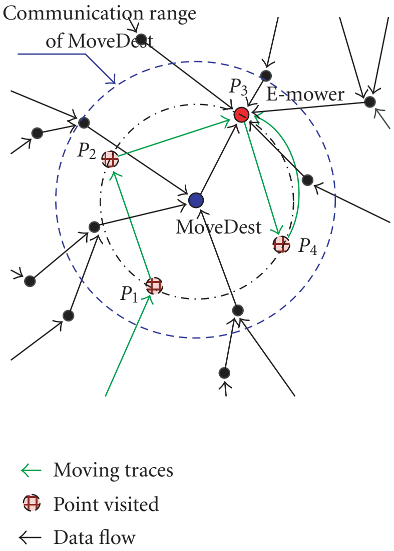

Figure 4: An example of that the energy mower employs the square hopping mechanism to choose a proper sojourn position around MoveDest.

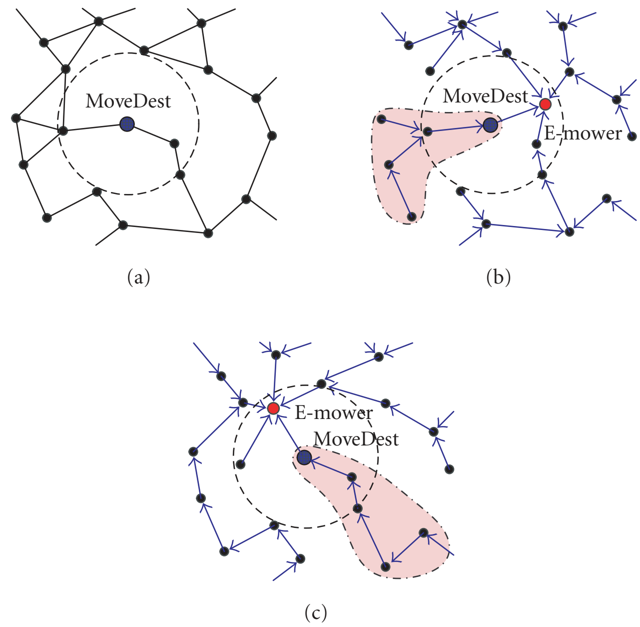

Figure 5: The energy mower cannot make MoveDest take on heavy forwarding workloads because of the topology near MoveDest. (a) Network topology around MoveDest; (b) workload of MoveDest when the energy mower stays in the right of it; (c) workload of MoveDest when the energy mower stays in the up-left of it.

energy mower will decide its sojourn position according to whether the energy of MoveDest can be consumed efficiently, which is illustrated by the example in Figure 3. If the energy mower stays at the position like in Figure 3(a) and gathers data for several periods, MoveDest has to forward data for its five one-hop neighbor nodes and their children nodes. However, if it stays at the position like in Figure 3(b), MoveDest only forwards data for one neighbor node and its children nodes in each period. In the case of Figure 3(b), the energy mower has to spend many data-gathering periods staying beside MoveDest to burn up its energy, which is dangerous for the nodes with inadequate energy in the vicinity of MoveDest.

We propose a square hopping mechanism to help the energy mower to determine a preferred position around MoveDest to sojourn. In the mechanism, the energy mower selects four points uniformly on a circle whose center is MoveDest and the radius is a little smaller than the communication range of MoveDest. The main reason for selecting a smaller radius is to provide a satisfying packet reception rate [34]. The energy mower visits each of the points and stays there for a data-gathering period. When it stays at each point, it records the number of data packets received from MoveDest. After visiting all the points once, the energy mower determines which point is the appropriate position to force MoveDest to transport most data in one data-gathering period. The energy mower then moves back to the point and stays there to gather data until the current MoveDest no longer has the highest energy among all the sensor nodes. If another sensor node becomes a new MoveDest before the energy mower finishes visiting all the points on the circle, the energy mower will give up the old MoveDest and approach the new one.

An example of the usage of the square hopping mechanism is shown in Figure 4. When the energy mower moves to the position P_1, it has entered the communication range of MoveDest; then it chooses four points, P_1–P_4, to visit. After it gathers data for a period at each point, it moves back to P_3 where it can make MoveDest forward the most data.

4.4. Blacklist-based mechanism

Because of the impacts of topology, link quality of communication, and routing strategies, MoveDest perhaps cannot be selected as the next-hop node by most of its one-hop neighbors, thus it may forward only a few data and consume a little energy even if the energy mower stays around it for many periods. For example, if MoveDest has few one-hop neighbor nodes due to the node deployment, as shown in Figure 5(a), wherever the energy mower stays around it, MoveDest forwards data for few nodes, so that MoveDest still keeps high residual energy, such as the situations in Figures 5(b) and 5(c). This problem makes the energy mower incline to select the same node as MoveDest in many data-gathering periods and exhaust the energy of MoveDest's neighbors instead of MoveDest itself. Therefore, we propose a blacklist-based mechanism to prevent the energy mower from being infatuated with these dangerous nodes.

We make the energy mower maintain a blacklist to record the dangerous nodes in the network. When the number of data-gathering periods in which the energy mower selects the same node as MoveDest exceeds a threshold TH_P, and the number of the total data received from the same MoveDest is below another threshold TH_D, the energy mower will add the current MoveDest into the blacklist and temporarily prevent it from being selected as MoveDest again. After a predetermined interval, the energy mower will remove the record entry of the node from the blacklist and give it another chance. The maximum length of the blacklist is determined by the two thresholds and some other factors such as node deployment, node density, and routing protocol.

