

# Adaptive Sink Mobility in Event-Driven Densely Deployed Wireless Sensor Networks

ZOLTÁN VINCZE<sup>1</sup>, DOROTTYA VASS<sup>1</sup>, ROLLAND VIDA<sup>1</sup>,  
ATTILA VIDÁCS<sup>1</sup> AND ANDRÁS TELCS<sup>2</sup>

<sup>1</sup>*Department of Telecommunications and Media Informatics,  
E-mail: vidacs@tmit.bme.hu*

<sup>2</sup>*Department of Computer Science and Information Theory,  
Budapest University of Technology and Economics,  
Magyar tudósok krt. 2, H-1117 Budapest, Hungary*

*Received: September 11, 2006. Accepted: October 30, 2006. In Final from: April 7, 2007.*

Optimizing energy consumption in wireless sensor networks is of paramount importance. There is a recent trend to deal with this problem by introducing mobile elements (sensors or sink nodes) in the network. The majority of these approaches assume time-driven scenarios and/or single-hop communication between participating nodes. However, there are several real-life applications for which an event-based and multi-hop operation is more appropriate. In this paper we propose to adaptively move the sink node inside the covered region, according to the evolution of current events, so as to minimize the energy consumption incurred by the multi-hop transmission of the event-related data. Both analytical and simulation results are given for two optimization strategies: minimizing the overall energy consumption, and minimizing the maximum load on a specific sensor respectively. We show that by adaptively moving the sink, significant power saving can be achieved, prolonging the lifetime of the network.

*Keywords:* Wireless sensor networks, energy efficiency, lifetime, controlled mobility, multi-hop communications.

## 1 INTRODUCTION

Wireless sensor networks constitute an emerging technology that has received recently significant attention both from industry and academia. On the one hand, there is an ever-widening range of attractive applications (e.g., disaster

and environmental monitoring, wildlife habitat monitoring, intrusion detection, security surveillance) sensor networks can be used for. On the other hand, sensor networks are self-organizing ad-hoc systems where optimized energy consumption is of paramount importance; therefore, relaying information between sensors and a sink node, possibly over multiple wireless hops, in an energy-efficient manner is a challenging task that preoccupies the research community for some time now.

Sensors are tiny devices with sensing, processing, and transmitting capabilities; they are of low cost, but have a consequently low storage and computational capacity, and a limited energy supply. It is usually considered impossible or impractical (from a technical or economical point of view) to recharge their batteries; thus, they should be managed in such a way to ensure the unattended operation of the network for a long enough time period (e.g., several months).

Information gathering in sensor networks can follow different patterns, depending mostly on the specific needs of the applications. In a time-driven scenario all sensors send data periodically to the sink. As opposed to this, in the event-driven case sensors start communicating with the sink only if sensing an *event*, i.e., a situation that is worth reporting. Finally, in a query-driven scenario a sensor transmits its data only if the sink asks for it. Most of the research papers in the area address the time-driven scenario, and provide energy-efficient solutions for homogeneous networks, with sensors having constant and equal amounts of data to send in all parts of the covered region. Previous papers showed that in such a case of uniform distribution and uniform reporting, the network cannot be energy balanced if a single static sink is used [21]. However, there are a large number of applications (e.g., intrusion detection, seismic activity monitoring, animal movement tracking) where an event-driven approach is more appropriate. Hence, in our paper we address only this scenario.

As we noted before, energy efficiency is the main requirement for the operation of a sensor network. Sensors consume energy for sensing the field, for digitizing and processing the data, but the most penalizing task is by far the transmission of the information [24]. In the most commonly accepted power attenuation model [27], signal power falls as  $d^{-\alpha}$ , where  $d$  is the distance from the transmitter antenna and  $\alpha$  is a constant dependent on the wireless transmission environment, typically between 2 and 4. Therefore, assuming that all receivers have the same power threshold for signal detection, typically normalized to one, the energy required to support communication between the two nodes is  $d^\alpha$ . In such conditions it is straightforward to assert that by minimizing the distance between a sensor and a sink node we can efficiently reduce power consumption, both for single- and multi-hop communications (reducing the length of the multi-hop path results in fewer and/or shorter hops, i.e., less energy is needed to relay data to the sink).

Besides analyzing the general case of an event-driven scenario, we intend also to have a closer look on a specific example where events move inside the observed region following a correlated random walk model. There are several



concrete use cases this example can be relevant for. In [4] authors show that animal movements can be described as a correlated random walk. A similar result is obtained in [1] for the specific case of caribous. Moreover, the model should fit intrusion detection and target tracking applications as well. In this paper we propose to analyze, both analytically and through simulations, the efficiency of adaptively moving the sink node so as to react to dynamic events that follow a correlated random walk mobility model. Results are compared to three alternative solutions: the case of a static sink, the case of random mobility (the sink moves randomly, e.g., according to the random waypoint mobility model, independently of the events), and the case of predictable mobility (the sink moves on a predefined path, e.g., on the periphery of the network [19], independently of the events).

The rest of this paper is organized as follows. In Section 2 we present related work in the area of energy optimization and sink mobility. In Section 3 we describe the assumed network model, and calculate the overall energy requirement an event poses on the network, as well as the maximum energy consumption of a specific sensor. According to these analytical results, in Section 4 we show how to find the optimal position of the sink inside the network so as to minimize overall or maximum energy consumption. However, this optimal position is specific only to a given snapshot of events that are present in the network. Moreover, in a real application the sink cannot usually move directly to the optimal position, it can only take a step towards it in a certain period of time. Therefore, to continuously optimize energy consumption in the case of dynamically evolving events, we should give efficient strategies for adaptive sink mobility. This is the topic of Section 5. First we present the assumed dynamic event model, and then give an analytical description of how to choose the next location within the reach of the sink for the two strategies (minimizing the overall or the maximum consumed energy). In Section 6 we present some implementation issues, such as routing or updating sink location information, that have to be dealt with in a real scenario. Section 7 presents our simulation results, while Section 8 concludes the paper.

## 2 RELATED WORK

There were many proposals recently targeting the energy efficiency of wireless sensor networks. Some approaches focused on energy conserving routing techniques, i.e., finding optimal routes in terms of consumed power, and balancing the energy consumption among all nodes [8,14,28,39]. Others were based on topology control schemes, i.e., deploying sensor and sink nodes in an efficient way or reshaping the topology through dynamic power control of the participating sensors [6,18,23,26,35]. Clustering techniques are also widely employed; the network is divided into small clusters, a cluster head being responsible for aggregating and relaying towards the sink the information gathered from the sensors of its cluster [15,37].

In all the above approaches the elements of the network are all considered static. However, there is a recent trend to explore mobility as a way of enhancing energy efficiency. In [5] sensors dynamically react to the environmental changes and move towards areas where events occur frequently. In [33] sensor mobility is exploited at the deployment phase, to eliminate coverage holes that are discovered through the use of Voronoi diagrams. Mobile sensors are also considered in [29] to provide an extension of a stationary sensor network.

Moving the sink node is also a widely explored solution. The approaches can be classified into three categories: random, predictable, and controlled mobility of the sink. In [25] the authors propose an architecture that builds on the *random mobility* of mobile agents, called data MULEs (Mobile Ubiquitous LAN Extensions), to collect sensor data in sparsely deployed networks. A similar approach, but for dense networks, is used by SENMA (Sensor Networks with Mobile Agents) [30]; data is sent directly to the mobile agent that is flying above the sensor field, the transmission being triggered based on the estimated fading state of each sensor in its communication with the agent. [11] uses a random walk model for a mobile relay to theoretically derive parameters such as delay and data delivery ratio. A serendipitous movement of the sink nodes is also assumed in [16]. However, the authors propose an inversed scenario, where there is a single sensor that transmits data to a large number of mobile sinks. They describe the SEAD (Scalable Energy-efficient Asynchronous Dissemination) protocol to build and maintain an energy-efficient dissemination tree that covers all the sink nodes.

A *predictable mobility* solution is presented in [7]. The sink (called observer) moves along a predefined path, and pulls data from sensors in single-hop communication when arriving near to them. A predefined path of the sink is used in [19] as well. The authors show that moving the sink at the periphery of the covered circular region ensures energy-efficient operation; the approach is considerably different from other mobile sink solutions in that it assumes multi-hop communication between all the sensors and the sink.

There are also several solutions that propose a *controlled mobility* of the sink nodes. In the AIMMS (Autonomous Intelligent Mobile Micro-server) system [13,12] a mobile micro-server moves across the network, along a specific trail, to route data from the deeply embedded nodes. Its mobility is controlled in order to spend extra time (e.g., stop or slow down) in regions where there is a large amount of data to send or the communication channel is constrained. The idea of using mobile nodes for message ferrying is also considered in [38]; these nodes provide non-random proactive routes in highly-partitioned wireless ad-hoc networks.

An attempt to determine specific sink movements for energy optimization is presented in [9]. The authors argue that multi-hop communication results in the sensors neighboring the sink being depleted at a fast pace. Therefore, they propose to employ multiple sink nodes that periodically change their locations, and present an ILP (Integer Linear Programming) model to obtain

the optimal positions of these sinks. A linear programming solution to determine the movement of the sink and its sojourn time in different points of the network is given in [34] as well. Both the sensors and the sink are placed on a bi-dimensional grid. The sink moves along the grid, sojourn times in the specific grid points being calculated so as to maximize the network lifetime.

