Sink Mobility Protocols for Data Collection in Wireless Sensor Networks *

Ioannis Chatzigiannakis
Computer Technology Institute (CTI) and University of Patras, Patras, 26500, Greece
ichatz@cti.gr

Athanasios Kinalis
Computer Technology Institute (CTI) and University of Patras, Patras, 26500, Greece
kinalis@cti.gr

Sotiris Nikoletseas
Computer Technology Institute (CTI) and University of Patras, Patras, 26500, Greece
nikole@cti.gr

ABSTRACT

In wireless sensor networks data propagation is usually performed by sensors transmitting data towards a *static* control center (sink). Inspired by important applications (mostly related to ambient intelligence) and as a first step towards introducing mobility, we propose the idea of having *a sink moving in the network area and collecting data from sensors*. We propose *four characteristic mobility patterns* for the sink along with *different data collection strategies*. Through a detailed simulation study, we evaluate several important performance properties of each protocol. Our findings demonstrate that by taking advantage of the sink's mobility, we can significantly reduce the energy spent in relaying traffic and thus greatly extend the lifetime of the network.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless Communication

General Terms: Algorithms, Design, Performance

Keywords: Data Propagation, Mobility, Performance Evaluation, Wireless Sensor Networks

1. INTRODUCTION, CONTRIBUTION AND RELATED WORK

Wireless Sensor Networks are visioned as very large collections of smart sensor nodes that 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 largely deployed in areas of interest for fine grained monitoring in different classes of applications [2]. The sensor devices are battery powered thus energy is the most precious resource of a wireless sensor network since periodically replacing the battery of the nodes in large scale deployments is infeasible. The collected data is disseminated to a *static* control point – data sink in the network, using node to node – *multi-hop* data propagation, [6, 8]. However, sensor devices consume significant amounts of energy in addition to increased implementation complexity since a routing protocol is executed. Also, a point of failure emerges in the area near the control center where nodes relay the data from nodes that are farther away.

Mobility. Additionally, 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. Clearly, the usual approach of having a statically placed control center is unable to operate efficiently in such scenarios.

Recently, a new approach has been developed that shifts the burden from the sensor nodes to the sink. The main idea is that the sink has significant and easily replenishable energy reserves and can move inside the area the sensor network is deployed. When moving inside the network 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. The mobility assumption may be especially useful in particular application classes, e.g. in emergency preparedness [4] (where a set of nodes called actors aggregate and evaluate the collected data in order to detect and react to emergency situations), mobile elements can move in close proximity of an area and better assess an emergency situation.

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, e. g. nice solutions in the state of the art (like [15] that performs geometric routing greedily by keeping vectors of hop distance between nodes) must be extended. Despite the apparent difficulties, this data col-

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

MobiWac'06, October 2, 2006, Torremolinos, Malaga, Spain.

lection paradigm has many attractive properties [11]. The major advantage of having a mobile sink (or more) is an increase in system lifetime. A mobile agent that moves closer 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.

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.

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, security can be enhanced as the data does not traverse multiple hops across potentially compromised nodes, while an adversary sniffing the data packets from an area will receive only the information regarding that area. 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 [13].

Our Contribution. In this work we propose and investigate *sink mobility* as a method for efficient 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 introduce mobility of the sink. We propose four *mobility patterns* for the sink, mostly *randomized* (such as the simple random walk, biased random walks and walks on spanning subgraphs) as well as predictable mobility (moving on a straight line or cycle). These patterns assume and exploit different degrees of freedom, simplicity and network knowledge. To get data from sensors, the sink movement is combined with *three data collection strategies*: a passive, a multi-hop and a limited multi-hop.

Each approach has several advantages and disadvantages and achieves different trade-offs, mostly between energy dissipation and time efficiency. We investigate several important performance properties of our protocols through a detailed, large scale simulation evaluation. Our findings demonstrate that having a mobile sink can greatly extend the lifetime of the network. In fact, even by having a very limited sink mobility, the overall success rate can be improved by more than 50% and the energy dissipation drop by more than 30%. If the sink can be fully mobile (i.e. be able to visit all the areas of the network), we can further reduce the energy consumption and achieve very high success rates, very close to 100%.

Moreover, we feel that our work is a first step towards introducing mobility in wireless sensor networks, e.g. having many/multiple sinks collecting data in 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. Recently, applications that motivate mobility in wireless sensor networks appeared; [10] 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 with the bus, and wake up accordingly to transfer their data.

