

Scalable Data Collection Protocols for Wireless Sensor Networks with Multiple Mobile Sinks *

Athanasios Kinalis and Sotiris Nikolettseas

Computer Technology Institute (CTI),

N. Kazantzaki Str., 26500 Patras, Greece.

Tel: +30 2610 960324, Fax: +30 2610 960442.

Department of Computer Engineering and Informatics,

University of Patras, 26500 Patras, Greece. Tel: +30 2610 960459

nikole@cti.gr, kinalis@ceid.upatras.gr

Abstract

Data propagation in wireless sensor networks is usually performed as a multihop process. To deliver a single message, the resources of many sensor nodes are used and a lot of energy is spent. Recently, a novel approach is catching momentum because of important applications; that of having a mobile sink move inside the network area and collect the data with low energy cost. Here we extend this line of research by proposing and evaluating three new protocols.

Our protocols are novel in a) investigating the impact of having many mobile sinks b) in weak models with restricted mobility, proposing and evaluating a mix of static and mobile sinks and c) proposing a distributed protocol that tends to equally spread the sinks in the network to further improve performance. Our protocols are simple, based on randomization and assume locally obtainable information. We perform an extensive evaluation via simulation; our findings demonstrate that our solutions scale very well with respect to the number of sinks and significantly reduce energy consumption and delivery delay.

1. Introduction

Wireless Sensor Networks are visioned as large collections of very small autonomous devices, that can sense environmental conditions in their immediate surroundings and

have limited processing and communication capabilities. These smart nodes form ad hoc distributed sensing and data propagation networks that collect quite detailed information about the ambient environment. In a usual scenario, these networks are deployed in areas of interest for fine grained monitoring in different classes of applications [2]. The sensor devices are battery powered, thus energy is their most precious resource since periodically replacing the battery in large scale deployments is infeasible. The collected data is usually disseminated to a *static* control point – data sink in the network, using node to node – *multihop* data propagation, [9, 12]. This process consumes significant amounts of energy especially in the area near the sink where nodes need to relay the data from nodes that are farther away. Thus, a hotspot of increased energy consumption emerges and failure of these nodes due to strained energy resources leads to a disconnected and dysfunctional network.

Mobility. In recent years, a new category of important sensor networks applications emerges where motion is a fundamental characteristic of the examined system. In such applications sensors are attached to vehicles, animals or people that move around large geographic areas. Data exchange between individual sensors and infrastructure nodes will drive applications such as traffic and wild life monitoring, smart homes and hospitals and pollution control. In scenarios where some or all of the nodes are mobile, network topology is highly dynamic and connectivity can't be guaranteed, the usual approach of having one or more statically placed control centers requires the implementation of complex protocols in order to cope with increased network dynamics thus leading to increased resource use and inefficient operation.

Motivated by these developments, a new approach has been introduced that shifts the burden of delivering the data, from the sensor nodes to the sink. The main idea is that

*This work has been partially supported by the IST Programme of the European Union under contract number IST-2005-15964 (**AEOLUS**). Also, by the Programme **PENED** under contract number 03ED568, co-funded 75% by European Union – European Social Fund (ESF), 25% by Greek Government – Ministry of Development – General Secretariat of Research and Technology (GSRT), and by Private Sector, under Measure 8.3 of O.P. Competitiveness – 3rd Community Support Framework (CSF).

the sink has significant and easily replenishable energy reserves and can move inside the area the sensor network is deployed. When moving inside this area, the sink is constantly in close proximity to a (usually small) subset of the sensor devices and can acquire the data collected by these nodes at very low energy cost. By travelling in the whole network area, the control center is capable of collecting all the available data.

However, traversing the network area in a timely and efficient way is critical since failure to visit some areas of the network will result in data loss, while infrequently visiting some regions will result in large delivery delays. Also, routing and localization problems in the case of mobile sinks become more difficult to cope with. Despite the apparent difficulties, this data collection paradigm has many attractive properties [15]. A mobile sink that moves close to the nodes can help conserve energy since data is transmitted over fewer hops thus reducing the number of transmitted packets. The extra energy spent for the operation and movement of the sink doesn't affect overall sensor network lifetime since the mobile sink is considered an external to the network factor e.g. could be a man navigated vehicle or an unmanned robot that periodically returns to a fixed point in order to recharge itself.

Another important advantage is that sparse and disconnected networks can be better handled. A mobile sink allows to monitor a region with fewer sensor devices thus decreasing the operational cost of the network. Also, the sensor devices can reduce their transmission range to the lowest value required to reach the mobile infrastructure. Additionally, the mobile sink can possibly navigate through or bypass problematic regions where sensor devices can't operate, such as small lakes, large boulders that block the propagation path and other obstacles. In such conditions, conventional multi-hop protocols either fail or spend too many resources in order to overcome such obstacles.

Moreover, increased throughput and data fidelity can be achieved with the use of a mobile sink. By reducing the number of hops, the probability of transmission error and collisions also reduces. Also, for applications where security is concerned, with the use of a mobile sink the presence of the network is more difficult to detect since few low power transmissions are used. Also, attempts to compromise sensor nodes or deploy "bugs" in order to overhear the collected data become irrelevant. Only the information regarding a small area can be collected by the adversary because the data does not traverse many hops. A very nice and complete categorization and discussion of several different mobility models and their relevance to different application types and real world scenarios can be found in [18].