In another scenario, if a sensor node has the highest energy in the network, and meanwhile it is in the communication range of a quasi-hotspot, the node should not always be selected to be MoveDest because the energy mower will threaten the lifetime of the quasi-hotspot when coming close to it. Therefore, this kind of node should also be put into the blacklist of the energy mower temporarily.

The blacklist-based mechanism protects the low-energy nodes that are near the nodes with the highest energy and helps to balance the energy consumption among the nodes further. Moreover, it is beneficial to the topology adaptability of our moving strategy, in particular, when the node density is low.

5. SIMULATION

5.1. Simulation setup

In our simulation experiments, we adopted the practical radio energy model described in [35]. In this model, the transmitter needs energy to run the radio electronics and a power amplifier, and the receiver consumes energy to run the radio electronics. For a relatively short distance, the propagation loss is modeled as being inversely proportional to d^2, whereas for a longer distance, the propagation loss is modeled as being inversely proportional to d^4. Therefore, to transmit and receive a K-bit packet in a distance d, the radio expends the following energy, respectively:

$$E_{\text{Tx}} = \begin{cases} K \cdot E_{\text{elec}} + K \cdot \varepsilon_{\text{friis-amp}} \cdot d^2, & \text{if } d < d_{\text{crossover}}, \\ K \cdot E_{\text{elec}} + K \cdot \varepsilon_{\text{two-ray-amp}} \cdot d^4, & \text{if } d \geq d_{\text{crossover}}, \end{cases}$$
$$E_{\text{Rx}} = K \cdot E_{\text{elec}}, \tag{4}$$

where $d_{\text{crossover}}$ is the cross-over distance for *Friis* and *two-ray* ground attenuation models. E_{elec} is the electronics energy that depends on factors such as digital coding, modulation, and filtering of the signal before it is sent to the transmit amplifier. The parameters $\varepsilon_{\text{friis-amp}}$ and $\varepsilon_{\text{two-ray-amp}}$ depend on the required sensitivity and the noise figure of the receiver.

We performed our simulations in GloMoSim [36]. We employed CSMA as the MAC protocol and combined our moving strategy with a short-path-like routing protocol, which was described in [37]. The routing protocol compromises between path length and packet loss rate according to the suggestions discussed in [34, 38]. In all our experiments, each sensor node sent a data packet to the energy mower every five minutes and retransmitted the packet for up to three times if an acknowledgment was not received in time. The main simulation parameters are listed in Table 1.

Table 1: Main simulation parameters.

Parameter	Value
Length of the neighbor discovery process	30 seconds
Length of a data-gathering period	300 seconds
Length of the first phase of a period	10 seconds
Length of the second phase of a period	200 seconds
Length of the third phase of a period	60 seconds
Length of a data packet	88 bytes
Length of ACK for data reception	4 bytes
Length of a HELLO message	7 bytes
Length of a position notification message	5 bytes
MAC protocol	CSMA
Radio frequency	433 MHz
Radio bandwidth	19.2 Kbps
Transmission power for sensor nodes	−18 dBm
E_{elec} in the energy model	1.16 μJ/bit
$d_{\text{crossover}}$ in the energy model	40.8 m
$\varepsilon_{\text{friis-amp}}$ in the energy model	5.46 pJ/bit/m^2
$\varepsilon_{\text{two-ray-amp}}$ in the energy model	0.00325 pJ/bit/m^4

We compared the network lifetime performance of HUMS with that of other three data-gathering strategies: a conventional strategy where a stationary sink node locates at the network center, a random moving strategy where a mobile sink moves randomly in network region, and a peripheral moving strategy where a sink moves along the periphery of the network [25]. The peripheral moving strategy was theoretically proved to be a near-optimal moving strategy when an ideal short path routing was employed in [25] because it offered a maximal balance between the sensor nodes near the center of the network and those close to the edge. In this paper, we focus only on the metric of network lifetime because the other metrics such as delay and throughput are mainly determined by the routing protocol and the MAC protocol employed, which are the same in the four strategies under comparison. The network lifetime in this paper is defined as the period of time until the first node dies. We did not compare the performance of HUMS with some reactive moving strategies such as [32, 33] because we think it is not quite reasonable to rudely transplant the strategies designed for event-driven networks to the networks where all the sensors report data periodically. In addition, if these strategies serve in a data-gathering network, the mobile sinks would likely hover near the center of the network and perform closely to the scheme with a stationary sink.