In [14] 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 a central control center. In [11] 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 [9] where multiple mobile entities that move on a line are examined and an algorithm to load balance the data collection process is proposed under the assumption of full coverage of the network by the mobile entities. [12] 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. Optimization of the data propagation process using mobility is examined in [7]. Under the assumption that all sensor nodes can move, the authors propose an algorithm for rearranging the position of nodes in a propagation path in order to improve performance.

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 [6]). 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 distribution. *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 $particles/m^2$). However, in several scenarios the network implementors are expected to deploy more sensors in areas where fine grained monitoring is required. We model such scenarios by assuming p_n "pockets" i.e. regions in the network area with high particle density. For the sake of simplicity, each pocket is a circular area of radius r_p, pockets don't overlap and the density of sensors in these areas is d_p. For the rest of the network area, the density is d_n. We denote $\mathcal{A}_p = \pi {r_p}^2 p_n$ the area occupied by the pockets. Let n_p the number of sensors contained in pockets, clearly $n_p = d_p \mathcal{A}_p$. Likewise, the number of sensors contained in the rest of the network is $n_n = d_n(\mathcal{A} - \mathcal{A}_p)$. Then the total number of sensors is given by $n = n_p + n_n$.

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 $\mathcal{C}$. 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.

There is a special node within the network area, which we call the sink $\mathcal{S}$, that represents a control center where data should be collected. An important modelling assumption that differentiates our approach from most standard models in the state of the art is the *mobility* of the sink. The sink is not resource constrained i.e. it is assumed to be powerful in terms of computing, memory and energy supplies. The sink can calculate accurately it's position (e.g. by using navigational equipment, such as GPS) and is aware of the dimensions of the network area. The sink moves following 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 $p_0, p_1 = \mathcal{M}(p_0), p_2 = M(p_1), \ldots, p_n = \mathcal{M}(p_{n-1})$. 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 the 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, application is executed by the particles that form the network. Applications for wireless sensor networks fall in three major categories [3]: (i) *Periodic Sensing* (the particles are always monitoring the physical environment and continuously report their sensors' measurements to the control center $\mathcal{S}$), (ii) *Event driven* (to reduce energy consumption, particles operate in a silent monitoring state and are "programmed" to notify about events) and (iii) *Query based* (queries can be propagated to the particles arbitrarily by the control center $\mathcal{S}$, according to the application and/or user's will). We model in an abstract way the kind of high-level application by the message generation rate in each sensor.

3. PROTOCOLS FOR DATA COLLECTION WITH SINK MOBILITY

There are many different approaches when considering the mobility pattern that the mobile sink should follow. Depending on the application scenario and the network size and conditions, different approaches may yield diverse results and affect drastically the achieved network performance. A coarse-grained categorization of mobility patterns will identify the following different approaches (see [13]): (i) Random mobility, (ii) predictable mobility and (iii) controlled mobility.

The differences between the categories of mobility suggest that different data collection protocols are appropriate in each case. There is certainly a large space of options for protocol design, depending on which mobility strategy and which data collection mechanism are used. We partially explore this space, by proposing and investigating below a few characteristic protocols composed of a mobility pattern and an appropriate data collection process, along with a summary discussion of their potential.

3.1 P1: Random Walk and Passive Data Collection

The simplest of all possible mobility patterns is the *random walk*, where the mobile sink can move chaotically towards all directions at varying speeds. We define $\mathcal{M}_{random}$ as a function that implements random walks in our scenarios. At each invocation $\mathcal{M}_{random}$ selects a random uniform angle 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 it guarantees visiting all sensors in the network and thus collecting data even from disconnected areas of few/faulty sensors or obstacle presence. However in some network structures it may become inefficient, mostly with respect to latency.

Data is collected in a passive manner. Periodically a beacon message is transmitted from the sink. Each sensor node that receives a beacon attempts to acquire the medium and transmit its data to the sink. This will lead to many collisions, thus an appropriate MAC layer protocol with an efficient backoff function is essential for the proper deployment of this protocol. 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 to visit sensors.