Our Contribution. In this work we propose and investigate *sink mobility* as a method for efficient, scalable and robust data delivery in wireless sensor networks. Our work is one

of the first few attempts in the relevant state of the art that introduces *multiple mobile sinks*. We propose three protocols that consist of a *mobility pattern* for the sink and an appropriate *data collection* method. Our mobility patterns, in contrast to most other relevant solutions where sinks move in a strictly deterministic manner, are mostly *randomized*. Starting with the simple random walk we also propose a combination of random walk and deterministic biased walk that attempts to equally distribute mobile sinks in the whole network area. In particular: a) the first protocol introduces many mobile sinks; its main advantage is significantly reducing latency, even with few sinks; also, energy dissipation is very low. b) Our second protocol introduces a more aggressive data collection strategy and uses a combination of static together with mobile sinks; great benefits, in terms of reduced latency and increased delivery rate, can be achieved even when only a small percentage of mobile sinks is used. c) Our final protocol builds upon the second protocol and further enhances its' performance. By loosely coordinating the mobile sinks using only local information, even lower latency and higher delivery rates are achieved. To retrieve the data from sensors, the sink movement is combined with *two data collection strategies*: a passive single-hop and a limited multi-hop.

We investigate several important performance properties of our protocols through a detailed, large scale simulation evaluation. We emphasize in examining the effect of having *different degrees of mobility* and on the *scalability* of increasing the number of sinks, mainly with respect to the delivery delay. Our findings demonstrate that by lightly increasing the number of mobile sinks we can greatly reduce the delivery delay, without introducing any additional energy overhead on the sensors. Even when having many static sinks, adding a number small of mobile sinks can significantly increase performance.

Moreover, we assume a *weak yet still realistic model*: sensors and sinks are not required to have network topology knowledge, the mobility strategies assume and exploit locally available information and the proposed data propagation protocols are simple enough to be implemented in current day sensor devices (see [1]). We feel that our work is a significant step towards introducing and studying mobility of multiple nodes and loosely coordinated mobility in wireless sensor networks, e.g. having a network composed of mobile sensors (or combinations of mobile and static sensors). Though efficient and important in practice, mobility introduces complications and new challenges for protocol design that should be investigated in future research.

Related Work and Comparison. Wireless sensor networks have been extensively studied and many interesting and efficient routing and energy conservation protocols have been developed for the case of statically placed sensors and sinks. In [3], **PEQ** is presented, an algorithm that

uses the publish/subscribe paradigm to disseminate requests across the network and an ack-based scheme to provide fault tolerance and low latency. [4] proposes a cluster-based routing protocol that reduces latency and network traffic, by grouping sensor nodes in clusters; nodes with more residual energy are selected as aggregator nodes that relay data to the sink. Also, [6] proposes and evaluates a propagation protocol that optimizes paths using limited knowledge. The protocol operates in phases, first it gathers locally network knowledge of the immediate area surrounding a sensor, subsequently this knowledge is used to optimize the routing path, thus achieving low power consumption and delay as well as high resilience in the presence of failures.

In [17], **SW-PFR** is proposed which is a randomized multipath routing protocol that performs a probabilistic selection to route messages with high probability over paths close to the optimal line connecting the sender and the sink. **SW-PFR** also reduces the energy consumption by alternating the operating state of sensor nodes between normal operation and a low power consumption “sleep” state. Also, in [11] a randomized routing algorithm is proposed, that uses an elaborate probabilistic selection in order to favor nodes with high energy and route messages over long distance hops, close to the optimal route. The authors combine their algorithm with a sleep-awake scheme and show that high delivery rate and low energy consumption is achieved. A sleep-awake based power conservation protocol that adapts to local conditions, is proposed and extensively analyzed in [7], under dynamic and heterogeneous network settings.

Although, mobility has been extensively examined in the context of mobile ad-hoc wireless networks and many fine communication protocols have been proposed, most of the findings do not directly apply to wireless sensor networks. Recently, applications that motivate mobility in wireless sensor networks appeared; [14] is a case study of applying peer-to-peer techniques in mobile sensor networks designed for wildlife position tracking for biology research. In [5] a data sink was mounted on a public transportation bus. The sensor nodes learn the times at which they have connectivity, and wake up accordingly to transfer their data.

In [19] a three-tier architecture is proposed that exploits the random motion of mobile entities, such as humans or animals, in order to collect information from the sensors and relay that information to access points. Using a simplified analytical model of a sensor network it is shown that great energy savings can be achieved at the cost of increased latency. In [15] the authors perform an experimental evaluation of a small sensor network with one mobile entity that moves back and forth on a straight line. This idea is further extended in [13] where multiple mobile entities that move deterministically on precomputed linear trajectories are examined; an algorithm to load balance the data collection process is proposed under the assumption of full coverage

of the network by the mobile entities.

[16] nicely investigates how to optimally (with respect to energy) move the sink on a cycle (i.e. the optimal positioning and radius of the moving cycle are analytically estimated) considering multi-hop propagation effects, thus balancing energy consumption and increasing network lifetime. Optimization of the data propagation process using mobility is examined in [10]. Under the assumption that all sensor nodes can move in a controllable manner, the authors propose an algorithm for rearranging the position of nodes in a propagation path, thus minimizing hop-distance and reducing energy consumption.

In [20] the authors propose a two-tier data dissemination model that supports many sinks. Each sink selects a sensor node to act as its primary agent, that collects all information and propagates it to the sink. As a sink moves, the path to its primary agent must be maintained until the sink selects a one. Also, the sink issues queries that contain location information used in routing data back to the sink.

This work effectively and non-trivially continues previous research of our group. In [8] we gained an insight in the benefits of having a *single* mobile sink, when compared to the static sink approach, even in cases where sink mobility is reduced in a fixed trajectory. Here we significantly extend this previous work by examining the scalability of the mobility assumption (since we allow multiple sinks) and by introducing loose coordination among the mobile sinks.

Our approach is different to the presented solutions in at least the following aspects, a) our model is weak but realistic; we assume no geolocation capabilities on the sensor devices, we use only local knowledge of the network and we examine scenarios where mobility is reduced. At the same time we assume a detailed energy model and a realistic radio channel. b) Our protocols are very simple, with low memory and communication footprint, and distributed without centralized coordination. c) We investigate the impact of multiple mobile sinks and also we consider heterogeneous settings of mixtures of static and mobile sinks.