Finding the optimal position of the sink is addressed in [2] as well, even if mobility is not involved. The authors assume a time-driven scenario, where all sensors send data at a constant rate; the problem is how to deploy  $n$  sink nodes so as to ensure an energy-efficient operation of the network. [22] addresses the static deployment problem as well, as it proposes to find the optimal locations of multiple sinks in sparse networks of aggregator points that send data directly to these sinks.

In this paper we propose a solution that is significantly different from all the above approaches. We assume an event-driven scenario, where sensors that detect an event send data to the sink node in a multi-hop manner. (For the single-hop communication case in clustered sensor networks, please refer to [31].) Our goal is to control the mobility of the sink so as to ensure an energy-efficient operation of the network. The sink node is alerted about the current events, estimates the evolution of these events, and decides about the optimal neighboring place where it should move so as to maximize network lifetime.

### 3 ENERGY CONSUMPTION IN EVENT-DRIVEN NETWORKS

This section gives the description of our assumed network model as well as the analytical calculations on the energy requirements of network operation. The network communication is event-driven in the sense that whenever the sensor nodes detect something that is worth reporting, their task is to transmit a message to the sink in a multi-hop manner. Typically, several sensors report on the same event at a time; their messages are sent in parallel to the sink, making the overall traffic load on relaying nodes significant. The energy requirement of radio communication primarily depends on how far the packets must propagate to reach the sink. The larger the communication distance, the higher the energy consumption. Both the total energy requirement and the maximum load on a particular network node are calculated in this section. Based on these results, we propose an efficient sink relocation strategy to significantly decrease power consumption (see Section 4 later). (A simplified version of the proposed analytical model was given in [32].)

#### 3.1 Analytical Model

We assume a dense and strongly connected network (see Figure 1). There are  $N$  sensor nodes distributed within a circle of radius  $R$ , and a single sink node  $S$  placed at location  $(x_S, y_S)$  to collect the data. The sensor nodes are distributed evenly with density  $\rho$  within a circle of  $\mathcal{C}_{O,R}$ . Thus, the total number of sensor

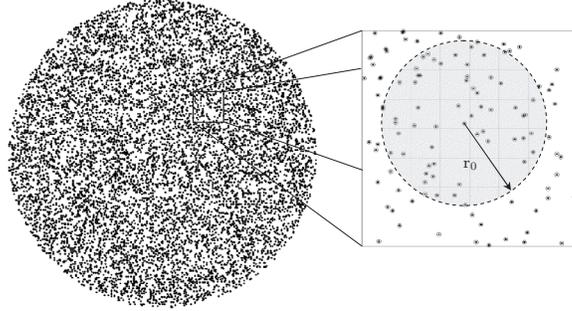


FIGURE 1  
The sensor field.

nodes  $N$  is  $\rho R^2 \pi$ . The sensing range of each sensor is  $r_0$ . The network is event-driven, i.e., whenever an *event*  $Z$  occurs at location  $(x_Z, y_Z)$ , all sensors that are within a circle  $\mathcal{C}_{Z, r_0}$  become *active*. All active sensor nodes generate a message for the sink, and repeat it periodically, until the event persists. Since each node is only able to communicate with neighbors within its radio range  $r_f$ , the message must be routed towards the sink hop-by-hop. We assume multi-hop communication with an ideal short path routing, and without data aggregation.

### 3.2 Events

From the abstract modeling point of view, we call an event any situation that is worth reporting to the sink (e.g., an “intruder” is sensed by the sensors within the monitored area). In general, an *event* is a random subset  $Z(t)$  of  $\mathcal{C}_{O, R}$ . The event activates at time  $t$  the  $i$ th sensor at location  $p_i$  for which

$$d(p_i, Z(t)) \leq r_0. \quad (1)$$

$Z(t) = \emptyset$  by definition before the event occurs and after it left the field or disappeared (e.g., fall below the measure sensitivity of the sensors). Such activation is represented by a pair of time and location  $(t, p)$ . The events are random in space and time, and their location can change during their lifetime. This means that the same event may trigger several  $(t_i, p_i)$  pairs (with the same or different  $t_i$  and/or  $p_i$ ). In our model events are modeled as single points (or locations) within the sensor field. When taking only a “snapshot” of the system at a particular time instant, all existing events at that time are given by their location coordinates only. Assuming  $I$  events, the coordinates of the  $i$ th event  $Z^i$  are  $(x_{Z^i}, y_{Z^i})$ ,  $i = 1, 2, \dots, I$ . All sensors that are closer to an event than their sensing range  $r_0$  become active. In other words, all sensors within the circles  $\mathcal{C}_{Z^i, r_0}$ ,  $i = 1, 2, \dots, I$  are active.

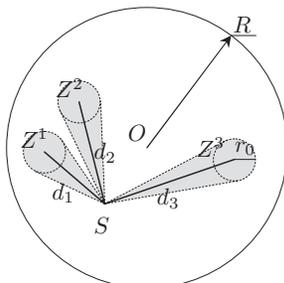


FIGURE 2  
Active sensor areas for three events.

### 3.3 Total Energy

To calculate the overall energy requirement that an event poses on the network, we proceed as follows. The task is to add up all the transmission energies for messages generated by all sensors that were activated by the event. As a result of multi-hop communication, sensors are communicating not only when they are sensing an event but also when they are forwarding the reports of other active sensors. In our model for the analytical calculations we assume ideal short path routing, that is, the sensors are deployed densely and evenly enough to find a straight linear path towards the sink. Thus, all sensors that are in between an active sensor and the sink node will also be active during the communication. Figure 2 denotes all active regions (marked as gray), assuming three events as an example. The distance  $d_i$  denotes the (geographical) distance of event  $Z^i$  from  $S$ , i.e.,

$$d_i = \sqrt{(x_{Z^i} - x_S)^2 + (y_{Z^i} - y_S)^2}, \quad i = 1, 2, \dots, I. \quad (2)$$

The total energy needed to report an event to the sink is directly proportional with the number of active sensors  $N_s$  that are sensing that event and the *average* hop count  $\bar{k}$ , i.e.,

$$E_{total} = N_s \bar{k} E_{hop}. \quad (3)$$

where  $E_{hop}$  is the energy required to pass a message at distance  $h$  in one hop. Knowing that an event  $Z$  activates sensors that are within sensing range, i.e., within  $\mathcal{C}_{Z, r_0}$ , the number of sensing nodes ( $N_s$ ) in our model is  $r_0^2 \pi \rho$ . The average number of hops can be well approximated as

$$\bar{k} \approx \max(1, d/h + 0.5), \quad (4)$$

where we also took into account that the sensing range  $r_0$  is smaller than the hop length  $h$ . (See Appendix A for the justification of the approximation.)

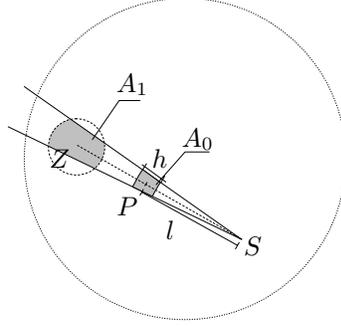


FIGURE 3  
Approximating transit load.

Thus, by substituting (4) into (3) we approximate  $E_{total}$  by

$$E_{total} \approx \begin{cases} (d/h + 0.5)r_0^2\pi\rho E_{hop}, & \text{if } h/2 < d \\ r_0^2\pi\rho E_{hop}, & \text{if } d \leq h/2 \end{cases} \quad (5)$$

Assume that there are  $I$  events on the sensor field instead of one. In this case, the total energy requirement of the whole network is given by

$$E_{total}^{SN} = \sum_{i=1}^I E_{total}^i, \quad (6)$$

where  $E_{total}^i$  is the energy required to report  $Z^i$  to the sink, and is given by (5).

### 3.4 Transit Load and Maximum Energy

Sensors can sense an event and forward packets from other nodes at the same time. Furthermore, one sensor can be requested to forward (much) more than one packet towards the sink, even if it is far from any events to be sensed. This happens to sensors that are close to the sink node, and results in highly uneven load distribution, which plays a key role in our investigations.

To calculate the transit load on a given node, we proceed as follows. As Figure 3 shows, the average *transit* load  $L_P$  on a particular node  $P$  is approximated as

$$L_P \approx A_1/A_0, \quad (7)$$

where nodes within area  $A_0$  must relay messages originating from nodes within  $A_1$  that sense the event directly.