3.2 P2: Partial Random Walk with Limited Multi-hop Data Propagation

Another form of random walk is performed by using a set of predefined areas and random transitions between the areas according to their connectivity. In this mobility function $\mathcal{M}_{graph}$, a graph formation phase is executed by the sink during the network initialization. The network area is partitioned in $j \times j$ equal square regions, the center of each region is considered as a vertex in a graph that is connected with unidirectional edges only to the 4 vertices corresponding to adjacent regions. Thus a lattice graph G_o is created which is overlayed over the network area. Initially the mobile sink is positioned on or near one of the nodes of G_o, $\mathcal{M}_{graph}$ calculates the next position by selecting uniformly randomly one the neighbors of the current vertex in the graph. The speed of the movement is constant and predefined by the network implementors. In this way, the area covered by the sink in a single move covers the maximum number of sensor nodes and thus fewer distance needs to be traveled by the mobile element. Full coverage of the network is achieved when $j = \lceil D\sqrt{2}/R \rceil$ since every sensor node in a $D/j \times D/j$ area can communicate with the center of the area and thus the sink.

In this partial walk setting, a data collection protocol may need to communicate with nodes outside the immediate range of the sink. The protocol presented here forms propagation trees with the sink as root. The sink periodically broadcasts a beacon message, which carries a hop counter h_c (initially $h_c = 0$), a time to live counter ttl. Each sensor node maintains a hop distance h_d from the 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 it's 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 it's 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 it's 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 it's 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.

This process creates a number of propagation trees within the network with the roots of these trees being one hop away from the sink. Sensor nodes that belong to a propagation tree may begin immediately forwarding their data to the sink. The depth of these trees is determined by the ttl value which is an operational parameter of the mobile sink. In this way, the network operator can tune the trade-off between reduced delay and increased energy consumption. The value of ttl should be set according to the selected j, when the sink is positioned in the center of a square the maximum hop distance of a sensor node is $\lceil \frac{D/j\sqrt{2}}{R} \rceil$. Note that when $ttl = 1$ this method is equivalent to the passive collection described earlier.

As the 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 can participate to a new tree.

This protocol assumes and uses more knowledge of the network, it can accelerate times to visit network nodes, reduces the distance traveled by the sink when compared to P1 and leads to improved time efficiency. On the other hand, it is also more expensive in terms of communication and computational cost on the sensor devices.

3.3 P3: Biased Random Walk with Passive Data Collection

The idea of using a logical graph can be extended in a way that certain areas of the network are favored (i.e. more frequently visited) by the sink in order to improve the data collection process or to overcome problems that arise from the network topology. More specifically, the mobility pattern $\mathcal{M}_{bias}$ forms a logical overlay graph (as described in 3.2) $G_o = G(V, E)$ with $j = \lceil D\sqrt{2}/R \rceil$; it associates two counters c_v and d_v for every vertex v, initially $c_v = d_v = 0$ $\forall v \in V$. These counters store data about the network conditions for each area visited. When the mobile sink enters the area corresponding to vertex u it increases the associated counter c_u by 1. Also, while the sink remains in the area corresponding to vertex u, it increases the counter d_u for every previously unseen sensor node that has received a message from. Thus, the frequency of visits and the density of each area can be estimated by the sink. The selection of the next area to visit is done in a biased random manner depending on these two variables as described below.

Frequency biased. If the mobile element is currently on vertex u of degree deg_u, then we define $c_{neigh}(u) = \sum_v c_v$ for all $v : (u, v) \in E$. Then the propability $p(f)_v$ of visiting a neighboring vertex v is calculated as $p(f)_v = \frac{1 - c_v/c_{neigh}(u)}{deg_u - 1}$ when $c_{neigh} \neq 0$. When $c_{neigh} = 0$ we have $p(f)_v = 1/deg_u$. Thus less frequently visited areas are more likely to be visited when the sink is located at a nearby area.

Density biased. If the mobile element is currently on vertex u of degree deg_u, then we define $d_{neigh}(u) = \sum_v d_v$ for all $v : (u, v) \in E$. Then the propability $p(d)_v$ of visiting a neighboring vertex v is calculated as $p(d)_v = \frac{1 + d_v/d_{neigh}(u)}{deg_u + 1}$ when $d_{neigh} \neq 0$. When $d_{neigh} = 0$ we have $p(d)_v = 1/deg_u$. Thus, areas with many sensor devices are more likely to be visited when the sink is located at a nearby area.

After calculating $p(f)_v$ and $p(d)_v$ the final probability of transition to the area corresponding to vertex v is given by

$$p_v = \alpha \cdot p(f)_v + \beta \cdot p(d)_v$$

where α, β are positive real numbers such that $\alpha + \beta = 1$. Data is collected in a passive manner as described in 3.1.

This protocol uses knowledge collected by the sink in order to speed up the coverage of new areas (when $\alpha > \beta$) or increase data delivery in areas with many nodes (when $\alpha < \beta$). Overall, when compared to P1 we expect to achieve faster network coverage, higher delivery rates and lower latency with an increase in computational overhead at the sink.