2. The Model

Sensor networks are comprised of a vast number of ultra-small homogeneous sensor devices (which we also refer to as sensors) (see also [9]). Each sensor is a fully-autonomous computing and communication device, characterized mainly by its available power supply (battery) and the energy cost of data transmission and the (limited) processing capabilities and memory. Sensors (in our model here) do not move. The network area $\mathcal{A}$ is a flat square region of size $D \times D$; this assumption can be easily relaxed to include general network areas of arbitrary shapes.

The positions of sensor nodes within the network area are random and in the general case follow a uniform distri-

bution. Let n be the number of sensors spread in the network area and let d be the density of sensors in that area (usually measured in numbers of particles/m²).

Sensor devices are equipped with a set of hardware monitors that can measure several environmental conditions. Each device has a *broadcast* (digital radio) *beacon mode* of fixed transmission range R , and is powered by a battery. Also a sensor is equipped with a *general purpose storage memory* (e.g. FLASH) of size C . Let $\mathcal{E}_i$ be the available energy supplies of sensor i at a given time instance. At any given time, each sensor can be in one of three different modes, regarding the energy consumption: (a) transmission of a message, (b) reception of a message and (c) sensing of events.

In our model, for the case of transmitting and receiving a message, we assume that the radio module dissipates an amount of energy proportional to the message's size. To transmit a k -bit message, the radio expends $E_T(k) = \epsilon_{trans} \cdot k$ and to receive a k -bit message, the radio expends $E_R(k) = \epsilon_{recv} \cdot k$ where $\epsilon_{trans}, \epsilon_{recv}$ are constants that depend on the radio module and the transmission range of the sensors. For the idle state, we assume that the energy consumed for the circuitry is constant for each time unit and equals E_{elec} .

Overall, there are three different types of energy dissipation: (a) E_T , the energy dissipation for transmission, (b) E_R , the energy dissipation for receiving and (c) E_{idle} , the energy dissipation for idle state. For the idle state, we assume that the energy consumed for the circuitry is constant for each time unit and equals E_{elec} (the time unit is 1 second). We note that in our simulations we *explicitly measure the above energy costs*.

There is set of k special nodes within the network area, which we call sinks, $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k\}$ that represent data collection points. We assume that any data packet received by any $\mathcal{S}_i$ is considered delivered and the information it is transferring is available to the network implementors. This means that there aren't sink specific data flows and that sensors should propagate the collected information to any available sink to complete a sensing task. We assume that sinks can be deployed in a similar manner to sensor devices, thus their initial positions within the network are random. An important modelling assumption that differentiates our approach from most standard models in the state of the art is the *mobility* of the sinks. Sinks are not resource constrained i.e. they are assumed to be powerful in terms of computing, memory and energy supplies. Sinks can calculate accurately their position (e.g. by using navigational equipment, such as GPS) and is aware of the boundaries of the network area. Sinks moves driven by a high level mobility function which we symbolize by $\mathcal{M}$. If p_n is the position of the sink in a given moment then $\mathcal{M}(p_n)$ will return a new position p_{n+1} towards which the sink should move.

This defines a trajectory for the sink as a series of points $p_0, p_1 = \mathcal{M}(p_0), p_2 = \mathcal{M}(p_1), \dots, p_n = \mathcal{M}(p_{n-1})$. Also, the function $\mathcal{M}$ defines the speed $s_n = \mathcal{M}(s_{n-1})$ by which the sink moves from position p_{n-1} to position p_n , the speed is bounded by a maximum value which is depended on the scenario and models the mobility capacity of the sink; we call this limit s_{max} . A valid definition of $\mathcal{M}$ returns positions that are within the network area and $s \leq s_{max}$. The mobility function can be invoked at anytime even before reaching the designated point. The actual mechanism that moves a mobile entity from position p_{n-1} to position p_n is beyond the scope of this paper since it can be a human driver or an automated navigation system. However, in order to simplify our model we assume that all changes in speed and direction can be done instantly. Finally, we assume that a specific, high-level, sensing application is executed by the particles that form the network which we model in an abstract way by defining the message generation rate in each sensor.

3. Mobility strategies for multiple sinks

A) The fully mobile strategy. In this section we examine the simple case of *having k sinks moving independently, without any coordination, inside the network area collecting data*. The sinks follow each a simple *random walk* movement, where they can move chaotically towards any direction. We define $\mathcal{M}_{random}$ as a function that implements random walks in our scenarios. At each invocation $\mathcal{M}_{random}$ selects an angle as a random number uniformly in $[-\pi, \pi)$ radians. This angle defines the deviation from the mobile sinks current direction. In our version of random walk the speed of the movement is constant s_{randw} and predefined by the network implementors. To determine the new position, $\mathcal{M}_{random}$ selects a uniform random distance $d \in (0, d_{max}]$ which is the distance to travel along the newly defined direction. If the new position falls outside the network area, $\mathcal{M}_{random}$ crops the position to fall on the boundary of the area. This is the simplest possible movement; no network knowledge at all is assumed. Furthermore, this method is very robust, since due to its randomness it guarantees visiting all sensors in the network and thus collecting data even from disconnected areas of few/faulty sensors. However, in some network structures it may become inefficient, mostly with respect to latency, especially in the case of one (or few) sinks, as shown by our latency results in Sec. 4.

Data is collected in a passive manner. Periodically a beacon message is transmitted from each sink (at randomly selected times to avoid creating a broadcast storm problem). Each sensor node that receives a beacon attempts to acquire the medium and transmit its data to the corresponding sink. Clearly, this approach minimizes energy consumption since

only a single transmission per sensed event is performed; however time efficiency may drop due to long times needed to visit all sensors, especially in the case of a few sinks only.

