Achieving High Lifetime and Low Delay in Very Large Sensors Networks using Mobile Sinks

Wint Yi Poe, Michael Beck, Jens B. Schmitt

Distributed Computer Systems Lab (DISCO), University of Kaiserslautern, Germany

***Abstract*—For smaller scale wireless sensor networks (WSN) it has been clearly shown that a single mobile sink can be very beneficial with respect to the network lifetime. Yet, how to plan the trajectories of *many mobile* sinks in *very large* WSNs in order to simultaneously achieve *lifetime* and *delay* goals has not been treated so far. In this paper, we delve into this difficult problem and propose a heuristic framework using multiple orbits for the sinks' trajectories. The framework is carefully designed based on geometric arguments to achieve both, high lifetime and low delay. In simulations, we compare two different instances of our framework, one conceived based on a load balancing argument and one based on a distance minimization argument, with a set of different competitors spanning from statically placed sinks to battery-state aware strategies. We find our heuristics to perform very favorably: both instances outperform the competitors in both, lifetime and delay. Furthermore, and probably even more importantly, the heuristic, while keeping its good delay and lifetime performance, scales well with an increasing number of sinks.**

I. Introduction

There is a growing trend for ever larger wireless sensor networks (WSN) consisting of thousands or tens of thousands of sensor nodes. For example, the WSN built by the GreenOrbs project at Zhejiang Forestry University for forest surveillance [6] employs $1000+$ nodes. We believe this trend will continue and thus scalability plays a crucial role in all protocols and mechanisms for WSNs. Another trend in many modern WSN applications is the sensitivity to the delay for the information transfer from sensors to sinks. In particular, WSNs are a central part of the vision of cyber-physical systems and as these are basically closed-loop systems many WSN applications will have to operate under stringent timing requirements. Hence, information transfer delays need to be controlled. On the other hand, since most WSNs are still based on battery-operated nodes, energy-efficiency clearly remains another premier goal in order to keep network lifetime high. How to achieve a lifetime prolongation by using mobile sink(s) to collect the data of a WSNs has already been investigated in many works [2, 3, 7, 8, 9, 10, 16, 17, 19, 20]. All of these leverage on the effect that the burden of being close to a sink is shared over time among all the nodes in the field. This alleviates the typical hot-spot problem, where nodes near the sink drain their battery much faster than others since they have to relay many data packets for other nodes. However, the effects on information transfer delay are either completely neglected or simply observed without taking actions to establish delay as an objective of equal importance to lifetime maximization. By using mobile sinks, in general, the information transfer delay from sensors to sinks increases. This is simply due to the fact that there is always a delay-optimal placement of the sinks and leaving it the message transfer delay becomes worse. This conflict between lifetime and delay shows that these two goals have to be carefully traded off against each other when planning the trajectories of multiple mobile sinks. In our work, we target very large WSNs and strive for the problem of finding good trajectories for multiple mobile sinks such that we keep the maximum message delay low and still achieve a long lifetime, which sets us apart from the current state-of-the-art.

The contributions of our paper are:

- To the best of our knowledge, we are the first to tackle the trajectory planning problem for multiple mobile sinks in very large WSNs under lifetime *and* delay goals.
- We derive a heuristic framework that keeps up its delay and lifetime performance in very large WSNs as long as a constant node to sink ratio is retained. (→Section III)
- A thorough simulative investigation and comprehensive comparison with alternative approaches inspired by literature is presented. (→Section IV)

Our work is based on using multiple orbital trajectories for the mobile sinks and segment the sensor field into a so-called polar grid. In each orbit the sinks are moved synchronously (e.g., once a day). As mentioned in [8, 10, 20] where the movement of a sink is abstracted as a sequence of a static sink placements assuming that the time scale of sink mobility is much larger than that of data delivery, we follow the assumption of slow mobility in our work. For the case of very large WSNs with many mobile sinks (say hundreds) this n-orbit model generalizes recent previous work of ours using only 2 orbits [12]. While we base on this work with respect to giving the problem a geometric interpretation, we remark that the n-orbit case is significantly harder. Most severely, the distribution of K sinks over n orbits leads to a combinatorial explosion of the search space for the values of K that we require in very large WSNs. Similarly, the optimal choice of the number of orbits n as well as the sizing of their radii become very difficult questions. We address these questions with a heuristic framework. It is built on a geometric reduction of the problem, where the two performance characteristics, delay and lifetime, are amalgamated into minimizing the Euclidean distance between nodes and sinks. The intuition behind this is that both, delay and lifetime, benefit from nodes being closer in terms of Euclidean distance to their assigned sinks.