3.4 P4: Deterministic Walk with Multi-hop Data Propagation

Here we use a simple form of controlled mobility where the mobile entity moves on a predefined trajectory. We examine the cases where the trajectory is a line (we call the mobility function $\mathcal{M}_{line}$) or a circle ($\mathcal{M}_{circle}$) that is fully contained in the network area. The trajectory is characterized by its length l. In particular, the linear trajectory consists of a horizontal or vertical line segment passing through the center of the network. The sink moves from one edge of the line to other and returns along the same path. For the case of the circular trajectory the circle is centered at the center of the network and its radius is defined as $r = \frac{l}{2\pi}$ (for comparison fairness with the line case). Initially the sink is positioned on the circumference of the circle and continues along this path.

Since the mobile sink covers only a small network area it is necessary to collect data with a multi-hop data propagation protocol. We use a tree formation protocol similar to the one presented in 3.2, without the timeout and time-to-live mechanism, thus paths are created according to minimum hop distance and span in the whole network area.

When the root of a propagation tree disconnects from the sink, it simply caches all data, both generated and relayed, and waits to hear another beacon message with $h_c = 0$ to resume the propagation process. Another approach is to invalidate that tree and let the tree formation process reassign the nodes to a new tree, which will happen only if the sensor network is connected. However, this approach has a maximum hop distance equal to that of the static sink approach.

The kind of mobility presented here models situations where the mobile sink can't execute complex movements, movement on the terrain is possible only in certain areas or the mobile sink doesn't have enough energy to traverse the whole network area. The deployment of this protocol imposes a high cost on the sensor devices that perform tree formation and multi-hop propagation, however the delivery latency is lower than any of the three previous protocols.

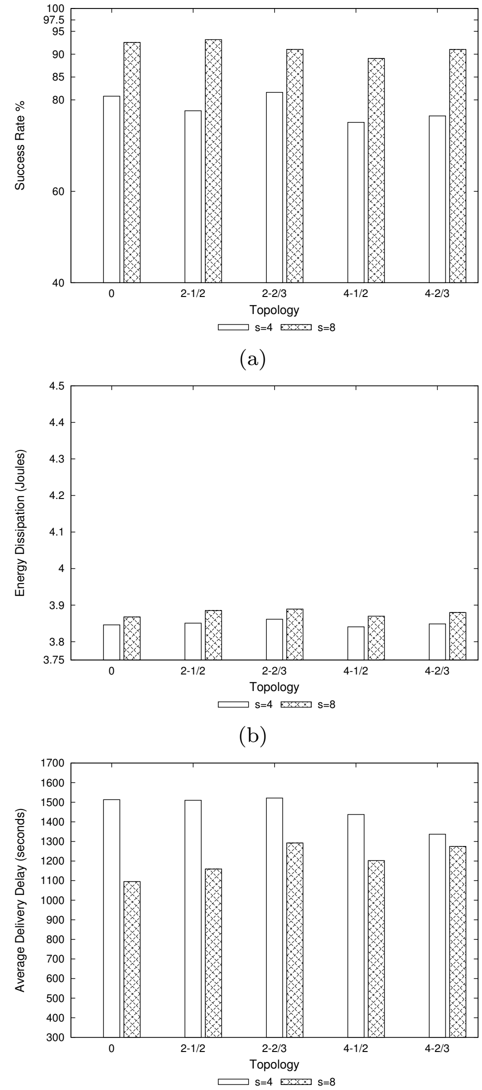
Furthermore, the selection of the trajectory length l introduces a trade-off between the cost at the sink and the cost at the sensors e.g. increasing l makes the sink move more and get closer to the sensors, thus reducing energy dissipation. Finally, it is interesting to investigate which shape (line or cycle) is best with respect to cost and performance (for equal shape lengths).

4. EXPERIMENTAL EVALUATION

We implement and evaluate the above discussed protocols in the network simulator platform $ns-2$ version 2.26. We also included for comparison a well known static multihop data propagation paradigm, Directed Diffusion (see [8]). We considered different simulation setups for various network sizes, number of particles and mobility parameters. We here present the results for the set of experiments that consider several network topologies. In particular the size of the network area is $200m \times 200m$, 300 sensor nodes are deployed. We consider a uniform deployment and deployment of nodes in $p_n \in \{2, 4\}$ pockets of radius $r_p = 2R$ whose positions are identical to the placement of dots on the facet of a dice. The ratio of the number of nodes in pockets over the number of nodes in the rest of the network is $\frac{n_p}{n_n} \in \{1/2, 2/3\}$. The transmission range of the sensors and the sink $\mathcal{S}$ is set to $R = 15m$ and the speed of the movement of $\mathcal{S}$ is set to $4m/s$ and $8m/s$.