Based on our analytical model (see Appendix B for more details), the transit load can be approximated as

$$L_P = \begin{cases} \frac{4dr_0}{h^2}, & \text{if } l \leq \frac{h}{2} \\ \frac{2dr_0}{lh}, & \text{if } \frac{h}{2} < l < d - \left(r_0 + \frac{h}{2}\right). \end{cases} \quad (8)$$

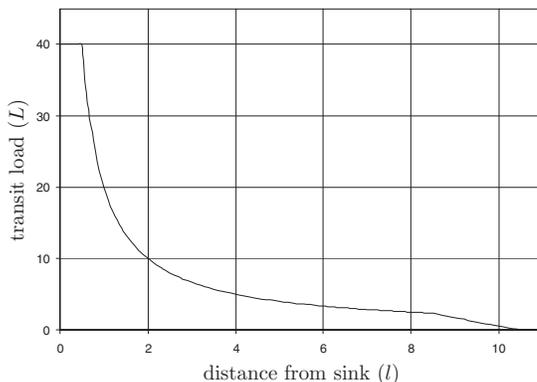


FIGURE 4

Transit load as a function of the distance from the sink ( $l$ ). (The event distance from the sink is  $d = 10r_0$ ,  $h = 1$ .)

The transit load  $L_P$  is plotted on Figure 4 as a function of the distance from the sink. As the figure shows, the load increases hyperbolically when approaching the sink, and is maximal when the transit node  $P$  is only a half hop away from the sink.

The energy requirement of the most loaded sensor node ( $E_{\max}$ ) can be approximated using (8), i.e.,

$$E_{\max} = L_{(h/2)} E_{hop} = \frac{4dr_0}{h^2} E_{hop}. \quad (9)$$

Thus,  $E_{\max}$  is a linear function of the distance  $d$  between the sink and the event location.

Assume again that there are more than one event at a time. In this case, using (9) we can identify for each event  $Z^i$  ( $i = 1, \dots, I$ ) the most heavily loaded sensor with energy requirement  $E_{\max}^i$ . By comparing these highly loaded sensors on the sensor field, we get the highest energy requirement by

$$E_{\max}^{SN} = \max_{1 \leq i \leq I} E_{\max}^i. \quad (10)$$

This energy load is on the sensor that is close to the sink and in the direction towards the most distant event  $Z^j$ , that is given by

$$j = \arg \max_{1 \leq i \leq I} d(Z^i, S). \quad (11)$$

We should note, that here we neglected the fact that one sensor could take part in relaying messages of more than one event at a time. However, since the most loaded sensors are on the line between the event and the sink, and the load decreases rapidly if the sensor is further away from this line (see Figure 16 in Appendix B), we basically neglect only the case when there are two or more events directly “behind” each other.

## 4 OPTIMAL SINK LOCATION

After calculating the total as well as the maximal communication energy that is required to report the events to the sink, we can derive the optimal location where the sink should be placed in order to decrease power consumption and thus extend the network lifetime.

The so-called “facility location” is a classical problem of operations research that has also been examined in the computational geometry community. The task is to position a point in the plane (the *facility*, which is the sink in our case) such that the distance between the facility and given points (active sensors) is minimized or maximized. The optimal facility location is NP-hard, thus, the problem is usually solved using either a hill-climbing heuristic or linear programming.

### 4.1 Minimizing Total Energy

In our case, the first idea is to place the sink node so as to minimize the overall energy consumption of the network. Since there can be more than one event at a time, the task is to minimize  $E_{total}^{SN}$  given by (6). Since the energy requirement of reporting an event is proportional to the event distance  $d_i$  (see (5)), this is equivalent to minimize the sum of event distances, i.e.,

$$\sum_{i=1}^I \max(h/2, d_i) \rightarrow \min, \quad (12)$$

where the maximum means that there is no gain when moving closer to a particular event than the half of the hop length. Practically, this is the location that gives the minimal average distance from the events. There is no closed formula to find this location, but the problem can be solved numerically.

### 4.2 Minimizing Maximum Energy

The problem with the total energy minimization approach could be that—although the overall energy consumption is minimized—it can happen that the energy contributions of the sensors are rather uneven. In order to avoid this problem, one would think of minimizing the transmission energy for the most heavily loaded sensor in the network. Hence, energy consumption will be more balanced. As the maximal traffic load depends on the biggest event distance from the sink node (see (9) and (10)), this strategy is equivalent with that of minimizing the maximum event distance from the sink, i.e.,

$$\max_{1 \leq i \leq I} d_i \rightarrow \min. \quad (13)$$

This minimization task is equivalent to the Minimal Enclosing Circle Problem, where the task is to find the minimum radius circle that encloses all points of a point set on the plane. There are several algorithms to solve this problem. For example, it has been shown that it can be solved in  $O(n)$  time using the prune-and-search techniques for linear programming [20].

## 5 ADAPTIVE SINK MOBILITY

The optimal positioning of the sink, presented in the previous section, is specific only to a given snapshot of events that are present in the network. Moreover, in a real application the sink cannot usually move directly to the optimal position, it can only take a step towards it in a certain period of time. Therefore, to continuously optimize energy consumption in the case of dynamically evolving events, we should give efficient *strategies* for adaptive sink mobility. The specific application area we focus on is the so-called intrusion detection and tracking task.

To reduce the communication overhead, and thus prolong the network lifetime, we give in this section two algorithms to *relocate the sink node* from time to time in an energy efficient way. What we want to maximize is the network lifetime. To achieve this, we have two ways to proceed: (1) to minimize the *expected value* of the total energy spent in the next round, or (2) the *expected value* of the maximum energy load on a sensor. We do this by moving the sink node to the best possible location within reach. We assume that the sink makes a relocation decision (SRD) periodically, i.e., it calculates the optimal position where the energy consumption is minimal, and moves there (or at least towards it). The idea is that if we can predict the future location of the event, then we can select the optimal location of the sink node accordingly. The inputs for this decision are all the past observations the sensors reported (and, alternatively, the status of the sensor network if known by the sink, including sensor locations, network topology, routing protocol, and energy status of sensors). We assume that all calculations are performed by the sink node centrally. In order to find the best sink location, the full history of the intruder movement is used to forecast the intruder's future positions. The forecast can be given for the next location, but can also be extended to the consecutive steps.

### 5.1 Target Detection and Tracking

The goal of intrusion detection and tracking is to detect “intruders” (or targets) entering the observed area, to estimate their initial position and to *track* the position estimate as the target moves. To localize the target, the readings of a certain minimum number of nodes have to be combined. By considering the current position as well as past positions, the speed and the direction of the target can also be estimated.

Our chosen model here is basically an *intruder movement* model. (We should note, however, that the sink relocation strategies that we propose do not only apply to this particular intruder movement model. Our target tracking scenario only serves as an example to elaborate the proposed strategies.) We assume that “intruders” appear (uniformly and independently) on the boundary of the area described by a Poisson process of fixed rate  $\lambda$ , and start their own independent movement in the field. Let  $t_0$  be the “starting time” of the

intruder at  $Z_0 = Z(t_0) \in \partial C_{O,R}$ . In each unit of time it makes a “step” of fixed length  $l$ . What changes is the direction of the step. The direction of the first step, denoted by  $\theta$ , is uniformly chosen in  $[-\pi/2, \pi/2]$ . This is the main direction the intruder follows. Each further step has length  $l$  and the direction angle is chosen uniformly from  $[\theta - \sigma, \theta + \sigma]$ , where  $\sigma \in [-\pi, \pi]$  is the (only) free parameter of the model. This  $\sigma$  determines how closely the intruder follows the originally chosen direction  $\theta$ . It is clear that if  $\sigma = 0$ , then the movement follows a straight line, while  $\sigma = \pi$  is a random walk without any direction preference.

Let  $Z_k$  denote the event position at the  $k$ th step (at time  $t_k$ ). The evolution of the intruder’s trajectory is thus

$$Z_{k+1} = Z_k + l \underline{e}_{\theta_{k+1}}, \quad (14)$$

where  $\underline{e}_\varphi$  denotes the (unit) vector of  $(\cos \varphi, \sin \varphi)$ . The coordinates of  $Z_{k+1}$  are thus given by

$$x_{Z_{k+1}} = x_{Z_k} + l \cos \theta_{k+1}, \quad (15)$$

$$y_{Z_{k+1}} = y_{Z_k} + l \sin \theta_{k+1}. \quad (16)$$

The intruder leaves the network if  $Z_i \notin C_{O,R}$  for some  $i$ .