II. Network Model and Problem Statement

Let V be the set of sensor nodes with $|V| = N$; S is the set of sinks with $|S| = K$. For both, N and K we assume large scales with N being on the order of thousands and K up to the order of hundreds. The reachability between nodes is modelled as a directed graph, $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = V \cup S$. For all $a, b \in \mathcal{V}$, the edge $(a, b) \in \mathcal{E}$ exists iff a and b are within a disc-based transmission range r_{tx}. The sensor field is assumed to be a circular area with radius R.

A. The Nodes

The nodes are i.i.d. uniformly distributed over the sensor field. We assume the node density (governed by the parameters R and N) to be high enough to ensure connectivity with high probability (see also Section IV-D). The nodes are homogeneous with respect to their initial energy E and their transmission range r_{tx}. Also, the costs for sending and receiving messages do not differ from node to node. The amount of data produced by each of the nodes is the same and follows the same traffic pattern, e.g., a periodic data generation. The energy consumption for operations other than receiving or transmitting can be neglected, since for homogeneous nodes they consume the same amount of energy. The nodes are stationary and use multi-hop-communication to send their data to their assigned sink. This means the routing topology is actually a forest of sink trees embedded in the reachability graph G. The assignment of nodes to sinks is further discussed in Section III.

B. The Sinks

The sinks are assumed to have no energy constraints. We assume that the sinks' movement is synchronous, i.e., all sinks move at the same time. Further, sink movement takes places on relatively long time-scales (e.g., once a day), much larger than the time-scale of the message transfer delay from sensors to sinks (e.g., on the order of seconds). Therefore, we neglect the time periods when the sinks are actually moving (or being moved) and the sink mobility is abstracted as a sequence of sinks' locations. At each location the sinks stay for an equal amount of time, further on called epochs. In particular, we also assume that all data is flushed from the WSN before a sink movement takes place, i.e., there is no data dependency between epochs.

C. The Problem Statement

In this setting, we want to simultaneously achieve a low information transfer delay and a long lifetime. Here, we define lifetime as the time until the first node of the network "dies", i.e., its battery is depleted. For the delay, we consider the worst-case message transfer delay for the whole network. To that end, we use sensor network calculus [13] to compute the worst-case delays for each data stream from a node to its sinks and take the largest of these worst-case delays as the worst-case delay for the whole network.

To solve this dual-objective problem one basically has to answer three questions in each epoch:

1) **Sink Trajectories**: where should the sinks be positioned?
2) **Sink Selection**: which sink does a node choose to send its data to?
3) **Sink-Tree Routing**: which route the data follows the sink?

In this paper, we focus on the planning of the sink trajectories and "hard-code" the other two questions: for sink selection, each node chooses its nearest sink with respect to Euclidean distance (within the same orbit, more details are given in Section III); for the routing we assume shortest path routing in the reachability graph G, mainly because it is a frequent case. Yet, even under these restrictions, the problem is still a very hard one (even strong reductions of it are NP-hard, see also [12] for a discussion of this). The main complexity has its roots in the conflicting objectives delay and lifetime. The end-to-end delay, as well as the energy consumption, is dependent on the path between nodes and their sink, as well as any other path interfering with this one. Hence there is a dependency structure between the end-to-end delays which is very hard to untangle.

III. Heuristic Framework

In this section, we present our heuristic framework for planning the sink trajectories in very large WSNs with delay and lifetime goals. Due to the complexity of the problem, we reduce it to a geometric problem. This abstraction is justifiable by the large scale of the WSN as we target it in our work. The rationale behind it is that the delay (mainly governed by the number of hops) needed to reach a sink is proportional to the Euclidean distance from the nodes to their sinks. On the other hand, per epoch, we divide the network area in a number of cells (using a polar grid segmentation of the circular sensor field), corresponding to the number of sinks. In each epoch, each node is assigned to the sink of its currently corresponding cell. Thus, we can abstract the load assigned to a single sink as the area of its cell. This geometric interpretation of load and delay is instrumental in constructing good sink trajectories, because instead of complex delay and energy functions we can now formulate the problem in terms of the size of cells and Euclidean distances between nodes and sinks, which are considerably simpler measures.