We evaluate our protocols under different configurations, more specifically for $\mathcal{M}_{random}$ of P1, we set $d_{max} = 50m$. For P2 we examine 2 settings where $j = 10$, $ttl = 2$, $t_{thres} =$ 10sec and $j = 5$, $ttl = 4$, $t_{thres} = $ 10sec. For P3 we set $j = 20$ and present the results for various values of α and β, setting $\alpha = 1$ and $\beta = 0$, $\alpha = 0.5$ and $\beta = 0.5$ and $\alpha = 0$ and $\beta = 1$. For P4 we try different trajectory lengths, we set $l \in \{D, D/2, D/3\}$. For protocols P1, P2, P3 the mobile sink is initially positioned at point $(D/2, 0)$ and for all protocols it transmits a beacon message every $0.5sec$.

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 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 sink is collecting data and another $4000sec$ of simulation time are given in order to collect all the data, leading to $5000sec$ of simulation time.



(a) (b) (c)

Figure 1: Success rate (a), energy dissipation (b) and average delay (c) of protocol P1 for topologies with 0, 2 and 4 pockets with $\frac{n_p}{n_n} \in \{1/2, 2/3\}$

Conducting these experiments, we measure several metrics that depict the behavior of the protocols. We call *success rate* the percentage of data messages that were received to the sink 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 entity), as an absolute value in Joules. The *delivery delay* is defined as the time interval between the creation of a message and the time when it is delivered to the sink. Also, another set of metrics which is of particular interest for the experiments with mobility functions that traverse the whole network are the *covered set*, which is the percentage of the visited nodes (i.e. nodes that communicated with the sink directly or indirectly) over the total number of nodes. Note that due to lack of space, figures for the covered set metric where omitted, results are discussed in selected cases.

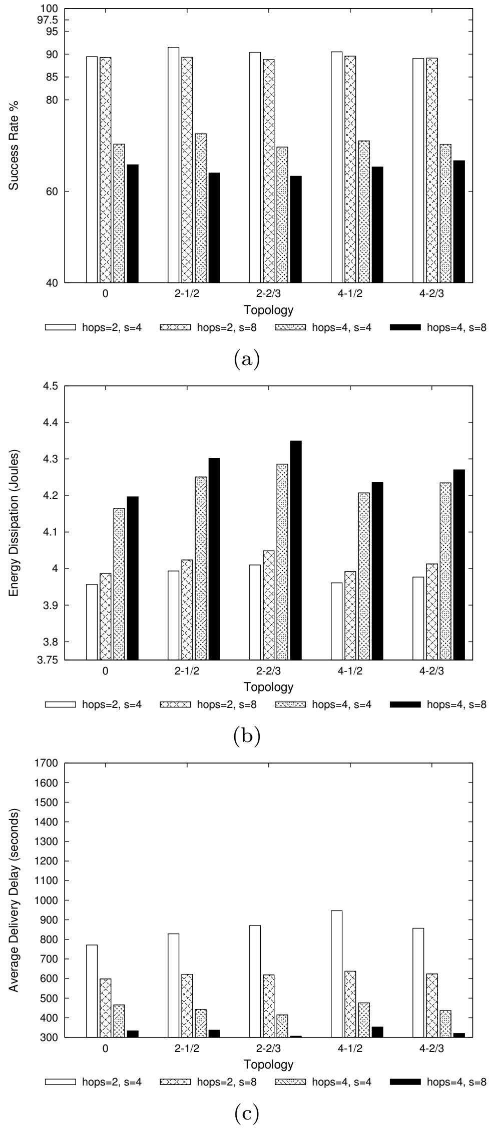


Figure 2: Success rate (a), energy dissipation (b) and average delay (c) of protocol P2 for different partitioning and hop limit, for topologies with 0, 2 and 4 pockets with $\frac{n_p}{n_n} \in \{1/2, 2/3\}$