## 5.2 Minimizing the Total Energy (*Mintotal*)

To calculate the desirable future position of the sink node, let  $V_{total}(s)$  denote the expected value of the total energy spent in the next round,  $E_{total}^{SN}$  given by (6), i.e.,

$$V_{total}(s) = \mathbb{E} \left\{ E_{total}^{SN} \right\} \quad (17)$$

$$= \mathbb{E} \left\{ \sum_{i=1}^I E(Z_{k+1}^i, s) \right\} \quad (18)$$

$$= \sum_{i=1}^I \mathbb{E} \{ E(Z_{k+1}^i, s) \}. \quad (19)$$

where the energy function  $E(z, s)$  is given by (5), and the notation emphasizes that this energy depends on the event location  $z$  and sink location  $s$ . The expected value  $\mathbb{E} \{ E(Z_{k+1}, s) \}$  can be calculated as

$$\begin{aligned} \mathbb{E} \{ E(Z_{k+1}, s) \} &= \iint E(y, s) \, d\mathbb{P}\{Z_{k+1} = y | Z_k = z_k\} \\ &= \int_{\theta-\sigma}^{\theta+\sigma} E(z_k + l \underline{e}_\varphi, s) \frac{1}{2\sigma} \, d\varphi. \end{aligned} \quad (20)$$

The task now is to minimize  $V_{total}(s)$ , i.e., the optimal position  $s_{opt}$  for the sink node is

$$s_{opt} = \arg \min_s V_{total}(s). \quad (21)$$

The problem with (20) is that the parameters  $\theta$  and  $\sigma$  are not known. One way to proceed is to estimate  $\theta$  and  $\sigma$  somehow and calculate (20) numerically for each SRD to get an estimate for  $V_{total}(s)$ . However, this solution would be computationally rather demanding when solving (21). Instead, we try to solve

$$\hat{V}_{total}(s) = \sum_{i=1}^I E(\mathbb{E}\{Z_{k+1}^i\}, s) \rightarrow \min. \quad (22)$$

By comparing (17) and (22) we should note that, in general,

$$\mathbb{E}\{E(Z_{k+1}, s)\} \neq E(\mathbb{E}\{Z_{k+1}\}, s). \quad (23)$$

However, we assume that the error of the estimate is below a certain threshold and can be neglected in our case. The task now is to estimate  $\mathbb{E}\{Z_{k+1}\}$  somehow.

There are many sophisticated ways to predict the next position of the moving event based on our assumed correlated random movement model. Our solution is a simple one, since the event prediction is not the main scope of the paper. For a detailed description on how the estimates can be derived please refer to Appendix C. The case of multiple step forecast is also discussed there. However, to predict and use more than one step ahead in minimizing the energy requirement is not a straightforward task. Here we restrict ourselves to one step ahead forecast only. To predict the expected value of the next step of the event,  $\mathbb{E}\{Z_{k+1}\}$ , the forecast is based on standard statistical methods. As a result, we have

$$\mathbb{E}\{Z_{k+1}\} = Z_k + l \frac{\sin \sigma}{\sigma} e_{\theta}. \quad (24)$$

Equation (24) can be used for prediction if we replace  $\theta$  and  $\sigma$  by their estimates  $\hat{\theta}_{k+1}$  and  $\hat{\sigma}_{k+1}$ , and  $Z_k$  is substituted by the observed value  $z_k$ , i.e.,

$$\hat{Z}_{k+1} = \hat{\mathbb{E}}\{Z_{k+1}\} = z_k + l \frac{\sin \hat{\sigma}}{\hat{\sigma}} e_{\hat{\theta}_{k+1}}. \quad (25)$$

(Please refer to Appendix C for the estimates  $\hat{\theta}_{k+1}$  and  $\hat{\sigma}_{k+1}$  as well.)

By substituting (25) into (22) we have

$$\hat{V}_{total}(s) = \sum_{i=1}^I E(\hat{Z}_{k+1}^i, s) \rightarrow \min. \quad (26)$$

As argued before (see (12)), this minimization is equivalent with minimizing the sum of event distances, i.e., the best position for the sink node in the next

round ( $S_{k+1}$ )—assuming that the sink can only move at limited speed—is given by

$$S_{k+1}^{total} = \arg \min_{s \in \mathcal{C}_{S_k, r_v}} \sum_{i=1}^I \max(h/2, d(\hat{Z}_{k+1}^i, s)), \quad (27)$$

where  $r_v$  is the maximum distance the sink node can move within one round, i.e., the future position of the sink can only be within the circle  $\mathcal{C}_{S_k, r_v}$  of radius  $r_v$ , with its center being the sink's present position. We call this strategy as *mintotal* in the following.

### 5.3 Minimizing the Maximum Energy (*Minmax*)

As mentioned before, it can happen that by using the *mintotal* strategy the energy consumptions of the sensors become highly uneven. To avoid this, we try to minimize the energy consumption for the most heavily loaded sensor. Let  $V_{\max}(s)$  denote the expected value of the maximal energy spent in the next round,  $E_{\max}^{SN}$  given by (10), i.e.,

$$V_{\max}(s) = \mathbb{E} \left\{ E_{\max}^{SN} \right\} \quad (28)$$

$$= \mathbb{E} \left\{ \max_{1 \leq i \leq I} E(Z_{k+1}^i, s) \right\}, \quad (29)$$

where the energy function  $E(z, s)$  is given by (9). Unfortunately, in general

$$\mathbb{E} \left\{ \max_{1 \leq i \leq I} E_{\max}^i \right\} \neq \max_{1 \leq i \leq I} \mathbb{E} \{ E(Z_{k+1}^i, s) \}. \quad (30)$$

However, we assume that the estimate

$$\hat{V}_{\max}(s) = \max_{1 \leq i \leq I} E(\mathbb{E}\{Z_{k+1}^i\}, s) \quad (31)$$

is a good approximation of the expected maximum energy requirement in the network. Thus, we have to minimize  $\hat{V}_{\max}(s)$ . To do this, we need the estimate of  $\mathbb{E}\{Z_{k+1}\}$  that we previously derived (refer to Appendix C), see (25).

By substituting (25) into (31) we have

$$\hat{V}_{\max}(s) = \max_{1 \leq i \leq I} E(\hat{Z}_{k+1}^i, s) \rightarrow \min. \quad (32)$$

Since this minimization is equivalent with the task of minimizing the maximum event distance from the sink (see (13)), we have

$$S_{k+1}^{\max} = \arg \min_{s \in \mathcal{C}_{S_k, r_v}} \left\{ \max_{1 \leq i \leq I} d(\hat{Z}_{k+1}^i, s) \right\}. \quad (33)$$

In the following, we call this strategy as *minmax*.

## 6 IMPLEMENTATION ISSUES

### 6.1 Routing

In our analytical investigations we assumed that the multi-hop communication is supported by an ideal short path routing mechanism; the sensors were considered to be deployed densely and evenly enough to find a straight linear path towards the sink. However, in a real world scenario there are many factors that make such an ideal routing impossible; sensors are not that densely deployed, the depletion of some sensors after a while may result in “black holes” in the area, etc. Therefore, in order to implement our proposed adaptive mobility strategies, a more realistic routing mechanism has to be used, that takes into account all these factors.

We considered a distributed routing solution, where there is no central authority to select the end-to-end route and inform the participating nodes about it. It is up to the nodes to decide locally to whom the packet should be handed over. The question is, how to choose the next hop among the neighbors within radio range. We applied the GOAFR routing algorithm [17] that is a variant of the original GFG algorithm [3]. The GOAFR routing combines the *greedy* and the *face* algorithms. The *greedy* algorithm always picks the neighbor closest to the sink to be next node for routing. However, in certain situations it can occur that no neighbor is closer to the sink than the current node, for example if there is a “hole” in the sensor field, as shown in Figure 5. In this case the routing switches to the *face* algorithm, and passes around the hole on its border. When it is possible, the routing switches back to the *greedy* algorithm.

### 6.2 Updates Related to Sink Relocation

If sensors want to send data to the sink, they have to know its position, if geographical greedy routing is used. Moving the sink node has a negative side-effect on the energy consumption: sensors should be alerted about the changed position of the sink through location update messages. Therefore,

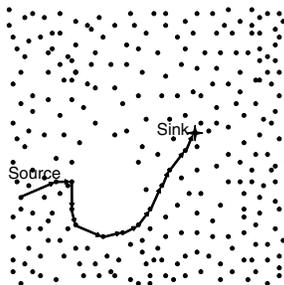


FIGURE 5  
The route between the sink and the reporting node.

having an efficient and power saving update mechanism is essential for a viable data gathering strategy.

In approaches where the sink node is static, no update messages are needed for that; information about the location of the sink is either hard-coded in the sensors or broadcasted at the start-up phase of the network. However, periodic flooding phases might also be used in static approaches. As an example, in Directed Diffusion [10] the sink periodically floods the network with query messages; sensors answer them along the reverse paths they received those queries on. This flooding mechanism is similar to a location update performed by a mobile sink, as the goal of both approaches is to refresh the data gathering paths.

Our solution assumes a periodical update message sent out by the mobile sink. However, there are several factors that make this mechanism “power-friendly”. First, sensors do not relay update messages among them; the sink node has the ability to cover the entire region through a single broadcast message, updating each sensor directly. Note that the sink does not have power limitations; thus, it can afford such a “costly” update mechanism. Moreover, if needed, dedicated powered relay nodes can be deployed in the region to forward these update messages.

Besides eliminating multi-hop update relaying, an important feature is that only sensors that sensed an event listen to the update messages. This can be done, for example, in the following way. Time is divided into sensing periods and communication periods. During a sensing period sensors are in a semi-sleep mode; they operate only as sensing units, but the communication tasks are disabled. If during this period a sensor observes something it wants to report to the sink, it prepares the data and waits for the communication period. When a reporting sensor receives the update, it sends its data towards the advertised location, incorporating the sink coordinates in its message as well. Hence, intermediate relaying nodes do not have to listen to the periodic updates of the sink.