B) A mixed strategy of static and mobile sinks. In several cases having k mobile elements is infeasible, due to the added cost of moving all sinks, or undesirable due to other factors e.g. since mobile sinks move unattended away from the supervision of the network implementors, they are more prone to failure, sabotage or theft.

We present here a hybrid solution where only a fraction of the sinks are moving. Let S_s the set of static (non-moving) sinks and $k_s = |S_s|$. Also, let S_m be the set of mobile sinks and $k_m = |S_m|$, then $S = S_s \cup S_m$ and $k = k_m + k_s$. In this setting the k_m sinks move according to $\mathcal{M}_{random}$ described earlier, while the k_s sinks remain fixed at their initial positions.

Data is collected by the protocol presented here which forms limited propagation trees with a sink as the root of each tree. Each sink periodically broadcasts a beacon message, which carries the identity of the sink, a hop counter h_c (initially $h_c = 0$) and a time to live counter TTL . Each sensor node maintains a hop distance h_d from the nearest available sink (initially $h_d = \infty$), a timestamp t_u indicating the last time h_d was updated and the network address of a parent sensor node (initially null). When a beacon is received the sensor decides if it needs to update his parent node according to the following rules: (1) If $h_c < h_d$, the node sets the sender of the message as its parent, sets $h_d = h_c$, sets $t_u = t_b$, increases $h_c = h_c + 1$, reduces $TTL = TTL - 1$ and if $TTL > 0$ broadcasts the beacon message to its neighbors. (2) If $h_c \geq h_d$ and $t_b > t_u + t_{thres}$, then the node sets the sender of the message as its parent, sets $h_d = h_c$, sets $t_u = t_b$, increases $h_c = h_c + 1$, reduces $TTL = TTL - 1$ and if $TTL > 0$ broadcasts the beacon message to its neighbors. The value t_{thres} is set by the network operator and expresses the time a propagation tree is considered valid. (3) If $h_c \geq h_d$ and $t_b \leq t_u + t_{thres}$, the node simply discards the beacon.

Sensor nodes that belong to a propagation tree may begin immediately forwarding their data to the corresponding sink. The depth of these trees is determined by the TTL value which is an operational parameter of the sinks. In this way, the network operator can tune the trade-off between reduced delay and increased energy consumption.

As a sink moves, whole propagation trees may become disconnected. When a node with $h_d = 0$ can no longer communicate to the sink, it simply caches all data, both generated and relayed, and waits to hear another beacon message with $h_c = 0$ to begin the propagation process again or after t_{thres} time the node initiates participation to a new tree.

C) The Equidistribution protocol. In both cases examined so far, sinks do not coordinate with each other thus leading to inefficient use of the sink multiplicity, since some nodes

may be within range of 2 or more sinks, while others are infrequently or not at all visited. Intuitively, a first attempt to solve this problem would be to have the sinks move to a specific position each so as to equally distribute the network nodes between sinks. However, this solution assumes that sinks have global knowledge, at least of the other sinks' positions and the network structure. Also, it would create hot-spot areas near each sink where sensor nodes are more strained due to relaying data. Furthermore, in case of dynamic changes in the network or uneven initial node distribution, sinks would receive unbalanced load. For the same reasons manually placing static sinks at a uniform arrangement may not always be possible or cost effective. The solution we present here loosely coordinates sinks based only on locally, at each sink, available information. More specifically, mobile sinks try to mutually detect the presence of other sinks and change their movement accordingly so as to avoid being in the same area.

More specifically, we assume that k_m and k_s sinks are deployed at random positions and perform limited multihop data collection as described in the previous section. The main difference is that sinks also include their position and a sequence number within the beacon messages. When a sink S_i overhears a beacon message from another sink S_j it attempts to modify its trajectory in order to avoid getting too close to the other sink. Specifically, S_i calculates a vector $\vec{r}$ with orientation along the line connecting S_i , S_j and direction away from S_j . Also, $|\vec{r}| = \gamma \cdot |s_{S_i}|$ where γ is a value depending on the distance of the 2 sinks, more specifically:

$$\gamma = \begin{cases} \frac{R}{d} & \text{if } \lceil \frac{d}{R} \rceil \leq \beta \\ 0 & \text{otherwise} \end{cases}$$

where d is the distance between S_i and S_j and R the transmission range of the sinks. β is a protocol parameter set by the network implementor that effectively controls the hop distance (given by $\lceil \frac{d}{R} \rceil$) within which the two sinks can affect each other. From this formula it's clear that $\gamma = 1$ when S_i and S_j are only 1 hop away, while γ becomes smaller when the sinks are farther apart. Then, S_i calculates a new direction of movement as follows: $\vec{s}_{S_i}' = \vec{r} + s_{S_i}$, while maintaining the velocity constant $|\vec{s}_{S_i}'| = |s_{S_i}|$. The same symmetric effect is applied on sensor S_j , in the usual case when S_j also detects the presence of S_i . For a graphical representation of initial movement, intersection and adjusted movement see Fig. 1. Also, S_i stores the information (sequence number) about S_j , the next time it receives a beacon from S_j it performs the trajectory recalculation only if the newly received beacon has a greater sequence number than the one stored. When a sink is not receiving beacon messages that force it to recalculate its trajectory, it follows the $\mathcal{M}_{random}$ mobility protocol.