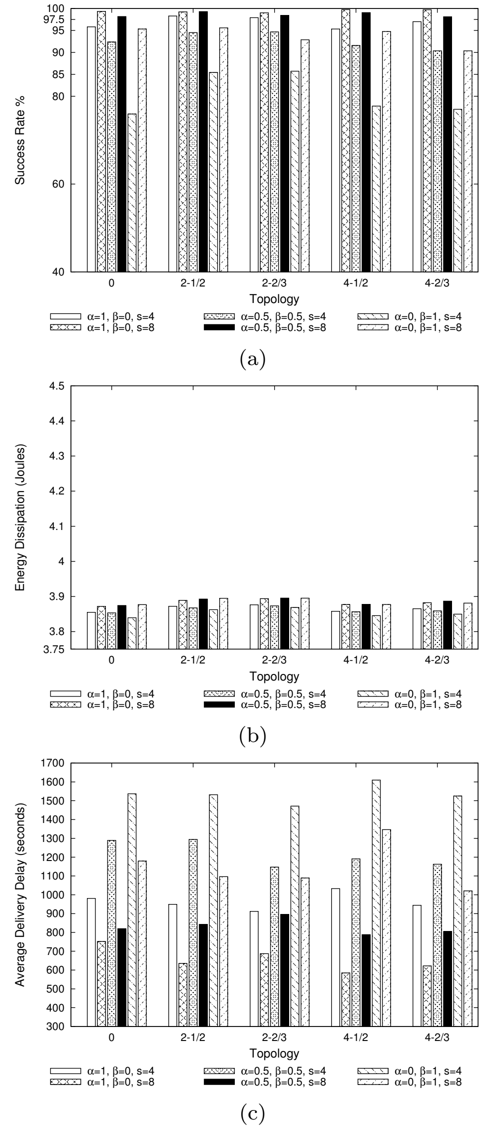


Figure 3: Success rate (a), energy dissipation (b) and average delay (c) of protocol P3 for different values of α, β for topologies with 0, 2 and 4 pockets with $\frac{n_p}{n_n} \in \{1/2, 2/3\}$

Figure 1 depicts the performance of protocol P1, we first observe that increased mobility speed s affects positively the performance of P1, since for $s = 8m/s$ it achieves success rate close to 95%, only a small increase in energy consumption, lower delay and almost full coverage of the network (95%–97%). When $s = 4m/s$ the covered set is about 85% meaning that about 15% of nodes remain unvisited, which explains the lower success rate. The protocol seems mostly unaffected from different topology settings which is expected given the nature of random walk.

In Figure 2 the experimental results for P2 are shown, it is striking that for a greater number of hops (when ttl=4) the performance drops significantly in terms of success rate and increased energy consumption. For 2 hops, P2 achieves high success rate (90%), compared to 60%–70% for 4 hops and low energy consumption around 4 Joules compared to 4.2–4.3 Joules for 4 hops. The delay in both cases is very low, especially when $ttl = 4$ which is almost 45% faster than the case of $ttl = 2$. In both cases P2 achieves full coverage ($> 97\%$) of the network, meaning that almost all nodes forwarded at least a packet to the sink. When s increases the success rate drops and the consumed energy increases, especially when $ttl = 4$, this is because nodes 1 hop away from the sink that relay traffic, don't have enough time to forward all messages to the sink and this results to re-transmissions and some packet loss. These results dictate that for proper selection of ttl, j and s, good trade-offs between success rate, energy consumption and delay can be achieved. When examining topologies with pockets we notice that the energy consumption and delay increase, especially for 2 pockets and when $\frac{n_p}{n_n} = 2/3$. In high density areas multihop protocols suffer from many collisions which explains our results.