A. Orbital Sink Trajectories

Our heuristic framework is based on an orbital model for the sink trajectories in order to achieve a small distance between nodes and their sinks, as well as a balanced division of the network area into cells [12]. In a nutshell, it works like this: we conceive several circles around the center of the sensor field, with different radii, called orbits; the sinks are placed on these orbits with regular interspaces and revolve around the center, like satellites move around the earth (see Figure 1). For a more detailed presentation of this n-orbit model, we introduce some definitions and notations (see also Table I):

Table I
NOTATIONS FOR THE ORBIT MODEL.

R	Radius if the network area.
N	Number of nodes used.
K	Number of sinks used.
L	Leftover sinks.
n	Number of orbits used.
K_i; $i \in 1, \ldots, n$	Number of sinks placed in the i-th orbit.
R_i; $i \in 1, \ldots, n$	Radius of the i-th circle, constructing the polar grid.
d_i; $i \in 1, \ldots, n$	Maximal distance of one point in a cell of the i-th orbit, to the corresponding sink.
a_i; $i \in 1, \ldots, n$	The area of one cell in the i-th orbit.
θ	Movement angle of the sinks between epochs.
$\gamma \in [1, 2]$	For $\gamma = 1$ we get the MD-approach, for $\gamma = 2$ the EA-approach.

- We call the innermost orbit the first orbit and the outermost orbit the last or n-th orbit.
- By a sink distribution we refer to how many sinks are placed in each of the orbits; we denote the number of sinks placed in the i-th orbit by K_i.
- The orbits and their sinks divide the network area in a polar grid as illustrated in Figure 1. The cells within the same orbit have the same shape and size. The sinks are located in the center of their cells, such that they minimize the maximal Euclidean distance of any point of this cell to themselves. This center can be calculated by replacing the cell by a trapezoid (or triangle, in the case of the first orbit) sharing the same corner points as the cell and determining the center of the minimal enclosing circle of this trapezoid. A formal proof for the correctness of this intuitive statement can be found in [11].
- The polar grid consists of n concentric circles segmenting the sensor field into circular segments as well as ring segments. By R_i ($i \in 1, \ldots, n$) we denote the radii of the concentric circles, where R_1 describes the radius of the circular segments. The ring segments in the i-th orbit have an outer radius of R_i and an inner radius of R_{i-1}. The choice of R_i affects both, the number of nodes in a cell and the maximal distance from any point in the cell to the sink.
- By d_i, we denote the maximal distance of a point within a cell of the i-th orbit to its corresponding sink. Further, by a_i, we denote the area of a cell in the ith orbit.
- To preserve the polar grid structure, after each epoch, the trajectories are constructed by rotating all the sinks by the same angle θ around the center (thus a θ of $0°$ or $360°$ would result in no movement at all). More complicated trajectories are conceivable: the angles by which a single sink moves may be different from other sinks, even if they are in the same orbit and could change from epoch to epoch. Such trajectories would, however, not preserve the polar grid structure and be difficult to analyze. Since we are considering very large networks, there might be a practical upper bound on the angle θ the sink can move between epochs, simply by the limited distance a real mobile sink may move between epochs.

The orbit model is flexible, since one can choose different sink distributions and number of orbits and also form the

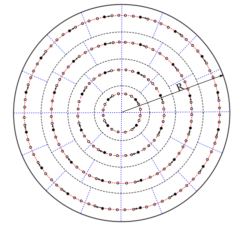


Figure 1. The n-orbit model.

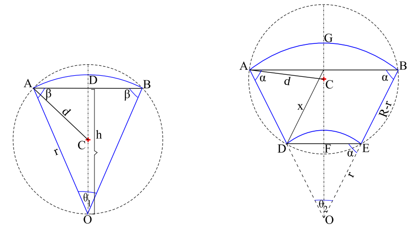


Figure 2. Illustration of the distance calculation.

cells by varying the radii R_i. Through this flexibility we are able to adress different goals like minimizing the overall Euclidean distance from any node to its sink or keeping the cells equally sized. Also the model scales naturally for an increasing number of sinks by simply increasing the number of orbits.