Note that while this protocol causes nearby sinks to drift

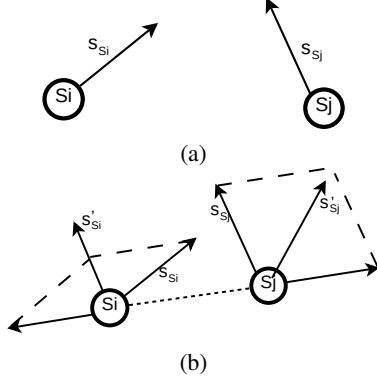


Figure 1. The effect of equidistribution a) Initial movement vectors b) Intersection and new movement vectors (notice that $|s'_{S_i}| = |s_{S_i}|$ and $|s'_{S_j}| = |s_{S_j}|$)

further apart from each other, it uses the random walk mobility strategy most of the time, so as to maintain properties, such as fairness, discussed earlier. Also, since the estimation of the positions of other sinks is based on purely local thus possibly inaccurate information, the equidistribution protocol doesn't completely guarantee that the sinks will be distributed equally in the network area at all times.

4. Experimental evaluation

4.1. Simulation Setup and Metrics

We implement and evaluate the above discussed protocols in the network simulator platform *ns-2* version 2.26. In particular we significantly extend the mobility functionality present in *ns-2*, to allow for much finer control when moving a node as required by the equidistribution protocol. We considered different simulation setups for various network sizes, number of sinks and mobility parameters. The particular settings for the results presented below are as follows: the size of the network area is $200m \times 200m$, 300 sensor nodes are deployed in a random uniform manner. The transmission range of the sensors and of the sinks in $\mathcal{S}$ is set to $R = 15m$ and the speed of the movement of the sinks is set to $4m/s$. The mobile sinks for all protocols transmit a beacon message every $1sec$.

The initial energy reserves of the nodes are high enough to ensure that nodes remain operational for all the duration of the simulation, also here we do not consider the possibility of node failures that would make harder to investigate the behavior of our protocols. The values of ϵ_{trans} , ϵ_{recv} and E_{idle} were set to match as close as possible the specifications of the *mica mote* platform [1]. We assume that a high

level periodic monitor application is executed by the sensor devices, the application is triggered at the beginning of the simulation and registers data about the network region. The data is generated at random times in packets of 36 bytes and at an average rate of 1 message every 10 seconds, the size of a beacon message is 24–32 bytes. Each node has a cache of 256KB. Each sensor device transmits 100 messages before the monitored phenomenon ends, meaning that the data generation phase lasts for about $1000sec$. During the data generation phase the mobile sinks are collecting data and another $4000sec$ of simulation time are given in order to collect all the data, leading to $5000sec$ of simulation time.

In our setting it is not feasible to calculate optimal positioning for the sinks since the positions of sensors are unknown, thus for all test cases the sinks, whether mobile or static, are deployed in a random uniform way within the network area. We conduct several experiments for each protocol, for evaluating the simple random walk with single-hop collection protocol we varied the number of sinks $k \in \{1, 2, 3, 4, 5, 6, 7\}$. For our second protocol we tried different combinations of static and mobile sinks, we set $k_s \in \{1, 2, 3, 4, 5, 6, 7\}$ and $k_m = k - k_s$ for $k = 8$, we also examine how our limited multihop scheme operates under this dynamic setting by varying the number of performed hops setting $TTL \in \{4, 6, 8\}$. For the equidistribution protocol we repeat the same experiments as in the previous setting but only for $TTL \in \{4, 6\}$ and set $\beta \in \{2, 3, 4\}$ for $TTL = 4$ and $\beta \in \{3, 4, 5\}$ for $TTL = 6$.

Conducting these experiments, we measure several metrics that depict the behavior of our protocols. We call *success rate* the percentage of data messages that were received by all sinks over the total number of generated messages. We measure the *energy* consumed at the sensor network (i.e. we do not measure the energy consumption of the mobile entities), as an absolute value in Joules. The *delivery delay* is defined as the average time interval between the creation of a message and the time when it is delivered to any of the sinks. Note that we assume that as soon as at least one sink gets a data packet, that data is considered successfully delivered (so there is no need to study intra-sink communication aspects). Thus, the overhead of having many mobile sinks is more or less proportional to their number; so, studying the impact of the number of sinks, we implicitly investigate the overhead incurred as well. Each particular setting is executed at least 10 times and the above metrics are averaged between all executions.

4.2. Simulation Results and Discussion

A) Fully Mobile Strategy. In Figures 2-4 we present the results of the evaluation of our first protocol. It is clear that by increasing the number of mobile sinks the overall network performance increases. Especially when considering

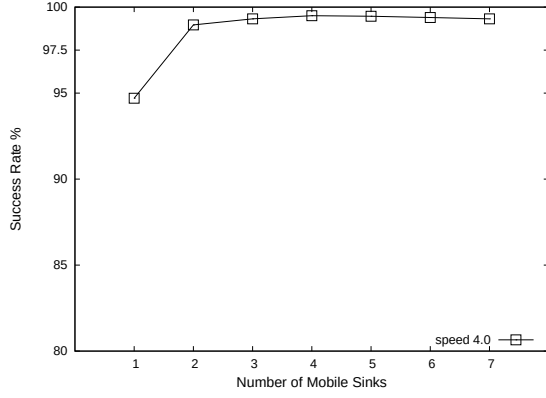


Figure 2. Success rate for multiple mobile sinks with random walk and passive data collection

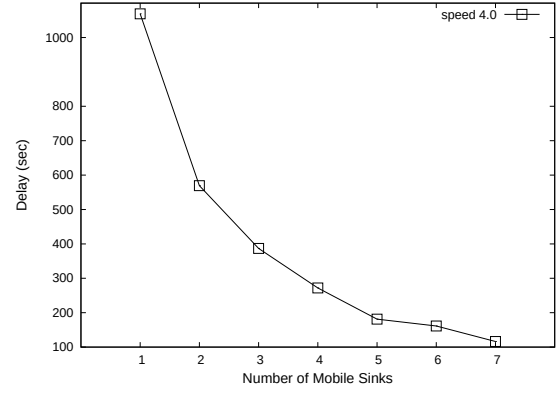


Figure 4. Average delay for multiple mobile sinks with random walk and passive data collection