5.2. Experimental results

5.2.1. In regular-shaped networks

In the first group of experiments, 100 sensor nodes with the same initial energy were distributed randomly in a square region of 200 m × 200 m. Figure 6 shows the network lifetimes of the four strategies varied with different initial energy of

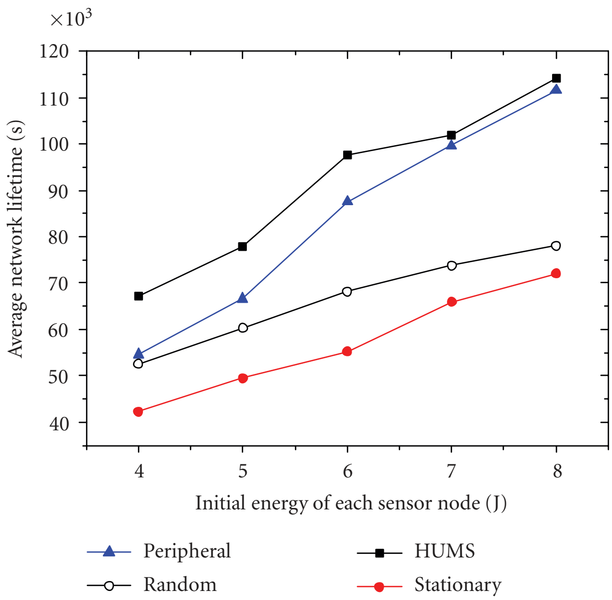

FIGURE 6: Network lifetimes varied with different initial energy for each sensor node.

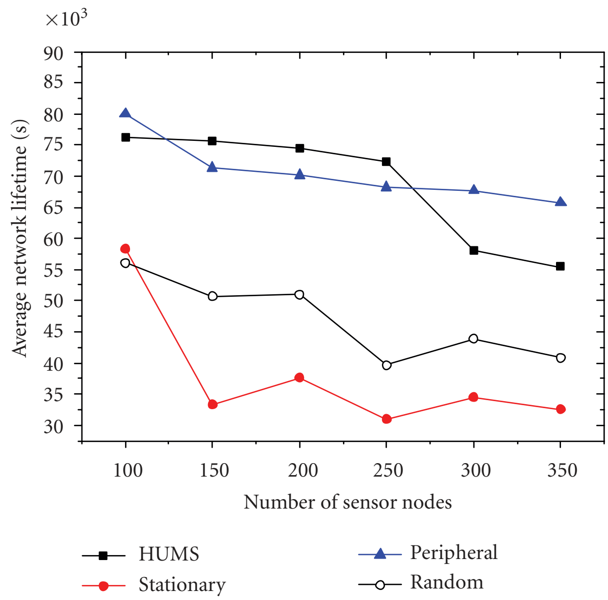

FIGURE 7: Network lifetimes varied with different node densities. (The initial energy for each node is 8 J.)

the sensor nodes. Every dot value in the figure is the average of the results of four experiments in different node deployments. The results indicate that, compared with the stationary strategy, all the other three moving strategies can extend the network lifetime. Moreover, our autonomous moving strategy, HUMS, achieved a higher performance than the other two moving strategies. The main reason of the fact that HUMS performed better than the peripheral moving strategy, which was proved to be near optimal, is because the latter is based on an ideal short path routing. As an energy-unconscious moving strategy, random moving strategy can only extend the network lifetime moderately; meanwhile, the performance of peripheral moving strategy was enhanced fast with the increase in the initial energy for each node and hit values close to that of HUMS.

In the second group of experiments, we studied the scalability of the four strategies by measuring the network lifetimes under different node densities and the results are shown in Figure 7. In the experiments, different numbers of sensor nodes with 8-joule initial energy were randomly deployed in a square region of 300 m × 300 m. The results show that when the node density increased, the network lifetimes of all strategies decreased because the sensor nodes had to forward more data in one data-gathering period, so that the average lifetimes of the sensors nodes decreased. However, compared with the stationary strategy, the lifetime improvement ratio of all moving strategies increased. In addition, the results also show that the performance of HUMS decreased below that of the peripheral moving strategy in the two high-density networks of the experiments. This is mainly because, with a limited moving speed, the energy mower will affect the medium nodes near its moving tracks in the course of approaching to MoveDest. The higher the node density is, the more energy of the medium nodes will be burned up,

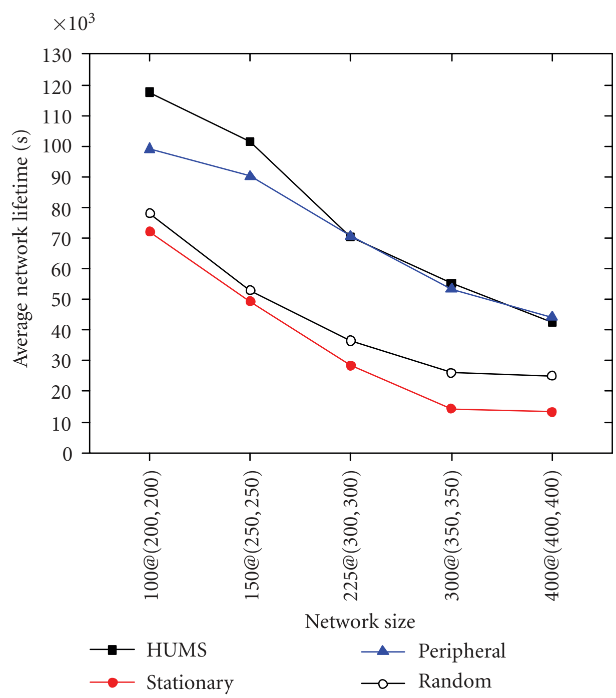

FIGURE 8: Network lifetimes varied with different network sizes. (The initial energy for each node is 8 J.)

so that it will be more difficult for the energy mower to keep an energy balance among all the sensors.