In the following we present two particular orbit models. The first has the goal of minimizing the maximum Euclidean distance from the nodes to their sink. The second has the goal of keeping the cells equally sized, while reducing the Euclidean distances as much as possible. Before we delve into the construction of these two orbit models we provide an overview about their construction. In a first step we found, by systematically searching the possible sink distributions and radii R_i, that these follow rather simple rules. In a second step, we search for the number of orbits, which results in the smallest maximal Euclidean distance. Up to this point, however, we handle the sinks, as if we could split them up and place them over several orbits, which is, of course, not possible. Hence in the last step, we distribute the sinks in such a way, that they get close to the formulations found in the first and second steps.

B. Distributing the Sinks and Sizing the Orbits

At first, we assume the number of orbits to be given and discuss the distribution of the sinks over the orbits (setting the K_i) and sizing the radii of the polar grid (setting the R_i). Based on this we compute the optimal number of orbits in the following subsection.

As mentioned above, we derive two types of orbital sink trajectories. The first has the goal of keeping the maximal Euclidean distance small. This goal however is hard to achieve, due to a very complex goal function and a large number of variables. The optimization problem can be formulated as follows (for a construction of it see [11]):

$$\min_{K_i:\, K_1+\ldots+K_n=K} \quad \min_{0 \leq R_i \leq R} \quad \max_{1 \leq i \leq n} d_i$$

with:

$$d_1 = d_1 = \begin{cases} R_1 \cos\beta & \text{if } K_1 = 3 \\ \frac{R_1}{2\sin(\frac{\pi}{2} - \frac{\pi}{K_1})} & \text{if } K_1 > 3 \end{cases}$$

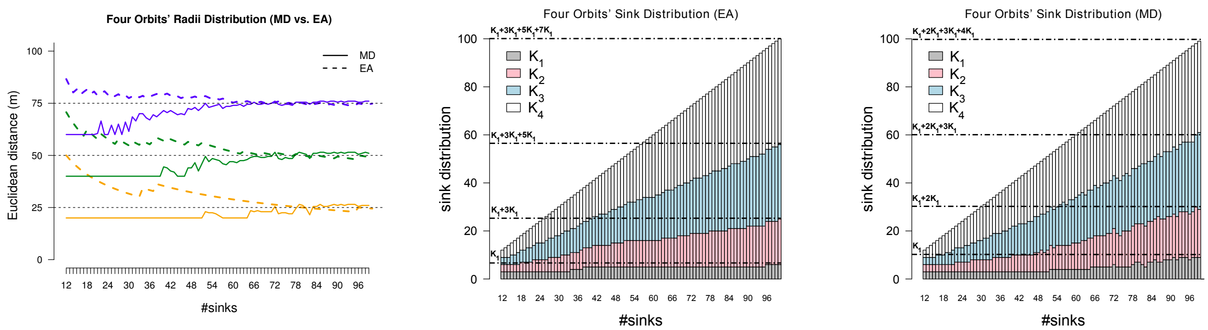


Figure 3. Radii and Sink distributions for MD and EA.

and for $i > 1$:

$$d_i = \begin{cases} R_i \cos \alpha_i & \text{if } R_i \cos \alpha_i \geq x \\ \frac{\sqrt{(R_i - R_{i-1})^2 + 4 R_i R_{i-1} \cos^2(\frac{\pi}{2} - \frac{\pi}{K_i})}}{2 \sin(\frac{\pi}{2} - \frac{\pi}{K_i})} & \text{if } R_i \cos \alpha_i < x \end{cases}$$

where α_i, respectively β, is the angle at A in the triangle ΔABO and x is the distance between D and the midpoint on the line between A and B (see Figure 2). To solve this problem, we have thoroughly explored the respective search spaces systematically, to find the best sink distribution and radii R_i. Due to the high-dimensional search space and a fine-grained sub-sampling of it this exploration involved a considerable amount of computational effort (several weeks of run-time on high-end PCs). The search does not only consist of the continuous parameters R_i which lie in $[0, R_{i+1}]$ (with $R_n \in [0, R]$), but also one has to consider a combinatorially growing amount of possible sink distributions (assuming orbits can be empty there are $\binom{K-n-1}{n-1}$ distributions).