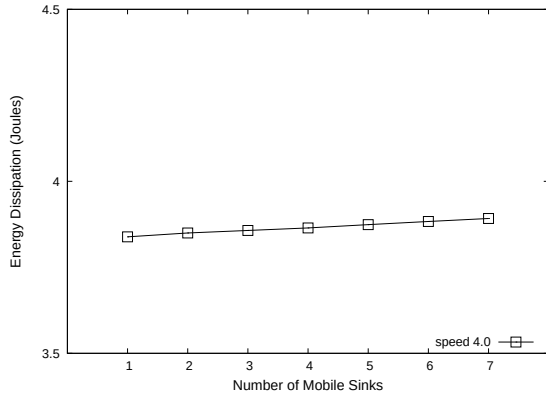


Figure 3. Energy dissipation for multiple mobile sinks with random walk and passive data collection

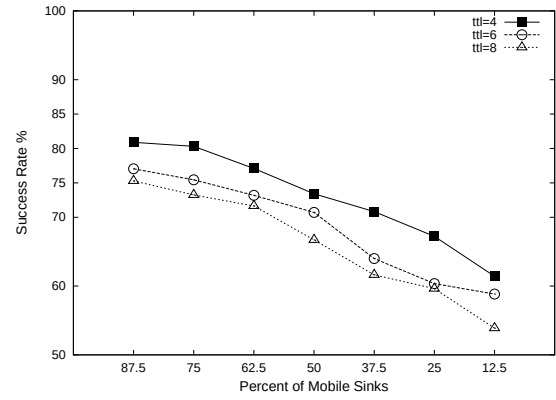


Figure 5. Success rate for mixed case of static and mobile sinks with random walk and limited multihop data collection

the average delivery delay (Fig. 4) we can see that we have a reduction by a factor of 10 (from about 1100sec to 115sec) with 7 mobile sinks, but even 4 mobile sinks suffice to significantly reduce latency. Success rate (Fig. 2) also tops at about 99% with only 3 sinks, while there is an insignificant increase in the energy consumption (see Fig. 3) due to the increased number of beacon messages received by the sensors. Our findings show that even with a small number of mobile sinks (up to 5) significant gainings can be achieved especially with respect to the reduction in the latency. There also appears to be a threshold in this improvement since further increasing the number of mobile sinks doesn't result in proportional improvements. Nevertheless this experiment shows that by adding multiple mobile sinks performance scales nicely according to the number of sinks.

B) Mixed (Static and Mobile) Case. In Figures 5-7 we present the results of the evaluation of our second protocol.

When examining the success rate (Fig. 5) we can see that as the amount of mobility is reduced the delivery rate also drops, which is expected since the static sinks only cover a limited area of the network and mobile sinks are required to visit the rest of the nodes. We also notice that despite of the increase in *TTL*, the success rate slightly drops. This is due to the increased overhead in propagation tree creation and reconfiguration and the inability of our simplified data propagation method to quickly detect and reconfigure a broken tree which leads to some data loss. Still, a very interesting finding is that a relatively low mobility percentage like when having 50% of the sinks mobile (in our case 4 sinks) suffices to keep success rate quite high.

In Fig. 6 we can see that the energy dissipation tends to drop as the amount of mobility is decreased, since the number of path reconfigurations required is smaller in this case; but also because fewer messages are delivered hence less

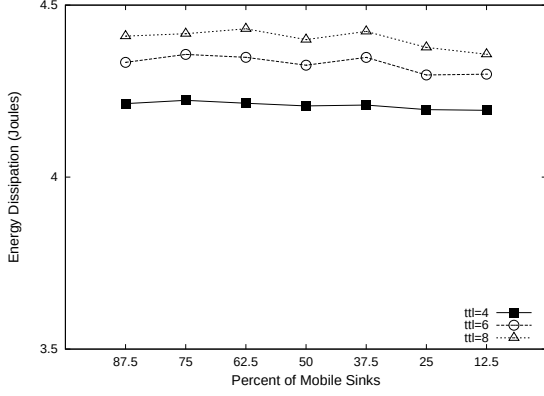


Figure 6. Energy dissipation for mixed case of static and mobile sinks with random walk and limited multihop data collection

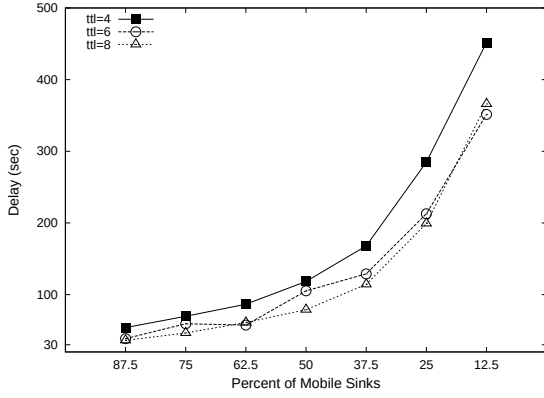
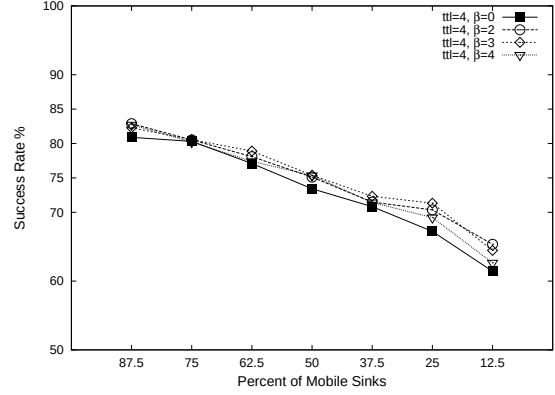