The third group of experiments aimed to evaluate the network lifetime performance of the four strategies when the network size scaled up under the same node density. The size of the network increased from 200 m × 200 m to

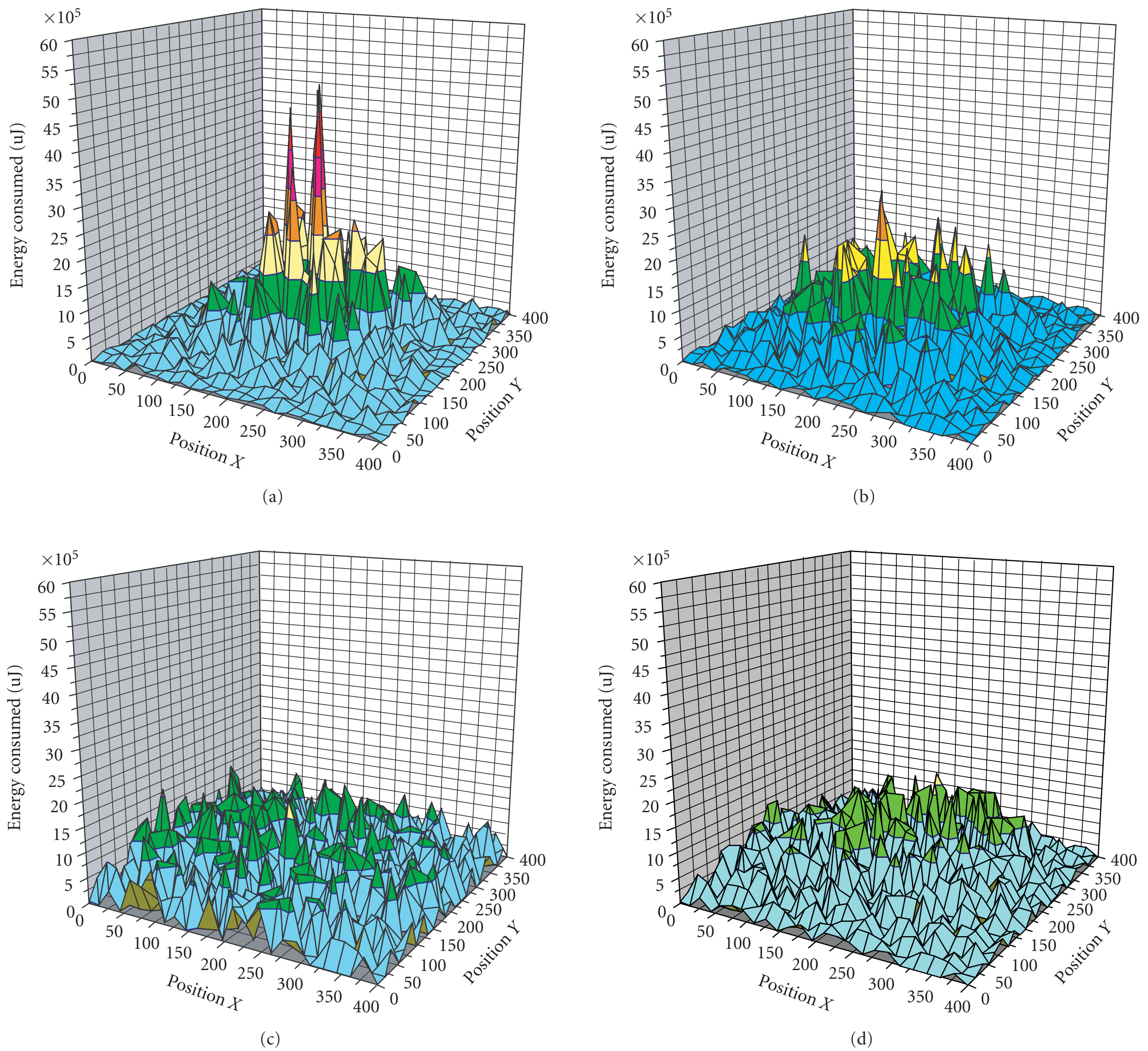

FIGURE 9: The snapshots of the energy consumption of the sensor nodes when the simulations were running in a network that had 400 sensor nodes in a region of 400 m × 400 m. (The initial energy for each node is 8 J.) (a) Stationary scheme; (b) random moving strategy; (c) peripheral moving strategy; (d) HUMS.

400 m × 400 m gradually in our experiments. As shown in Figure 8, the network lifetimes of all the strategies decreased with the increase of the network size. This is because (1) the number of the data packets that should be forwarded to the energy mower increased and (2) when the network size scaled up, the data packets had to experience more hops before they arrived at the energy mower, which further aggravated the burden of the sensor nodes. The results in Figure 8 show that HUMS can still perform well under different network scales. Moreover, compared with the stationary strategy, the lifetime improvement ratios of all moving strategies increased. In particular, the improvement ratios of HUMS and the peripheral strategy reached near 400% when they were employed in the networks that had 400 sensor nodes deployed in a region of 400 m × 400 m.