Table II
COMPARISON OF MD AND EA METHODS.

	Minimum Euclidean Distance	Equal Sized Area
Sink distribution	$\frac{K_1}{K_i} = \frac{1}{i}$	$\frac{K_1}{K_i} = \frac{1}{2i-1}$
Orbits' radii	$R_i = i \cdot \frac{R}{n}$	$R_i = i \cdot \frac{R}{n}$
Relationship of K, K_1,and n	$K = K_1 \sum_{i=1}^{n} i$	$K = K_1 \sum_{i=1}^{n} 2i - 1$

The second type of orbital sink trajectory, which has the goal to obtain a small maximal Euclidean distance while keeping the cells equally sized, is a bit easier to tackle. The reason is, that we know, for a given sink distribution, how to choose the radii to achieve equal areas:

$$R_1^2 = \frac{R^2 K_1}{K} \quad \text{and} \quad R_i^2 = \frac{R_i^2}{K_1} \sum_{k=1}^{i} K_k$$

However, it is still unclear how to distribute the sinks optimally, such that a low maximal Euclidean distance is achieved. So we still have to deal with the combinatorial explosion of possible sink distributions. Also for this approach we decided to search systematically for the best solution.

The results of these computations for four orbits (for other numbers of orbits the results look similar) can be found in Figure 3. One can observe in the first figure that the radii converge to $R_i = i \cdot \frac{R}{n}$. As seen in Figure 3 the difference between the two approaches lies mainly in the sink distribution. They follow the rules presented in Table II, the dashed lines in the figures represent how the sink distribution, for 100 sinks, would have to look like, if one applies the rule.

Taking a closer look we recognize that both approaches are in fact member of the same familiy. Their sink distributions can be formulated in a generalized setting as:

$$K_i = K_1 + \gamma(i-1)K_1; \qquad K_1 = \frac{2K}{n(n\gamma - \gamma + 2)}.$$

For $\gamma = 1$ this results in the approach of minimizing the maximal distance (further called MD) and for $\gamma = 2$ we get the equally sized area approach (further called EA). There are other values for γ imaginable, resulting in hybrid approaches, however, we will not consider other values for γ in this paper. For the rest of the paper we set $R_i = i \cdot \frac{R}{n}$ to make further steps tractable.

C. Choosing the Right Number of Orbits

Clearly, choosing the right number of orbits is an important factor. In the smaller scale setting of our previous work [12] we found significant gains when going from a single orbit to a two-orbit trajectory. Hence, in the very large-scale WSNs that we target in this work, we have to find out which number of orbits is optimal. For this purpose, we compute for different number of sinks the optimal number of orbits by checking through all numbers of orbits from 1 up to 100 for both, MD and EA. How this computation was performed for a given number of sinks is shown in Algorithm 1. The algorithm takes as inputs, the number of sinks and the value of γ and outputs the number of orbits, which results in the smallest maximal Euclidean distance between any point and its allocated sink. The alert reader may notice that the algorithm takes only the first and last orbit into account for calculating the maximal Euclidean distance. This is due to the following lemma:

Lemma 1. *Let K and n be such that $3n(n\gamma - \gamma + 2) \leq 2K$, then d_i is increasing in i for all $i \geq 2$.*

Proof: (Sketch, a complete proof can be found in [11]) d_i can be given by:

$$d_i = \frac{R}{n} \Big(\frac{1}{4 \sin^2(\frac{\pi}{2} - \frac{\pi}{(\gamma i - \gamma + 1) K_1})} + (i-1) i \cot^2(\frac{\pi}{2} \text{-} \frac{\pi}{(\gamma i - \gamma + 1) K_1}) \Big)^{1/2}.$$

Algorithm 1 Computing the optimal number of orbits.