Figure 7. Average delay for mixed case of static and mobile sinks with random walk and limited multihop data collection

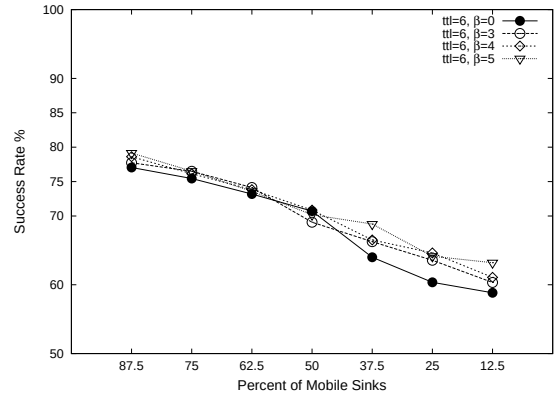
energy is spent. Still, it is very interesting that a very low mobility percentage (just 1 sink) keeps the energy dissipation at very low levels, essentially of the same order as in the fully mobile case. Also, in Fig. 7 we can see that the delivery delay gradually increases as the amount of mobility is decreased, which as we already discussed is due to the partial network coverage by the static sinks. Again, however with only 3 mobile sinks we can achieve quite low latency. Also, in this case we can see that when increasing the TTL value the delivery delay is slightly reduced.

Overall we observe that when full network coverage can't be achieved by static sinks, having a limited number of mobile sinks (up to 50% of the whole number of sinks) can significantly improve delivery delay and success rate. The findings regarding the mixed case suggest that the impact of mobility is very positive on all metrics and that performance

drops as mobility decreases. However, in all cases having a few mobile sinks, keeps performance at satisfactory levels. This is an encouraging result since sometimes the amount of mobility that can be used is limited; in such cases our findings indicate what mixture the network implementor should use to achieve certain performance.



(a)

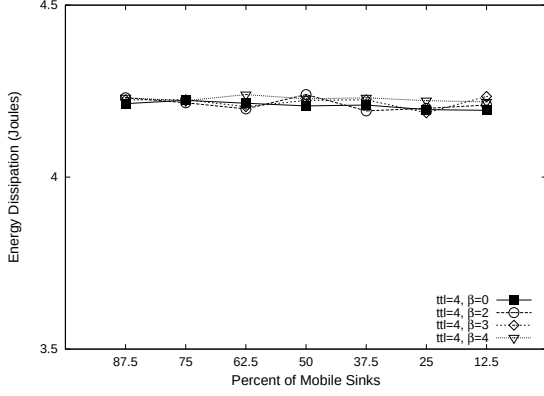


(b)

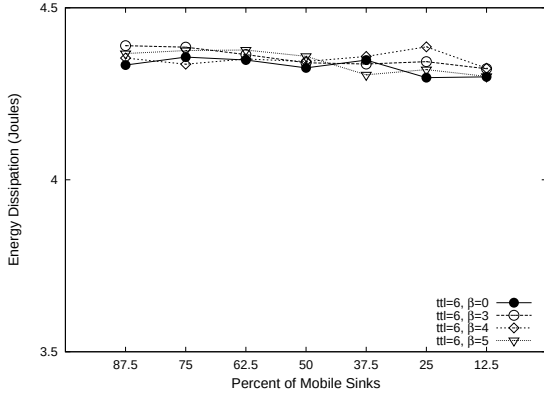
Figure 8. Success rate for mixed scenario of static and mobile sinks for the equidistribution protocol for $\beta \in \{2, 3, 4\}$ when $TTL = 4$ (top) and $\beta \in \{3, 4, 5\}$ when $TTL = 6$ (bottom)

C) The impact of the Equidistribution protocol. In Figures 8–10 we can see the performance metrics for the equidistribution protocol. In Fig. 8 we can see that the equidistribution protocol increases the success rate by up to 10% for a small percentage (25%) of mobile nodes when compared to the case when $\beta = 0$, i.e. when the equidistribution is off as in our second set of experiments. For different values of β the results seem slightly better for small values of β .

At the same time Fig. 9 we can see that the energy dissipation slightly increases which is somewhat expect since the higher success rate incurs an additional cost. For higher



(a)

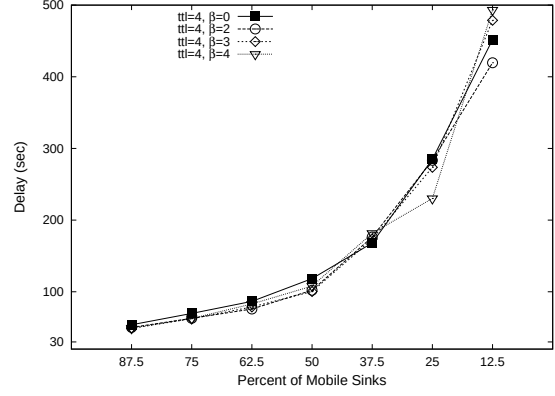


(b)

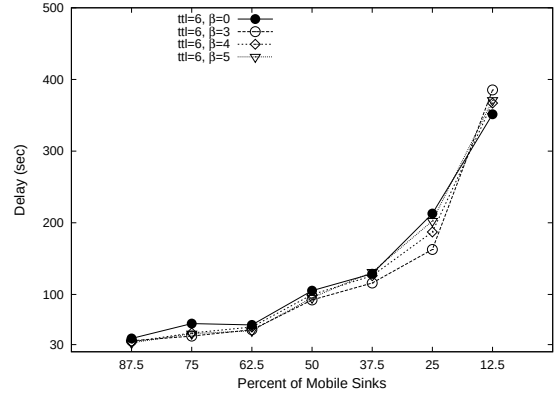
Figure 9. Energy dissipation for mixed scenario of static and mobile sinks for the equidistribution protocol for $\beta \in \{2, 3, 4\}$ when $TTL = 4$ (top) and $\beta \in \{3, 4, 5\}$ when $TTL = 6$ (bottom)

values of β the results seem slightly better. In Fig. 10 we can see that the equidistribution protocol also improves the delivery delay in general. Especially as the percentage of mobile nodes gets smaller, a very interesting behavior is observed. When the percentage of mobile nodes is between (50% – 25%) the best results are achieved, however for further reduction to 12.5% we experience longer delays than without the equidistribution. A possible explanation is that the mobile sink avoids areas with several static sinks; thus a small number of nodes, that is in between the mobile sinks but outside their area of service, is not frequently visited by any mobile sink.