We can see from the results in Figures 7 and 8 that the performance of HUMS decreased a little faster than that of the peripheral strategy with the increase of the network scale, which implies that the peripheral strategy may work better than HUMS in very large and high-density regular-shaped networks. We captured a snapshot of energy consumption of the sensor nodes for each strategy at the same simulation time when they were running in the networks of 400 m × 400 m, which were shown in Figure 9. In Figure 9, the network region is divided into 25 × 25 cells, and the z-axis denotes the average energy consumption of the nodes in the

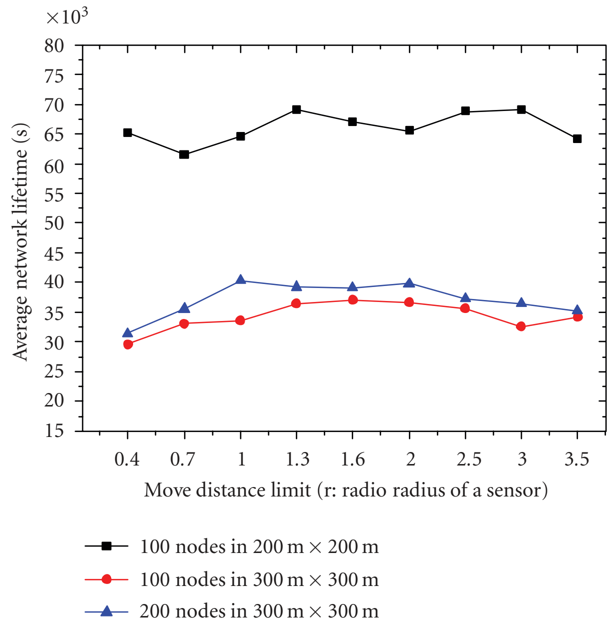

FIGURE 10: Network lifetimes varied with different move distance limits. (The initial energy for each node is 4 J.)

cells. The snapshots show that both HUMS and the peripheral strategy balanced the energy consumption of the sensor nodes and the latter performed a little better.

In the fourth group of experiments, we investigated the impacts of the distance limit of the energy mower for one moving step, which is mentioned in Section 4.1. We think it is necessary to restrict the move distance of the energy mower in one data-gathering period because an actual mobile device, which is perhaps carried by a human, an animal, or a robot, cannot move at an unlimited speed. As mentioned previously, we set the distance limit to the same length as the communication radius of a sensor node in the current implementation of HUMS. However, we found that the distance limit did not affect the lifetime performance of HUMS much. We adjusted the ratio of the distance limit to the communication radius of a sensor node from 0.4 to 3.5 and carried out each simulation in three networks with different sizes and node densities, respectively. As shown in Figure 10, the lifetime performance kept stable when the distance limit increased. Whereas, if the distance limit is very short, the energy mower will take a long time to arrive at MoveDest, and it may actually have no chances to reach MoveDest because it has to change its moving direction as the new MoveDest is selected. This is likely a potential trouble for a network that consists of the sensors nodes with very low initial energy because the sensor nodes that are passed by a slow energy mower are always inclined to be exhausted. Moreover, a short distance limit will prevent the energy mower moving back to the most appropriate position after it finishes visiting all the points around MoveDest with square hopping algorithm, which is described in Section 4.3. On the other hand, if the energy mower can move very fast, we can certainly adopt a long distance limit. In this case, it seems that the energy mower can *jump* to the sides of MoveDest directly and we can omit the half-quadrant algorithm from the programs performed on the energy mower.

5.2.2. In irregular-shaped networks

In many surveillance applications, the terrain shapes are not regular rounds or squares. In this case, if a fixed-track moving strategy is employed, the track has to be established manually as an infrastructure although sensors can be randomly scattered or dropped by planes into the region. In contrast, an autonomous moving strategy can reveal the full advantage on adapting to irregular-shaped networks.

In the fifth group of experiments, we investigated the adaptability of the four strategies. We designed an algorithm to generate irregular network coverage with a given number of sensor nodes. All the networks generated were restricted to a region of 150 m × 200 m and contained a mobile sink and 100 sensor nodes that were randomly deployed. In the simulations for the peripheral strategy, we had to make the sink traverse the tracks that covered the whole square region of 150 m × 200 m because it was hard to plan tracks along the exact edge around each irregular network. Figure 11 shows an example of the irregular network shapes adopted in our experiments. Figure 11(a) is a snapshot of the moving tracks of the energy mower using HUMS, Figure 11(b) is an illustration of the moving tracks of a mobile sink employing the peripheral strategy, and Figure 11(c) shows a run-time snapshot of the random moving strategy. Figure 12 shows the experimental results in the irregular networks and every point of the horizontal scale denotes a different irregular-shaped network. It is obvious that HUMS performed much better than all the other three strategies in the irregular networks, which means that HUMS can provide better adaptability to various network shapes.

HUMS contains several features, such as square hopping, and blacklist-based mechanism. We evaluated the effects of these features on the performance of HUMS, respectively. Figure 13 shows the comparisons among the performances of the full-featured HUMS, HUMS without blacklist, HUMS without square hopping, and HUMS without the energy-leveled threshold control when changing MoveDest, which is described in Section 4.1. We can see from the results that all the features can contribute to HUMS. Particularly, when the blacklist-based mechanism was cut off from HUMS, the lifetime performance swung dramatically in different irregular networks.

6. DISCUSSION

In this section, we discuss some design details of HUMS, which are explained briefly in the previous sections.