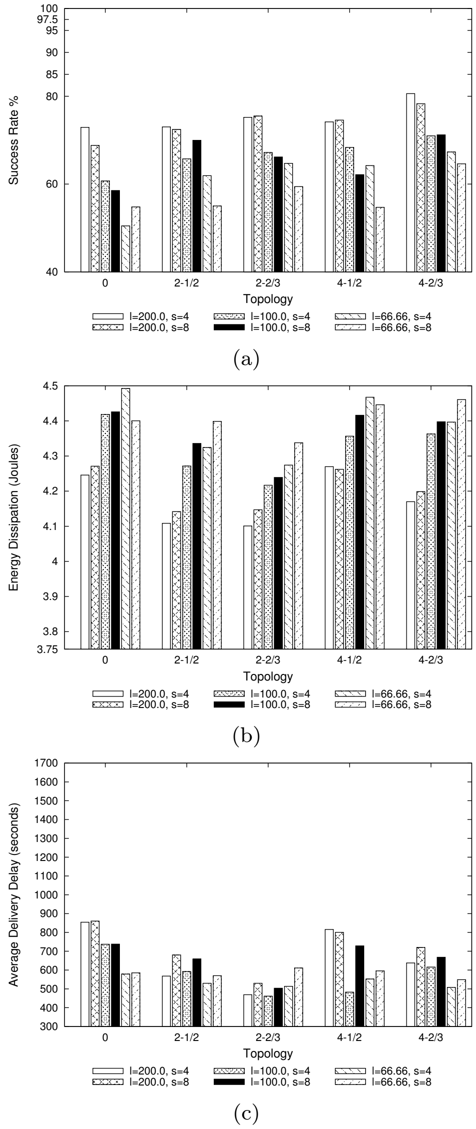


Figure 4: Success rate (a), energy dissipation (b) and average delay (c) for a circle trajectory of variable circumference l, for topologies with 0, 2 and 4 pockets with $\frac{n_p}{n_n} \in \{1/2, 2/3\}$

In Figure 3 we examine the performance of our biased protocol P3. We first notice that when the bias is based on the frequency of visits only ($\alpha = 1.0, \beta = 0.0$) the protocol performs very well, success rate reaches 99.7%, energy consumption and delay is low, full coverage is achieved. When the protocol is biased towards both frequency and density ($\alpha = 0.5, \beta = 0.5$) the protocol achieves good success rate (92%–98%) but slightly lower than the previous case, almost identical energy consumption and higher delay. When the protocol is biased towards density ($\alpha = 0.0, \beta = 1.0$) the protocol achieves success rate from (75%–95%), almost identical energy consumption to the previous cases, higher delay and lower coverage than the previous two cases. In all cases an increase in s results in improvement of the success rate and delay, especially when $\alpha = 0.0, \beta = 1.0$. These results show that when $\alpha = 1.0, \beta = 0.0$ the sink visits the whole network area fast and passes multiple times from each area thus collecting all the available data. On the contrary, when $\alpha = 0.0, \beta = 1.0$ the sink "stucks" around areas of relatively higher density and it takes a lot of time to visit new areas. When $\alpha = 0.0, \beta = 1.0$, the covered set is around 80-90% for $s = 4$ and about 95% for $s = 8$, the sink never visits all network areas. This can be also seen when there are 2 pockets in the network for $s = 4$, the sink stays close to the pockets and since the pockets contain many nodes it collects more data than in the case of 0 or 4 pockets when the bias is not that strong. When $\alpha = 0.5, \beta = 0.5$ an intermediate behavior is observed but the effect of the frequency bias appear to dominate, since the results are closer to the case of $\alpha = 1.0, \beta = 0.0$.

We examine the effect of deterministic mobility in Figure 4 for the case of a circular trajectory, figures for the case of a straight line show similar results and where omitted. We notice that the length of trajectory greatly affects the performance of P4. Both for line and circle trajectories when l decreases also the success rate decreases and the energy dissipation increases, since messages need to be propagated with more hops. Also the delay decreases when l decreases which is expected since the round trip time of the sink is reduced. Coverage is around 75%–90% thus whole areas of the network are disconnected from the sink. The effect of increased mobility speed is mostly negative since in most configurations success rate drops, while energy consumption and delay increases. This is because of reduced communication time of relay nodes to the sink in accordance to our observation when examining P2, as well as collisions. When examining different topologies, the behavior of P4 for a circle trajectory is better (with respect to consumed energy and success rate) when 2 and 4 pockets are present especially when $l = 200$ since the sink traverses regions near the pockets. Overall, P4 with the circle trajectory is better when $l = 200$ but as l decreases P4 with the line trajectory performs better.

In the final set of experiments we evaluate the overall advantages and disadvantages of our protocols (that exploit sink mobility) with "classic" multi-hop data propagation protocols (that assume a static sink). We select five different configurations of our protocols and compare them to a well known static multihop data propagation paradigm Directed Diffusion [8]. In this set of experiments we assume random uniform distribution of nodes (i.e. no pockets appear), the selected configurations are such that the success rate is maximum for the uniform topology and for the case of Directed Diffusion we fix the static sink at position $(D/2, 0)$. As shown in Figure 5, our protocols achieve much higher success rate (up to 140% higher) and significantly lower energy consumption (40%) than Directed Diffusion. This indicates that having a mobile sink (and by taking advantage of its mobility) can greatly extend the lifetime of the network. On the other hand, since the time-efficiency of our protocols is related to the mobility of the sink (speed, pattern), they suffer much higher delivery delays than Directed Diffusion that achieves a propagation delay of about 1.5 sec.