It is important, that all the data packets that are sent by a sensor have to arrive to the sink before it moves away. This can be ensured if the sink stays in its advertised position for a certain (guaranteed) period of time, also included in the update message. We assume that data delivery is fast enough to fit safely into this guaranteed time period. After the communication period ends, a new sensing period begins. Taking into consideration all the reports, the sink makes a prediction about the future evolution of the events and calculates the optimal neighboring position it should move to. Then, while the other nodes are in their sensing period, the sink moves to the calculated position. In the same time, the events propagate as well, according to the considered models (e.g., the intruder continues to penetrate into the region).

Note also that location update messages might constitute a threat to the security of the system, as a malicious node could advertise itself as being the mobile sink, attracting all the traffic towards it. There are several well

established solutions to deal with such attacks, even if some of them might need to be adapted to the specificities of sensor networks. However, security issues are out of the scope of this paper.

### 6.3 Communicating Neighbors

In our model we assumed that only a specific receiver node, chosen by the routing mechanism, has to listen to the transmission of the sender; none of the other neighbors within radio range have to waste energy on overhearing, i.e., on receiving packets that are not meant for them. This can be achieved, for example, by using the free sleep-schedule feature of the S-MAC protocol [36]. Using S-MAC, an idle sensor goes to sleep for some time; then it wakes up, and listens to see if any other node wants to talk to it. All nodes are free to choose their own listen/sleep schedules. These schedules are exchanged by broadcasting them to all immediate neighbors. This ensures that all neighboring nodes can talk to each other even if they have different schedules. For example, if node A wants to talk to node B, it just waits until B is listening. Hopefully, the other neighbors will be in sleep mode that time; thus, they do not waste energy for listening to the transmission of node A. If more nodes are in listening mode, the use of short RTS/CTS (Request-To-Send/Clear-To-Send) packets can help to further reduce overhearing (see [36] for more details).

## 7 SIMULATION RESULTS

We simulated the proposed sink relocation strategies using MATLAB. We assumed that the covered region is a circular area of radius  $R = 1000$  m, in which we randomly distributed 10,000 sensors, using a uniform distribution model. Both the sensing range and the maximum communication range of each sensor were fixed to 80 m. At the beginning of a simulation run each sensor was loaded with 1000 “units” of energy. The cost of receiving a packet was 1 unit. The cost of sending one packet depended on the transmission distance  $d(E_{Tx} \sim d^\alpha, \alpha = 3)$ ; the transmission consumed 1 unit of energy for  $d = h = 80$  m.

The events are reported to the sink through multi-hop routing. Figure 6 shows a snapshot of the network with five simultaneous events being reported. Around each event we marked the circular area containing the sensors that observed that event. All those sensors start to send data to the sink on multi-hop wireless paths that might overlap near to the sink node. Nodes on such an overlapping segment will have an increased load as they have to relay data from several sensors.

### 7.1 Event Modeling

Events occurred at uniformly chosen random locations on the periphery of the area. The probability of a new event occurring in a simulation round was

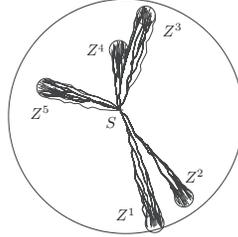


FIGURE 6  
Routes between the sink and the reporting nodes.

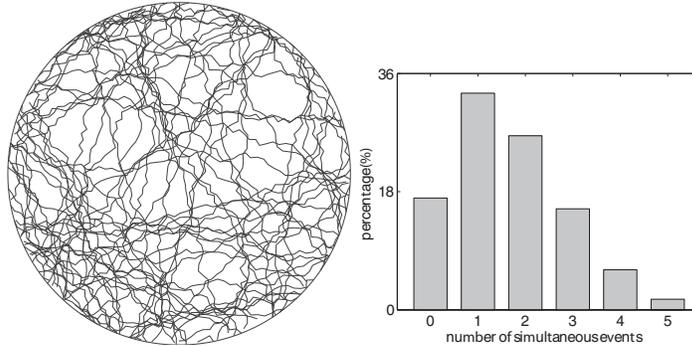


FIGURE 7  
Trajectories of 100 events (left) and the number of simultaneous events in the simulations (right).

0.03. The direction  $\theta$  of the first step taken by an event was uniformly chosen between 0 and  $\pi$ . For each further step the direction angle was uniformly chosen between  $[\theta - \pi/4, \theta + \pi/4]$ . Events moved with a speed of 40m/round.

In Figure 7 (left) we present the trajectories of 100 events that entered the region. We can observe that the different parts of the region were affected by these events in a homogeneous manner. Note that these events were not simultaneous ones, but appeared and disappeared according to our simulation model. In Figure 7 (right) we show a histogram on the frequency of simultaneous events during an entire simulation run. One can see that in most of the time there are two or more events present in parallel.

## 7.2 Sink Relocation Strategies

It is quite straightforward to assert that by adaptively moving the sink we can increase the energy efficiency of the network. As said before, the sink is relocated periodically in each round using either the *mintotal* or the *min-max* strategy proposed in Section 5. Once all messages are received from the reporting sensors, the positions of the sensed events are predicted: the next position of the sink can be determined with arbitrary precision by evaluating (27) and (33), respectively.

However, we should be able to quantify the performance increase compared to other solutions. In order to do so, we consider three other approaches. The first one, called *fix*, assumes the sink to be static, and located in the center of the covered area. The second one, called *circular*, proposes to move the sink along the periphery of the network, with a constant speed, independently of the occurred events. We consider this approach as authors in [19] argue in favor of it as being the optimal solution for lifetime elongation. However, they analyzed a time-driven scenario, as opposed to our event-driven model. Finally, the third approach, called *rwp* considered the sink to follow a random waypoint mobility model, again independently of the current events. The sink randomly chooses a point in the area, and goes towards it with a constant speed of 40 m/round; upon reaching it, it chooses a new direction.

### 7.3 Network Lifetime

Figure 8 presents the average lifetime of the network for the different strategies. We ran the simulation 10 times, and considered the network to be alive until one of the events was unobserved by the sink, i.e., either there were no sensors to detect it, or the information could not be relayed to the sink. We can see that the *mintotal* strategy outperforms all the other solutions, ensuring 16% longer lifetime than the *circular* strategy, and nearly 150% longer than the *fix* case. In the figure we presented the 95% confidence intervals as well.

### 7.4 Energy Consumption

To explore the reasons behind the lifetime elongation generated by the proposed strategies, the energy consumption of the network needs to be analyzed in more details. Figure 9 presents the total energy consumption in the network for the five different strategies. The results were obtained for one specific simulation run, for the same succession of events in the five cases. On the  $x$ -axis we present the number of rounds completed in the simulation. A curve ends in the figure if the network died for that specific strategy. We can see that by

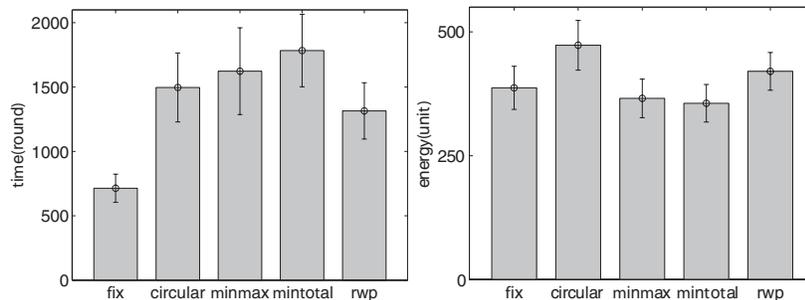


FIGURE 8  
Average lifetime of the network (left), and the average energy consumption (right) in a round for all five strategies.

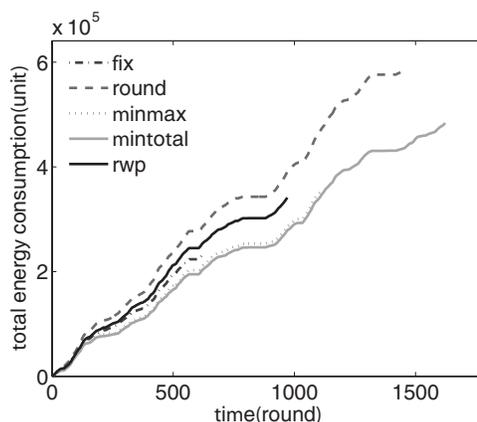


FIGURE 9  
Total energy consumption of the five strategies.

positioning the sink in the middle of the network we consume less energy in overall than by moving the sink along the periphery. However, nodes around the sink deplete their batteries rapidly, and the network dies. On the other hand, we can see that the *circular* strategy consumes significantly more energy than both of our proposals. Note also that there is practically no difference between our two solutions in terms of overall energy consumption. This is mainly due to the fact that in the majority of the cases there are few (0 to 2) simultaneous events in the region; thus, even if the sink moves to different locations (as shown in Figure 12), the overall power consumption will not differ significantly. Finally, moving the sink randomly inside the area consumes less energy than moving it on the periphery, but it is still less efficient than the adaptive strategies. By repeating the simulation several times, for different successions of events, we obtained similar shapes for all the curves, the only difference being in when the network dies in the different cases, a parameter that greatly depends on the occurred events.