(1) The number of the sectors divided by the energy mower when making moving decisions

We named our moving strategy a *half-quadrant-based* strategy because we made the energy mower divide the coordinate

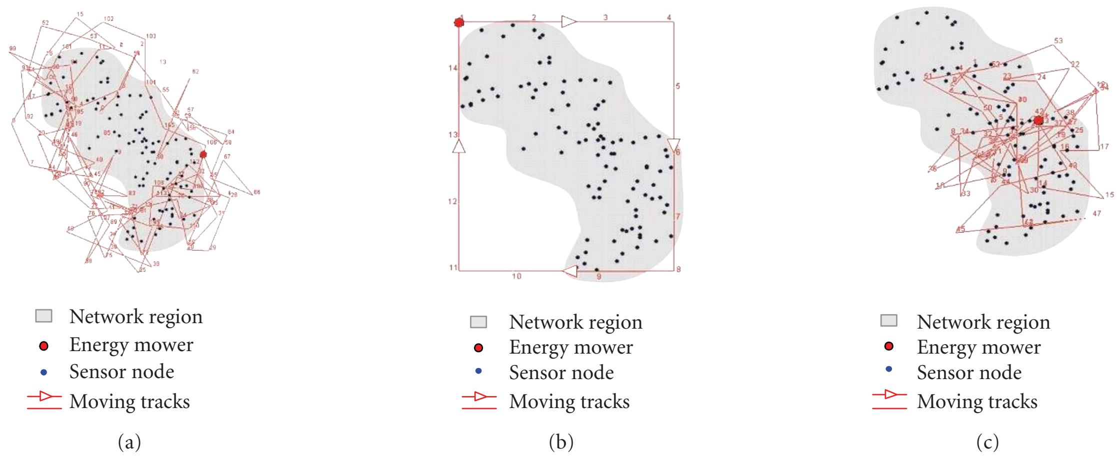

FIGURE 11: An example of the moving tracks of the energy mower that employed different moving strategies in an irregular network. (a) HUMS; (b) peripheral moving strategy; (c) random moving strategy.

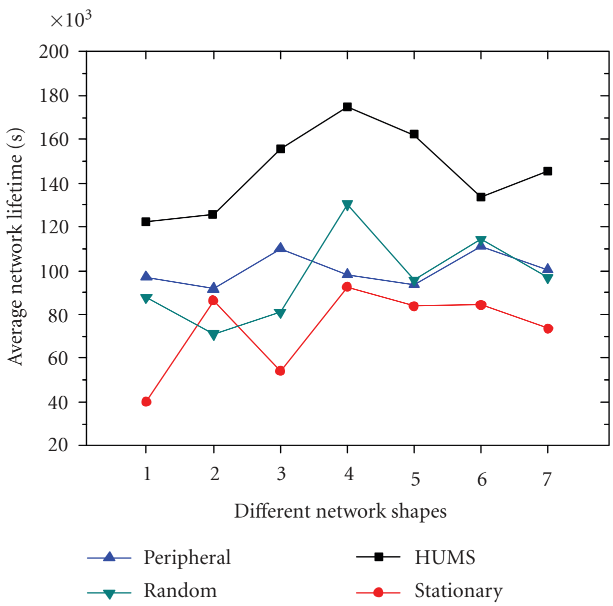

FIGURE 12: Network lifetimes varied with different network shapes. (The initial energy for each node is 8 J.)

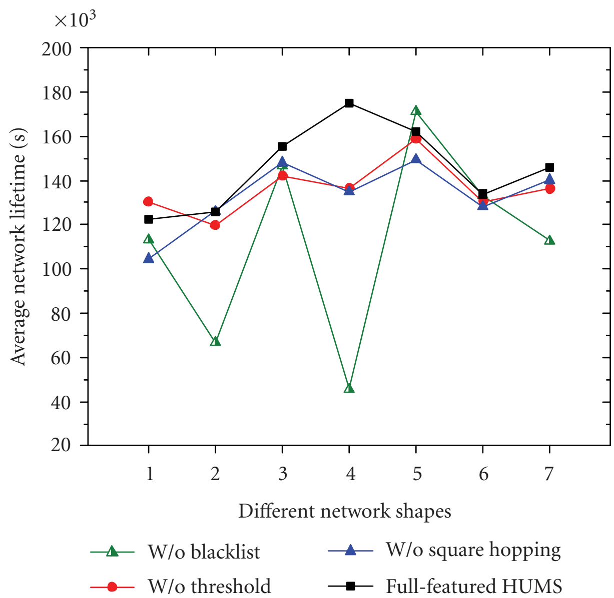

FIGURE 13: Network lifetimes of the variations of HUMS when cutting off different features. (The initial energy for each node is 8 J.)

system into eight sectors, which was described in Section 4.1. Actually, the number of sectors is not limited to eight. We tried different numbers of sectors from 6 to 10 and found that it did not affect the performance of HUMS much. The reason for adopting eight sectors in the current implementation of HUMS is mainly because it makes the calculation of point coordinates much easier. However, the optimum number of the sectors should be related to the node density, the network scale, and so on, and it may be worth investigating further.