Overall, the use of the equidistribution protocol tends to increase performance; still, in order to get more impressive performance gains sink interactions can be made more aggressive i.e. the size of the interaction vector can be made larger (by increasing γ), the distance at which interactions happen can be made smaller (by decreasing β).



(a)



(b)

Figure 10. Average delay for mixed scenario of static and mobile sinks for the equidistribution protocol for $\beta \in \{2, 3, 4\}$ when $TTL = 4$ (top) and $\beta \in \{3, 4, 5\}$ when $TTL = 6$ (bottom)

5. Conclusions and Future Work

In this work we presented 3 protocols that exploit sink mobility in order to improve overall network performance. Each protocol assumes different degrees of mobility and is combined with an appropriate data collection protocol. Our protocols are based on randomized methods and introduce a novel approach in loose coordination. Our results show that significant gainings, in terms of increased success rate and reduced energy consumption and delivery delay, can be achieved when exploiting the mobility assumption. Furthermore, in each of the examined protocols the achieved performance is more or less, analogous to the amount of assumed mobility. By introducing more mobile nodes performance scales accordingly. Finally, an interesting finding is that, while performance drops when mobility decreases, a relatively low mobility percentage (a few mobile sinks) suffices to keep performance at satisfactory levels.

In future research we intent to devise more elaborate sink coordination schemes based on the insight gained by the equidistribution protocol. We will also continue to investigate in greater detail scalability and tradeoff issues in wireless sensor networks with mobile sinks and also we plan to extend our model to represent more dynamic networks. Finally, we plan to compare the performance of our protocols to other existing relevant protocols.

References

- [1] Crossbow technology inc., mica motes. <http://www.xbow.com/>.
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. In *Journal of Computer Networks*, vol. 38, pages 393–422, 2002.
- [3] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 157–164, New York, NY, USA, 2004. ACM Press.
- [4] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo. Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol. In *30th IEEE Conference on Local Computer Networks*, pages 560–567, November 2005.
- [5] A. Chakrabarti, A. Sabharwal, and B. Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In *IPSN*, 2003.
- [6] I. Chatzigiannakis, A. Kinalis, and S. Nikolettseas. Efficient and robust data dissemination using limited extra network knowledge. In *2nd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 218–233. Also, in the *Journal of Parallel and Distributed Computing (JPDC)*, to appear in 2007.
- [7] I. Chatzigiannakis, A. Kinalis, and S. Nikolettseas. An adaptive power conservation scheme for heterogeneous wireless sensor networks with node redeployment. In *17th Annual Symposium on Parallel Algorithms and Architectures (SPAA 2005)*, pages 96–105, 2005. Also, in the *Theory of Computing Systems Journal (TOCS): Special Issue on selected papers of SPAA05*, to appear in 2007.
- [8] I. Chatzigiannakis, A. Kinalis, and S. Nikolettseas. Sink mobility protocols for data collection in wireless sensor networks. In *4th ACM/IEEE International Workshop on Mobility Management and Wireless Access Protocols (MobiWac)*, pages 52–59, Torremolinos, Spain, October 2006.
- [9] I. Chatzigiannakis, S. Nikolettseas, and P. Spirakis. Smart dust protocols for local detection and propagation. In *2nd ACM International Annual Workshop on Principles of Mobile Computing (POMC 2002)*, pages 9–16, 2002. Also, in the *ACM Mobile Networks (MONET) Journal, Special Issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks*, in MONE 10:1 (February 2005).
- [10] D. Goldenberg, J. Lin, A. Morse, B. Rosen, and Y. Yang. Towards mobility as a network control primitive. In *5th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc04)*, 2004.
- [11] A. Gopalan and T. Znati. A context-aware data forwarding algorithm in sensor networks. In *38th Annual Simulation Symposium (ANSS 2005)*, pages 7–14, 2005. IEEE Press.
- [12] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *6th ACM/IEEE Annual International Conference on Mobile Computing (MOBICOM 2000)*, pages 56–67, 2000.
- [13] D. Jea, A. Somasundara, and M. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *1st IEEE Conference in Distributed Computing in Sensor Systems (DCOSS)*, pages 244–257, 2005.
- [14] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS02)*, 2002.
- [15] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In *2nd ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys04)*, 2004.
- [16] J. Luo and J.-P. Hubaux. Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. In *24th IEEE INFOCOM*, Miami, USA, 2005.
- [17] S. Nikolettseas, I. Chatzigiannakis, H. Euthimiou, A. Kinalis, T. Antoniou, and G. Mylonas. Energy efficient protocols for sensing multiple events in smart dust networks. In *37th Annual Simulation Symposium (ANSS 2004)*, pages 15–24, 2004. IEEE Press.
- [18] C. Schindelhauer. Mobility in wireless networks. In *32nd Annual Conference on Current Trends in Theory and Practice of Informatics*, Czech Republic, January 2006.
- [19] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, 2003.
- [20] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks. In *ACM MOBICOM*, 2002.