In Figure 8 (right) we see the average energy consumption of the entire network *per round*, for the five different strategies. (We ran 10 times the simulation, and calculated a cumulated average.) It can be seen that our two adaptive strategies consume the less energy; they are around 30% better in average than the *circular* strategy, and around 15% better the case of the randomly moving sink.

It is also interesting to see how the five strategies affect the sensors located at different portions of the covered area. Figure 10 shows the average energy consumption of a sensor in a single round in function of its distance to the center of the area. We divided the total energy consumption of a sensor with the number of rounds it was alive, and calculated the average of this value for all the sensors located at the same distance from the center. The final averages

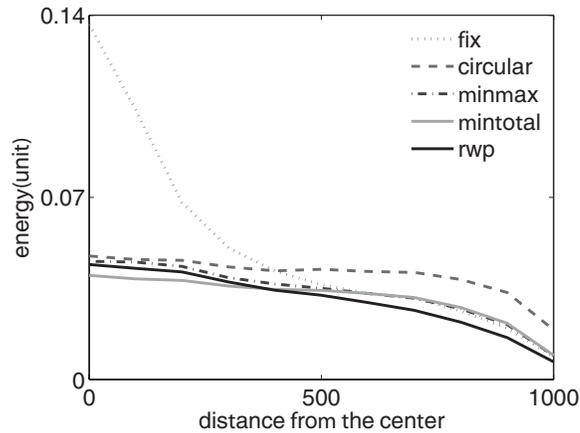


FIGURE 10  
The average energy consumption of the sensors as a function of the sensors' distance from the center of the sensor field.

were obtained after running the simulation 10 times. An obvious result is that the *fix* strategy depletes aggressively the sensors located close to the center. It can also be seen that the *circular* strategy ensures the most homogeneous energy consumption of the sensors. However, our two strategies consume less energy than the *circular* solution in all the areas of the network, which explains the resulting lifetime elongation. Sensors near the periphery consume less energy for all the strategies, as they are rarely selected to relay messages of other nodes.

Finally, in Figure 11 we present the remaining energy of the sensors in different areas of the covered region, after the network has died; an area is completely black if the corresponding sensors have 100% of their initial energy still available. The values are averages calculated over 10 simulations. It can be observed again that the *fix* strategy depletes the sensors around the center, while the *circular* one makes use of nearly all the sensors in a comparable way. Our two solutions conserve more energy in the network than the *circular*

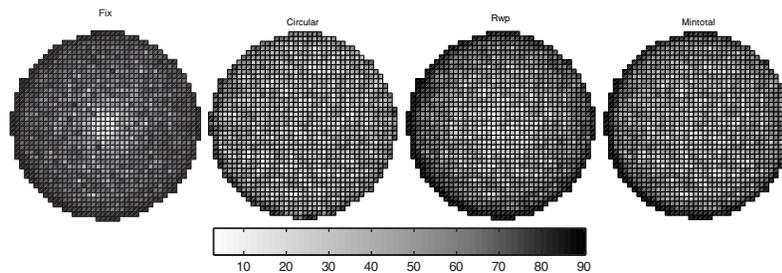


FIGURE 11  
The remaining energy in the network.

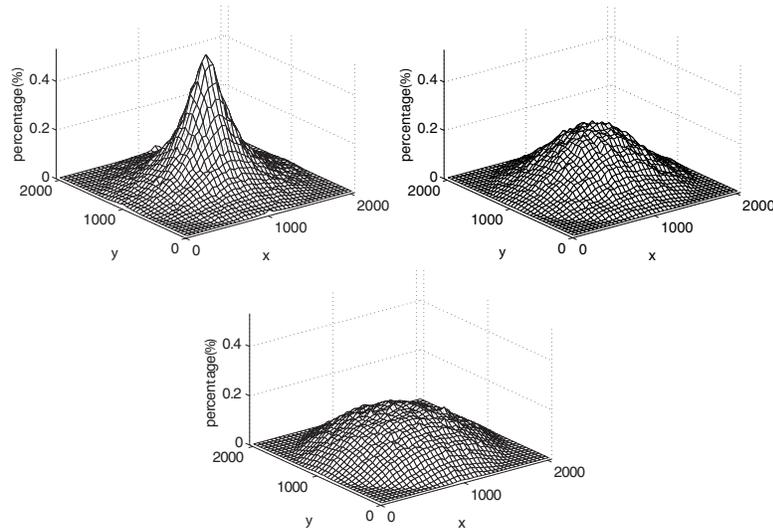


FIGURE 12 Histogram of sink positions for the *minmax* (upper left), *mintotal* (upper right) and *rwp* (bottom) strategies.

strategy, while still ensuring a longer network lifetime. As the results for the *minmax* and the *mintotal* cases were quite similar, we have chosen to show only one of them.

### 7.5 Sink Location Distribution

Figure 12 shows the distribution of the sink coordinates after 30.000 simulation rounds, for the *minmax* (upper left) and the *mintotal* (upper right) mobility strategies, respectively. We can see that in the *minmax* case the sink often resides near the center of the region, while in the *mintotal* case the distribution is more homogeneous. This is because in the *mintotal* strategy if there are several events nearby in a specific area, they will attract the sink even if there is another distant event, isolated in an opposite area, which will be negatively affected. On the other hand, in the *minmax* approach the isolated distant event is more strongly protected. The distribution of the sink coordinates in the *rwp* case is shown in Figure 12 (bottom). We can see that the *rwp* strategy ensures an even more homogeneous distribution than the *mintotal* case.

## 8 CONCLUSION

Enhancing energy-efficiency is primordial in a wireless sensor network. There are several techniques to achieve that, e.g., by using energy-aware routing protocols, topology control schemes, or clustering mechanisms. Many recent

papers propose to use mobile sinks to reduce energy consumption. However, they usually assume a time-driven scenario, and are frequently based on single-hop communication between the sensors and the mobile sink.

In this paper we proposed an adaptive mobility solution that is specific to event-driven applications and builds on multi-hop data relaying among sensors. We presented the analytical foundations of two sink relocation strategies: one optimizes the overall energy consumption in the network (*mintotal*), the other minimizes the energy consumption of the most loaded sensor (*minmax*). We showed through simulations that both strategies ensure a network lifetime of around 150% longer than in case of a fixed sink, and consume about 30% less energy than the *circular* strategy that moves the sink along the periphery of the network. Even if this *circular* strategy ensures a more homogeneous depletion of the sensors, the network dies more rapidly due to the increased overall energy consumption. Our adaptive strategies perform significantly better than the case of a randomly moving sink as well, both regarding network lifetime and energy consumption.

## REFERENCES

- [1] C. M. Bergman, J. A. Schaefer, and S. N. Luttich. (2000). Caribou movement as a correlated random walk. *Oecologia*, 123:364–374.
- [2] Andrej Bogdanov, Elitza Maneva, and Samantha Riesenfeld. (March 2004). Power-aware base station positioning for sensor networks. In *Proc., IEEE INFOCOM 2004*, Hong Kong.
- [3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. (2001). Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616.
- [4] P. Bovet and S. Benhamou. (1988). Spatial analysis of animals' movements using a correlated random walk model. *Journal of Theoretical Biology*, 131:419–433.
- [5] Zack Butler and Daniela Rus. (Oct.–Nov. 2003). Event-based motion control for mobile-sensor networks. *IEEE Pervasive Computing*, 2(4):34–42.
- [6] A. Cerpa and D. Estrin. (June 2002). ASCENT: Adaptive self-configuring sensor networks topologies. In *Proc., IEEE INFOCOM 2002*, volume 3, pages 1278–1287, New York, NY, USA.
- [7] A. Chakrabarti, A. Sabharwal, and B. Aazhang. (April 2003). Using predictable observer mobility for power efficient design of sensor networks. In *Proc., 2nd Int. Workshop on Information Processing in Sensor Networks (IPSN)*, pages 129–145, Palo Alto, CA, USA. Also in *Lecture Notes in Computer Science*, Vol. & (NO) (2634), pp. 129–145.
- [8] Jae-Hwan Chang and Leandros Tassiulas. (March 2000). Energy conserving routing in wireless ad-hoc networks. In *Proc., IEEE INFOCOM 2000*, volume 1, pages 22–31, Tel Aviv, Israel.
- [9] Shashidhar Rao Gandham, Milind Dawande, Ravi Prakash, and S. Venkatesan. (December 2003). Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Proc., IEEE GLOBECOM 2003*, volume 22, pages 377–381, San Francisco, CA, USA.
- [10] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proc., ACM MOBICOM 2000*, pages 56–67, Boston, MA, USA.
- [11] Sushant Jain, Rahul C Shah, Gaetano Borriello, Waylon Brunette, and Sumit Roy. (March 2004). Exploiting mobility for energy efficient data collection in sensor networks. In *Proc.*,