5. CONCLUSIONS AND FUTURE WORK

In this work we investigate the impact of having *a sink moving in the network area and collecting data*. We have presented a collection of mobility patterns and data collection strategies that can be employed in applications where

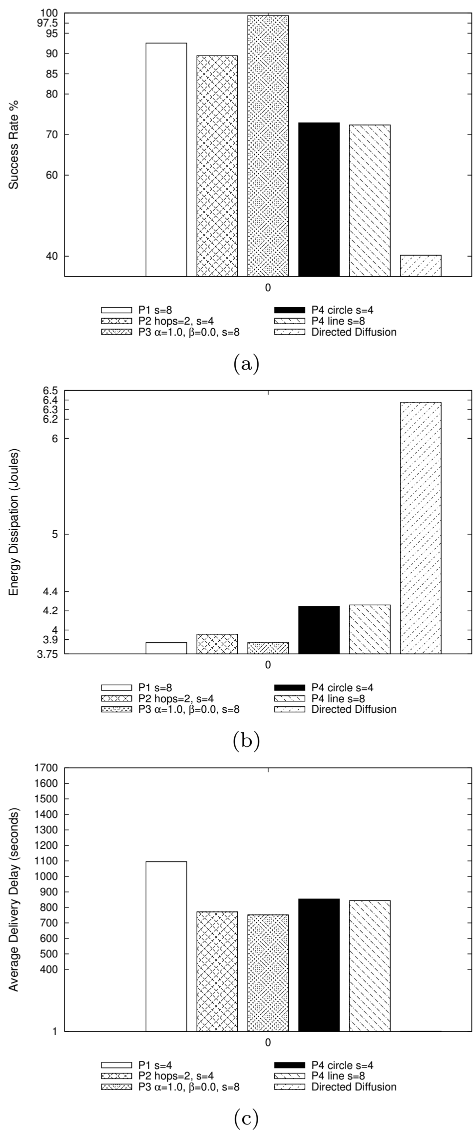


Figure 5: Success rate (a), energy dissipation (b) and average delay (c) for selected configurations of protocols P1, P2, P3, P4 and Directed Diffusion for topology with 0 pockets.

the sink is mobile (mostly related to ambient intelligence). Our experimental comparison demonstrates the relative advantages and disadvantages and different trade-offs achieved by each approach. Our results show that for applications where time efficiency is not critical it is better to let the sink suffer the burden of traversing the whole network area (as in protocols P1 and P3), since the greater energy savings are achieved in this way. By trading off some energy efficiency the delivery delay can be significantly reduced if a limited multihop approach is used as in protocol P2. For applications that the mobility capabilities of the sink are limited, but can tolerate some loss of information and increased energy consumption, the best approach is to follow a fixed trajectory with multihop data propagation.

We plan to further investigate the effect of the sink mobility on the performance of wireless sensor networks for different motion rates, network area sizes and particles density. Also we plan to look into different ways to perform a biased walk, possibly by also adjusting the movement speed and/or a walk over regular spanning regions of the original area. It is also interesting to further integrate the multihop approaches with the passive strategy leading to more efficient hybrid solutions.

6. 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. Pazzi, and R. Araujo. A supporting protocol to periodic, event-driven and query-based application scenarios for critical conditions surveillance. In *1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004)*, 2004.

[4] A. Boukerche, F. H. S. Silva, and R. B. Araujo. Context interpretation based wireless sensor networks for the emergency preparedness class of application. In *2nd International Workshop on Algorithmic Aspects of Wireless Sensor Networks(ALGOSENSORS 2006)*, 2006.

[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, S. Nikoletseas, 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).

[7] 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.

[8] 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.

[9] 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 (DCOSS05)*, pages 244–257, 2005.

[10] 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.

[11] 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.

[12] J. Luo and J.-P. Hubaux. Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. In *24th IEEE INFOCOM, Miami, USA, 2005.*

[13] C. Schindelhauer. Mobility in wireless networks. In *32nd Annual Conference on Current Trends in Theory and Practice of Informatics*, Czech Republic, January 2006.

[14] 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.

[15] M. Wattenhofer, R. Wattenhofer, and P. Widmayer. Geometric routing without geometry. In *12th Colloquia on Structural Information and Communication Complexity (SIROCCO 2005)*, pages 307–322. Lecture Notes in Computer Science (LNCS), Springer Verlag, Volume 3499, 2005.