(2) The number of the points selected in the square hopping mechanism

In our square hopping mechanism, the energy mower will select four points uniformly around MoveDest. The number of the points should not be big (e.g., 20) mainly because of the following two reasons: (1) it will take a long time for the energy mower to finish visiting all the points and likely burn up the energy of the nodes near MoveDest; (2) it is probably not necessary to make the energymower move with a small

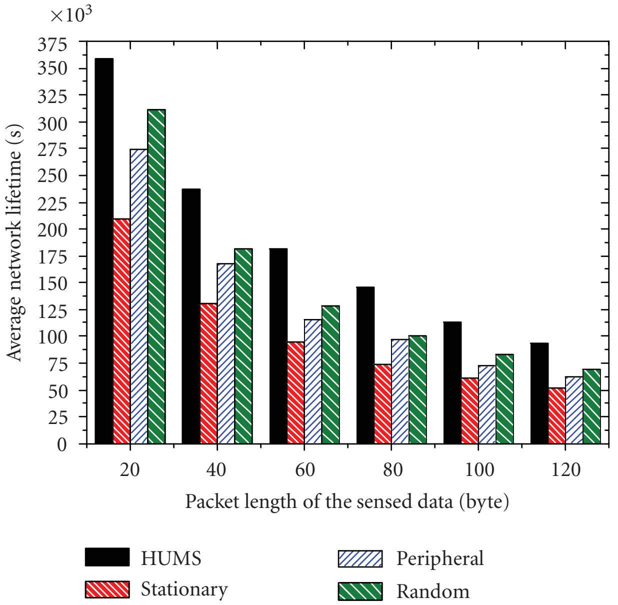

Figure 14: Network lifetimes varied with different packet sizes of the sensed data. (The initial energy for each node is 8 J.)

step around MoveDest because the routing topology changes little if the energy mower moves slightly, so that the traffic flows would not change much. On the other hand, if the energy mower selects few points to visit, it can hardly find an appropriate sojourn position to burn the energy of MoveDest fast. Therefore, we recommend that the number of the points can be from 3 to 10 according to different node densities, and the higher the node density is, the bigger the number of points can be.

(3) The overhead of our data-gathering scheme and moving strategy

The overhead of our solution is mainly composed of the notification of the positions of the energy mower, the additional information carried by the data packets to help the energy mower make moving decisions, and the cost of moving the energy mower. The position notification for the energy mower does not necessarily introduce extra overhead because many routing protocols proposed for the sensor networks with a stationary sink require diffusing messages across the whole network periodically [5–7]. Actually, as mentioned in Section 3, if the network size is not very large, the energy mower can send the position notification with a long communication radius so that the sensors need not forward the notification, which can preserve much energy of the sensors. Moreover, in HUMS, the energy mower will keep static when the sensor nodes are reporting data to it, which can avoid bringing the overhead of maintaining the delivery paths to a mobile sink.

In our solution, to make the energy mower move autonomously, each data packet has to carry some additional information. We attached 8 bytes to the original sensed data packet. We evaluated the overhead by simulations, whose results are shown in Figure 14. We adopted different packet sizes to contain sensed data. For example, when the packet size was 20 bytes, which was the smallest in our experiments, HUMS would use a 28-byte packet; meanwhile, the other three strategies in the experiments would use a 20-byte packet. The height of the columns in Figure 14 denotes the average lifetimes of all the irregular networks and the results show that the overhead caused by the attached bytes is negligible because HUMS performed better than the other three strategies when the data size varied from 20 bytes to 120 bytes.

On the other hand, this overhead caused by the extra bytes in data packets can be reduced further in the future. A possible solution is to design a local probability-based election algorithm to suppress most of the sensor nodes to attach useless additional information to their data packets. In this way, we can make only a few sensor nodes generate lengthened packets and most sensor nodes generate original-length packets in each data-gathering period.

The cost of moving an energy mower is another part of the cost of the whole network system, although it has not been taken into consideration in HUMS. However, enhancing the capability of the mobile sink to improve the energy efficiency of the sensor nodes is worth employing a proactive moving strategy, which often costs more than adopting a reactive moving strategy.

7. CONCLUSION

In this paper, we have presented a data-gathering scheme for sensor network with a mobile sink. In this scheme, we distribute three key tasks of a data-gathering period, which are moving the sink, collecting data, and notifying sensors of the sink's positions into three separate phases. The scheme provides design flexibility because of the loose coupling among the three phases. Under the scheme, we have proposed an autonomous moving strategy to take advantage of sink mobility to balance energy consumption among sensor nodes and prolong network lifetime. The proposed strategy can make a mobile sink act as an energy mower and try to cut the energy lawn in the network to a flat one, which results in a balance of the energy consumption in the network. We have compared the performance of network lifetime of our moving strategy with those of a stationary strategy, a random moving strategy, and a near-optimal fixed-tracking moving strategy. The experimental results show that the proposed moving strategy can extend network lifetime notably and provide better adaptability to irregular-shaped networks than the other three solutions.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China Grant no. 60673178, the National Basic Research Program of China (973 Program) Grant no. 2006CB303000, the National High Technology Research and

Development Program of China (863 Program) Grant no. 2006AA01Z218 and no. 2006AA01Z215. The preliminary version of this paper was accepted as a poster abstract in IEEE SECON 2006.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 38, no. 4, pp. 393–422, 2002.

[2] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–31, 2005.

[3] M. Haenggi, "Mobile sensor-actuator networks: opportunities and challenges," in *Proceedings of the 7th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA '02)*, pp. 283–290, Frankfurt, Germany, July 2002.

[4] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[5] A. Boukerche and S. Nikoletseas, "Protocols for data propagation in wireless sensor networks: a survey," in *Wireless Communications Systems and Networks*, pp. 23–51, Kluwer Academic Publishers, Boston, Mass, USA, 2004.

[6] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[7] D. Niculescu, "Communication paradigms for sensor networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 116–122, 2005.