- 2nd IEEE/ACM Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, Cambridge, UK.
- [12] A. Kansal, A. Somasundara, D. Jea, M. B. Srivastava, and D. Estrin. (June 2004). Intelligent fluid infrastructure for embedded networks. In *Proc., ACM MOBISYS 2004*, pages 111–124, Boston, MA, USA.
  - [13] Aman Kansal, Mohammad Rahimi, William J Kaiser, Mani B Srivastava, Gregory J Pottie, and Deborah Estrin. (October 2004). Controlled mobility for sustainable wireless networks. In *Proc., IEEE Sensor and Ad Hoc Communications and Networks (SECON)*, Santa Clara, CA.
  - [14] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas. (March 2003). Routing for network capacity maximization in energy-constrained ad-hoc networks. In *Proc., IEEE INFOCOM 2003*, pages 673–681, San Francisco, CA, USA.
  - [15] V. Kawadia and P. Kumar. (April 2003). Power control and clustering in ad hoc networks. In *Proc. IEEE INFOCOM 2003*, pages 459–469, San Francisco, CA, USA.
  - [16] Hyung Seok Kim and Tarek F. Abdelzaher. (November 2003). Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *Proc., ACM SENSYS*, Los Angeles, CA.
  - [17] Fabian Kuhn, Roger Wattenhofer, and Samantha Aaron Zollinger. (2003). Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proc., Proceedings of the 4th ACM international symposium on Mobile ad hoc networking computing, 2003*, pages 267–278, Annapolis, Maryland, USA.
  - [18] N. Li and J. C. Hou. (March 2004). Topology control in heterogeneous wireless networks: Problems and solutions. In *Proc., IEEE INFOCOM 2004*, Hong Kong.
  - [19] Jun Luo and Jean-Pierre Hubaux. (March 2005). Joint mobility and routing for lifetime elongation in wireless sensor networks. In *Proc., IEEE INFOCOM 2005*, Miami, FL, USA.
  - [20] N. Megiddo. (1983). The weighted Euclidean 1-center problem. *Mathematics of Operations Research*, 8(4):498–504.
  - [21] S. Olariu, I. Stojmenovic, and J. Urrutia. (April 2006). Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting. In *Proc., IEEE INFOCOM 2006*, Barcelona, Spain.
  - [22] J. Pan, L. Cai, T. Hou, Y. Shi, and X. Shen. (2005). Optimal base-station locations in two-tiered wireless sensor networks. to appear in *IEEE Transactions on Mobile Computing*.
  - [23] Jianping Pan, Y. Thomas Hou, Yi Shi Lin Cai, and Sherman X. Shen. (September 2003). Topology control for wireless sensor networks. In *Proc., 9th Annual International Conference on Mobile Computing and Networking*, pages 286–299, San Diego, CA, USA.
  - [24] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. (March 2002). Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50.
  - [25] Sushant Jain Rahul C. Shah, Sumit Roy and Waylon Brunette. (May 2003). Data MULEs: Modeling a three-tier architecture for sparse sensor networks. In *Proc., IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, pages 30–41, Anchorage, Alaska, USA.
  - [26] R. Ramanathan and R. Rosales-Hain. (March 2000). Topology control of multihop wireless networks using transmit power adjustment. In *Proc., IEEE INFOCOM 2000*, pages 404–413, Tel Aviv, Israel.
  - [27] T. S. Rappaport. (2002). *Wireless Communications: Principle and Practice*. Prentice Hall.
  - [28] A. Sankar and Z. Liu. (March 2004). Maximum lifetime routing in wireless ad-hoc networks. In *Proc., IEEE INFOCOM 2004*, Hong Kong.
  - [29] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie. (October 2000). Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27.
  - [30] Lang Tong, Qing Zhao, and Srihari Adireddy. (October 2003). Sensor networks with mobile agents. In *Proc., IEEE MILCOM 2003*, volume 22, pages 688–693, Boston, MA, USA.

- [31] Z. Vincze, D. Vass, R. Vida, and A. Vidács. (2006). Adaptive sink mobility in event-driven clustered single-hop wireless sensor networks. In *Proc., 6th Int. Network Conference (INC 2006)*, pages 315–322, Nice, France.
- [32] Z. Vincze, D. Vass, R. Vida, A. Vidács, and A. Telcs. (2006). Sink mobility in event-driven multi-hop wireless sensor networks. In *Proc., 1st Int. Conf. on Integrated Internet Ad hoc and Sensor Networks (InterSense)*, page Article No. 13, Nice, France.
- [33] G. Wang, G. Cao, and T.L. Porta. (2004). Movement-assisted sensor deployment. In *Proc., IEEE INFOCOM 2004*, Hong Kong.
- [34] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. (January 2005). Exploiting sink mobility for maximizing sensor networks lifetime. In *Proc., 38th Hawaii International Conference on System Sciences*, Big Island, Hawaii.
- [35] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang. (April 2001). Distributed topology control for wireless multihop ad hoc networks. In *Proc., IEEE INFOCOM 2001*, pages 1388–1397, Anchorage, Alaska, USA.
- [36] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. (September 2002). A twotier data dissemination model for large-scale wireless sensor networks. In *Proc., ACM MOBICOM 2002*, pages 148–159, Atlanta, GA, USA.
- [37] O. Younis and S. Fahmy. (March 2004). Distributed clustering in ad-hoc sensor networks: A hybrid, energy efficient solution. In *Proc., IEEE INFOCOM 2004*, Hong Kong.
- [38] W. Zhao and M. Ammar. (May 2003). Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks. In *Proc. of the 9th IEEE Workshop on Future Trends in Distributed Computed Systems (FTDCS 2003)*, pages 308–314, San Juan, Puerto Rico.
- [39] Gil Zussman and Adrian Segall. (April 2003). Energy efficient routing in ad hoc disaster recovery networks. In *Proc., IEEE INFOCOM 2003*, pages 405–421, San Francisco, CA, USA.



## APPENDIX

### A AVERAGE HOP COUNT

To calculate the average hop count, first let us derive the average distance of a sensing node from the sink. The average distance  $\bar{l}$  from sensors within  $\mathcal{C}_{Z,r_0}$  can be calculated (see Figure 13) as

$$\bar{l} = \frac{1}{|\mathcal{C}_{Z,r_0}|} \iint_{\mathcal{C}_{Z,r_0}} l \, dA, \quad (34)$$

that is

$$\bar{l} = \frac{1}{r_0^2 \pi} \int_{-r_0}^{r_0} \int_{-\sqrt{r_0^2-y^2}}^{\sqrt{r_0^2-y^2}} \sqrt{(x-d)^2 + y^2} \, dx \, dy. \quad (35)$$

where—without loss of generality—we assumed that the event is at location  $(0, 0)$  and the sink is located at  $(d, 0)$ . The integral (35) can be calculated numerically for different event distances  $d$  (see the solid line on Figure 14). The results show that—apart from small event distances—the average distance  $\bar{l}$  can be very well approximated by the event distance  $d$ .

The hop count  $k$  for each message can be calculated by dividing the distance  $l$  by the hop length  $h$ , and rounding the result up to the nearest integer, i.e.,

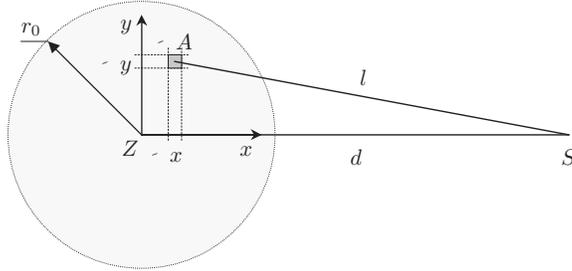


FIGURE 13  
Distance between the sink and a sensing node at a particular location. ( $\Delta A = (\Delta x \times \Delta y)$ ).

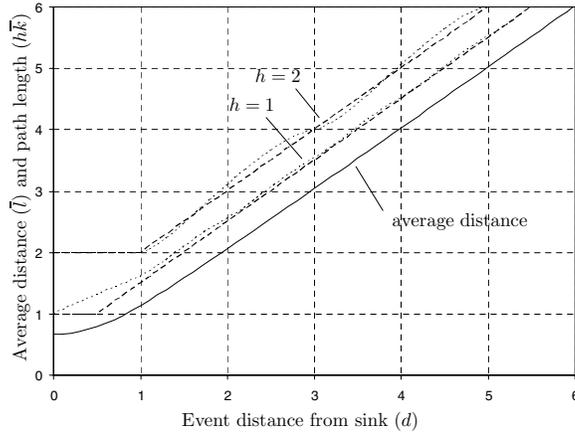


FIGURE 14  
Average distance  $\bar{l}$  (solid line), average path length ( $h\bar{k}$ , dotted line) and its approximation (dashed line) from a sensing node to the sink. (Parameters:  $r_0 = 1$ ,  $h = 1, 2$ ).

$k = \lceil l/h \rceil$ . Knowing only the average distance  $\bar{l}$ , a rough estimate for the average hop count  $\bar{k}$  could be  $\bar{l}/h$ ; however, this would underestimate the desired quantity. This is because the estimate does not take into account that, for a given path,  $\lceil l/h \rceil \approx l/h + 0.5$  “on the average”. Another candidate would be estimating  $\bar{k}$  by  $\lceil \bar{l}/h \rceil$ ; however, this can seriously overestimate the average hop count. In our model, the average number of hops is approximated as

$$\bar{k} \approx \max(1, d/h + 0.5), \quad (36)$$

where we also took into account that the sensing range  $r_0$  is smaller than the hop length  $h$ . This result is validated (see the dotted lines on Figure 14) by approximating each path length by  $h \lceil l/h \rceil$  and plotting its average for each  $d$ . (Here we neglected the fact that the last hop of the path can be shorter than  $h$ .)