Inputs: The number of sinks K, the network radius R, and the parameter $\gamma = \begin{cases} 1 & \text{for } MD \\ 2 & \text{for } EA \end{cases}$.

for $n = 1$ to 100 do

1. Compute $K_1 = \frac{2K}{n(\gamma n - \gamma + 2)}$
2. if ($K_1 \geq 3$)

2.1. Compute $d_1 = \begin{cases} \frac{R}{2n\sin(\frac{\pi}{2} - \frac{\pi}{K_1})} & \text{for } K_1 > 3 \\ \frac{R}{n}\cos(\frac{\pi}{2} - \frac{\pi}{K_1}) & \text{for } K_1 = 3 \end{cases}$

2.2. Compute

$d_n = \frac{R}{n}\frac{1}{2\sin(\frac{\pi}{2} - \frac{\pi}{\gamma n - \gamma + 2})}\sqrt{1 + 4(n-1)n\cos^2(\frac{\pi}{2} - \frac{\pi}{\gamma n - \gamma + 2})}$

2.3. Compute $D^n_{max} = max\{d_1, d_n\}$

endfor

return orbit j such that $D^j_{max} = \min_{1 \leq i \leq n} D^i_{max}$

Table III

THE OPTIMAL NUMBER OF ORBITS FOR MD AND EA.

MD(#sinks)	EA(#sinks)	#orbits
3-8	3-11	1
9-18	12-29	2
19-35	30-59	3
36-59	60-98	4
60-90	99-146	5
91-127	147-200	6
128-170	-	7
171-200	-	8

Since d_i is positive, it is sufficient to show that d_i^2 has a positive derivative, after factoring out $\cos(\frac{\pi}{2} - \frac{\pi}{(\gamma i - \gamma + 1)K_1})\sin(\frac{\pi}{2} - \frac{\pi}{(\gamma i - \gamma + 1)K_1}) \geq 0$, its numerator (the denominator is positive) can be given by

$$(2i-1)\sin\left(\pi - \frac{2\pi}{(\gamma i - \gamma + 1)K_1}\right) - \frac{\pi\gamma}{(\gamma i - \gamma + 1)^2 K_1}\left(4(i^2 - i)\sin(\frac{\pi}{2} - \frac{\pi}{(\gamma i - \gamma + 1)K_1}) + 1\right).$$

using a Taylor-Expansion around π for the first sum and using that $\sin(x) \leq 1$ for all x, we know that the above expression is larger than zero, if

$$\frac{2\pi^3}{3i^2K_1^3} \leq 1$$

which is true for all $i \geq 2$ and $K_1 \geq 3$, which is the case by our assumptions on K and n.

The condition of the lemma is not restrictive, as, for sink distributions according to Table II, it translates to having at least three sinks per orbit. Having two or less sinks in one orbit would mean to place them in the center of the whole sensor field as this minimizes the Euclidean distance, effectively wasting a complete orbit. Hence, excluding such cases does not influence the best selection of orbits. The optimal number of orbits for different number of sinks (up to 200) and the different approaches are displayed in Table III. ■

D. Distribution of Leftover Sinks

In our rules for the sink distribution we handle the sinks as real numbers. Of course, they are integral and thus we simply use the following sink distribution:

$$K_1 = \left\lfloor \frac{2K}{n(\gamma n - \gamma + 2)} \right\rfloor; \qquad K_i = \lfloor K_1 + \gamma(i-1)K_1 \rfloor$$

This however may lead to the problem that we cannot distribute all of the K sinks, i.e., $L = K - \sum_{i=1}^{n} K_i \neq 0$ is the number of leftover sinks. We deal with this leftover sinks simply by distributing them over the orbits, in a greedy fashion, according to the goal of the respective approach. In the MD case, we place one sink at a time in the orbit which currently exhibits the maximal Euclidean distance, whereas in the EA case we place the sink in the orbit which contains the cells with the largest area.

IV. PERFORMANCE EVALUATION

In discrete-event simulations, we evaluate the delay and the lifetime performance of our heuristic framework by comparing it to three different mobile sink trajectories and two static sink placement strategies.

A. Delay Performance

With the help of sensor network calculus (SNC) [13], we evaluate the worst-case delay performance. To apply the SNC, the network traffic has to be described in terms of arrival curves α_i for each node. An arrival curve defines an upper bound for the input traffic of a node. To calculate the outputs of the nodes service curves β_i are used. The service curve specifies the worst-case forwarding capabilities of a node. Based on arrival and service curves, we use the Pay Multiplexing Only Once (PMOO) analysis described in [15] for the end-to-end delay bound computation (for this we use the DISCO Network Calculator [14]).