[8] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA '03)*, pp. 30–41, Anchorage, Alaska, USA, May 2003.

[9] L. Tong, Q. Zhao, and S. Adireddy, "Sensor networks with mobile agents," in *Proceedings of IEEE Military Communications Conference (MILCOM '03)*, vol. 1, pp. 688–693, Boston, Mass, USA, October 2003.

[10] S. Jain, R. C. Shah, G. Borriello, W. Brunette, and S. Roy, "Exploiting mobility for energy efficient data collection in sensor networks," in *Proceedings of the 2nd IEEE/ACM Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '04)*, Cambridge, UK, March 2004.

[11] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '02)*, pp. 96–107, San Jose, Calif, USA, October 2002.

[12] S. Ganeriwal, A. Kansal, and M. B. Srivastava, "Self aware actuation for fault repair in sensor networks," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 5, pp. 5244–5249, New Orleans, La, USA, April-May 2004.

[13] A. Kansal, M. Rahimi, D. Estrin, W. J. Kaiser, G. J. Pottie, and M. B. Srivastava, "Controlled mobility for sustainable wireless sensor networks," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pp. 1–6, Santa Clara, Calif, USA, October 2004.

[14] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 1, pp. 377–381, San Francisco, Calif, USA, December 2003.

[15] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05)*, p. 287, Big Island, Hawaii, USA, January 2005.

[16] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, "Controllably mobile infrastructure for low energy embedded networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 958–973, 2006.

[17] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 111–124, Boston, Mass, USA, June 2004.

[18] A. Chakrabarty, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of a sensor network," in *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03)*, pp. 129–145, Palo Alto, Calif, USA, April 2003.

[19] W. Wang, V. Srinivasan, and K.-C. Chaing, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, pp. 270–283, Cologne, Germany, August 2005.

[20] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '05)*, pp. 300–308, Urbana-Champaign, Ill, USA, May 2005.

[21] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *Proceedings of the 24th Annual Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 4, pp. 2302–2312, Miami, Fla, USA, March 2005.

[22] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Communication power optimization in a sensor network with a path-constrained mobile observer," *ACM Transactions on Sensor Networks*, vol. 2, no. 3, pp. 297–324, 2006.

[23] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C.-G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pp. 386–395, Santa Clara, Calif, USA, September 2005.

[24] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS '04)*, pp. 296–305, Lisbon, Portugal, December 2004.

[25] J. Luo and J.-P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings of the 24th Annual Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1735–1746, Miami, Fla, USA, March 2005.

[26] P. Sikka, P. Corke, P. Valencia, C. Crossman, D. Swain, and G. Bishop-Hurley, "Wireless ad hoc sensor and actuator networks on the farm," in *Proceedings of the 5th International Conference*

on Information Processing in Sensor Networks, pp. 492–499, Nashville, Tenn, USA, April 2006.

[27] Y. Chen, Z. Wang, and J. Liang, "Optimal dynamic actuator location in distributed feedback control of a diffusion process," in *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC '05)*, pp. 5662–5667, Seville, Spain, December 2005.

[28] Z. Song, Y. Chen, J. Liang, and D. Ucinski, "Optimal mobile sensor motion planning under nonholomonic constraints for parameter estimation of distributed parameter systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 3163–3168, Edmonton, Alberta, Canada, August 2005.

[29] K. L. Moore, Y. Chen, and Z. Song, "Diffusion-based path planning in mobile actuator-sensor networks (MAS-Net): some preliminary results," in *Proceedings of Intelligent Computing: Theory and Applications II*, vol. 5421 of *Proceedings of SPIE*, pp. 58–69, Orlando, Fla, USA, April 2004.

[30] A. Bogdanov, E. Maneva, and S. Riesenfeld, "Power-aware base station positioning for sensor networks," in *Proceedings of the 23rd Annual Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 1, pp. 575–585, Hong Kong, March 2004.

[31] J. Pan, L. Cai, Y. T. Hou, Y. Shi, and S. X. Shen, "Optimal base-station locations in two-tiered wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 5, pp. 458–473, 2005.

[32] K. Akkaya, M. Younis, and M. Bangad, "Sink repositioning for enhanced performance in wireless sensor networks," *Computer Networks*, vol. 49, no. 4, pp. 512–534, 2005.

[33] Z. Vincze, D. Vass, R. Vida, A. Vidács, and A. Telcs, "Adaptive sink mobility in event-driven multi-hop wireless sensor networks," in *Proceedings of the 1st International Conference on Integrated Internet Ad Hoc and Sensor Networssks (InterSense '06)*, Nice, France, May 2006.

[34] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 14–27, Los Angeles, Calif, USA, November 2003.

[35] W. B. Heinzelman, *Application-specific protocol architectures for wireless networks*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Mass, USA, June 2000.

[36] GloMoSim, http://pcl.cs.ucla.edu/projects/glomosim/.

[37] Y. Bi, N. Li, and L. Sun, "DAR: an energy-balanced data gathering scheme for wireless sensor networks," to appear in *Computer Communications*, special issue on Network Coverage and Routing Schemes for Wireless Sensor Networks.

[38] I. Stojmenovic, A. Nayak, and J. Kuruvila, "Design guidelines for routing protocols in ad hoc and sensor networks with a realistic physical layer," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 101–106, 2005, Ad Hoc and Sensor Networks Series.