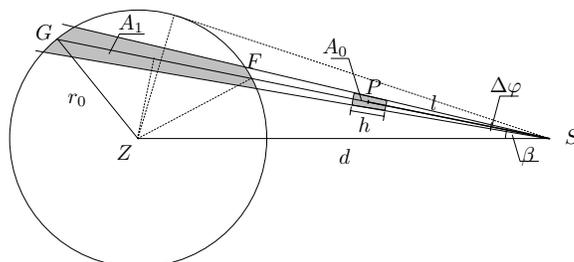


FIGURE 15  
Approximating transit load  $L$ .

## B TRANSIT LOAD

To calculate the transit load on a given node, we proceed as follows. As Figure 15 shows, the average *transit* load  $L_{(l,\beta)}$  on a particular node  $P$  at polar coordinates  $(l, \beta)$  where  $l = \overline{PS}$  and  $\beta = \angle ZSP$ , is approximated as

$$L_{(l,\beta)} \approx A_1/A_0, \quad (37)$$

where the area  $A_0$  can be calculated as

$$A_0 = \pi((l + h/2)^2 - (l - h/2)^2) \frac{\Delta\varphi}{2\pi} = hl\Delta\varphi, \quad (38)$$

and  $A_1$  is given by

$$A_1 \approx \Delta\varphi(g^2 - f^2)/2 \quad (39)$$

$$= 2d \cos \beta \sqrt{r_0^2 - d^2 \sin^2 \beta} \Delta\varphi, \quad (40)$$

where  $g = \overline{GS}$  and  $f = \overline{FS}$  are

$$g = d \cos \beta + \sqrt{r_0^2 - d^2 \sin^2 \beta}, \quad (41)$$

$$f = d \cos \beta - \sqrt{r_0^2 - d^2 \sin^2 \beta}. \quad (42)$$

When the area  $A_0$  overlaps with  $A_1$ , i.e., if  $l > f - h/2$ , the area  $A'_1$  whose generated traffic must be forwarded by nodes in  $A_0$  can be calculated as follows:

$$A'_1 \approx \Delta\varphi(g^2 - (l + h/2)^2)/2. \quad (43)$$

Thus, the transit load is given by

$$L_{(l,\beta)} = \begin{cases} \frac{g^2 - f^2}{h^2}, & \text{if } l \leq \frac{h}{2} \\ \frac{g^2 - f^2}{2lh}, & \text{if } \frac{h}{2} < l < f - \frac{h}{2} \\ \frac{g^2 - (l + h/2)^2}{2lh}, & \text{if } f - \frac{h}{2} \leq l < g - \frac{h}{2}. \end{cases} \quad (44)$$

The transit load  $L_{(l,\beta)}$  is plotted on Figure 16 as a function of path length  $l$  for different values of  $\beta$ . As the figure shows, the load is maximal when  $\beta$  is zero (i.e., on the straight path between the event and the sink node), and increases hyperbolically when approaching the sink.

We should note, that (44) does not take into account the discrete hops on the transmission path, but it assumes a continuous flow of traffic. The load distribution is not so “smooth” when hops of length  $h$  are considered. A dotted line is also drawn on Figure 16 to show the real multi-hop case when the network is still assumed to be dense and routing is ideal. In this case the transit load was approximated numerically. The result is in good agreement with our proposed model, apart from the discrete steps appearing at larger distances.

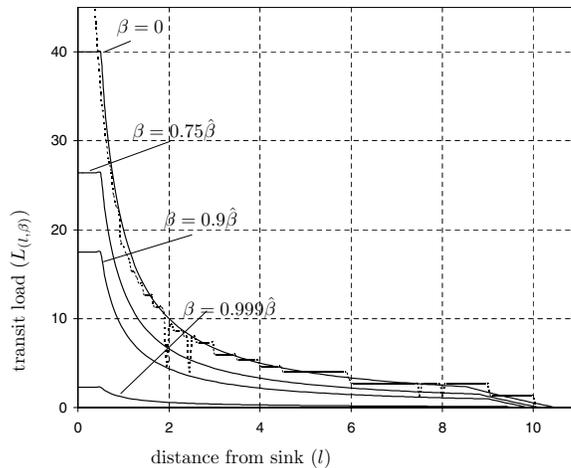


FIGURE 16  
Transit load as a function of path length  $l$  and direction  $\beta$ . (The event distance from the sink is  $d = 10r_0$ , and  $\hat{\beta} = \arcsin(r_0/d)$ ).

## C EVENT LOCATION PREDICTION

### C.1 One Step Forecast

Let us assume that  $k$  steps are already observed, i.e., the values  $\theta_i = \vartheta_i$  for  $i = 1, \dots, k$  are known. Next, two cases can be distinguished: either whether the sector angle  $\sigma$  is known, or not. In the former case, the best estimator for the unknown  $\theta$  is the sample mean, i.e.,

$$\hat{\theta}_{k+1} = \bar{\vartheta}_k = \frac{1}{k} \sum_{i=1}^k \vartheta_i. \quad (45)$$

It is also known that as  $k$  increases this estimate self-improves and  $\bar{\vartheta}_k$  is approximately normally distributed with mean  $\theta$  and dispersion  $\sigma/\sqrt{3k}$ .

If the sector angle  $\sigma$  is not known, one can use the sample mean and dispersion to estimate  $\theta$  and  $\sigma$ , but there is a commonly used alternative. In this case we can consider  $\eta = \theta - \sigma$  and  $\xi = \theta + \sigma$ , and the fact that the  $\theta_i$ -s are independent and uniformly distributed in the interval  $[\eta, \xi]$ . It is standard to estimate  $\eta$  by

$$\hat{\eta}_k = \min\{\vartheta_i : i = 1, 2, \dots, k\}, \quad (46)$$

and  $\xi$  by

$$\hat{\xi}_k = \max\{\vartheta_i : i = 1, 2, \dots, k\}. \quad (47)$$

These estimators are asymptotically unbiased estimators of  $\eta$  and  $\xi$ . Based on these, the estimate for the direction of the next step is

$$\hat{\theta}_{k+1} = (\hat{\xi}_k + \hat{\eta}_k)/2, \quad (48)$$

and an estimate of  $\sigma$  is given by

$$\hat{\sigma}_{k+1} = (\hat{\xi}_k - \hat{\eta}_k)/2. \quad (49)$$

To estimate the expected value of the coordinates of  $Z_{k+1}$  we can write

$$\mathbb{E}\{x_{Z_{k+1}}\} = x_{Z_k} + \frac{l}{2\sigma} \int_{\theta-\sigma}^{\theta+\sigma} \cos \varphi \, d\varphi \quad (50)$$

$$= x_{Z_k} + \frac{l}{2\sigma} (\sin(\theta + \sigma) - \sin(\theta - \sigma)) \quad (51)$$

$$= x_{Z_k} + l \frac{\sin \sigma}{\sigma} \cos \theta. \quad (52)$$

Similarly, we get

$$\mathbb{E}\{y_{Z_{k+1}}\} = y_{Z_k} + l \frac{\sin \sigma}{\sigma} \sin \theta. \quad (53)$$

Thus, we have

$$\mathbb{E}\{Z_{k+1}\} = Z_k + l \frac{\sin \sigma}{\sigma} \underline{e}_\theta. \quad (54)$$

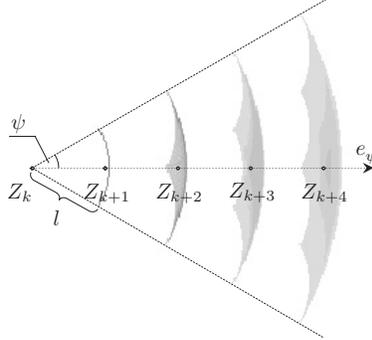


FIGURE 17  
Multiple step forecast.

Equation (54) can be used for prediction if we replace  $\theta$  and  $\sigma$  by their estimates  $\hat{\theta}_{k+1}$  and  $\hat{\sigma}_{k+1}$ , and  $Z_k$  is substituted by the observed value  $z_k$ , i.e.,

$$\hat{Z}_{k+1} = \hat{\mathbb{E}}\{Z_{k+1}\} = z_k + l \frac{\sin \hat{\sigma}}{\hat{\sigma}} e_{\hat{\theta}_{k+1}}. \quad (55)$$

### C.2 Multiple Step Forecast

Until now we have dealt with the forecast of one step in advance, but the forecast for more than one step in advance can also be developed. The random variable we consider is

$$Z_{k+i} = Z_k + l e_{\theta_{k+1}} + l e_{\theta_{k+2}} + \cdots + l e_{\theta_{k+i}}. \quad (56)$$

Since the  $\theta_i$  variables are modeled as independent, identically distributed random variables, we can use the estimate (55) recursively, and finally we have (see Figure 17)

$$\hat{Z}_{k+i} = z_k + il \frac{\sin \sigma}{\sigma} e_{\hat{\theta}_{k+1}}. \quad (57)$$