B. Lifetime Performance

We define the lifetime as the number of epochs until the first sensor node depletes its battery. The energy consumption for transmitting and receiving are taken into account using an energy model based on MICAz motes [1]. The model computes the energy level E_v^n of each sensor node v in epoch n using the following equations:

$$E_v^n = \sum_{w \in \mathcal{T}_v^n} E_{tx}(w, f_n(w)) + \sum_{w \in \mathcal{R}_v^n} E_{rcv}(w, f_n(w)), \quad (1)$$

with

$$E_{rcv}(w, f_n(w)) = E_{rcv}(f_n(w)) = P_{rcv} \cdot t_{rcv}(f_n(w)), \quad (2)$$
$$E_{tx}(w, f_n(w)) = P_{tx}(w) \cdot t_{tx}(f_n(w)). \quad (3)$$

Here, $\mathcal{T}_v^n$ and $\mathcal{R}_v^n$ denote the set of nodes to send to and receive from for v in epoch n. In (2), we see that the energy consumption for receiving $f_n(w)$, the amount of data from node w in epoch n, is just the time needed to receive the data $t_{rcv}(f_n(w))$ multiplied by the power consumption P_{rcv} of the receiving unit; this is independent of the distance between the sending and receiving node. In (3), the energy consumption for sending data is again the time needed to send the data $t_{tx}(f_n(w))$ times the power consumption of the sending unit $P_{tx}(w)$, which, however, now is dependent on the distance to w. Taking the values from the MICAz data sheet [1], we can calculate the power consumed by the receiver electronics P_{rcv} and the transmitting electronics $P_{tx}(w)$. The exact dependencies of $P_{tx}(w)$ on the distance to w is described by a model for the MICAz mote, which can be found in [18].

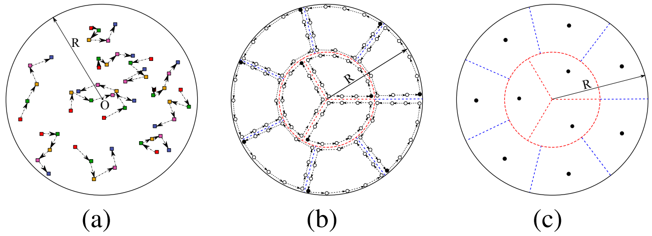


Figure 4. Competitors: (a) a random walk, (b) an outer periphery trajectory, and (c) a static MD.

C. Competitors

We have realized different competitors to compare our heuristic with. Unfortunately, the field of multiple mobile sink for very large WSNs is barely tapped so it was hard to find direct competitors. To create competitors we generalized ideas from other (smaller scale) proposals [4, 7, 9]. The competitors are briefly described in the following; some of them are illustrated in Figure 4.

Random Walk: Initially, sinks are placed uniformly randomly in the sensor field. At the start of each epoch, the sinks randomly choose a direction and step size (ensuring, however, that they do not leave the sensor field). We use this competitor as a baseline and also because it has been discussed in literature [3].

Outer Periphery: [7] remarks that, in the single sink case, a trajectory along the periphery of the network optimizes the lifetime by balancing the load distribution. We generalize this concept by moving each of our sinks along the cell peripheries, where the cells are formed according to the MD approach.

Following the Energy (FE): In this strategy, the sinks are placed randomly over the network area for the first epoch. For the following epochs, the K sensor nodes with the highest residual energy left are identified and the sinks move near to them. We use this one only as a competitor for lifetime, as its delay peformance is very bad. It is a simple representative of the group of state-aware trajectories (e.g. [9]).

K-Center Heuristic: [4] presents a polynomial 2-approximation for the NP-hard K-center optimization problem. The competitiveness of the algorithm is illustrated by the result of [5] which shows that if there exists an δ-approximation with $\delta < 2$ this results in $NP = P$. The authors use their algorithm on a fully connected weighted graph, nevertheless the idea can be carried over to our graph. This is a competitor only for the worst-case delay, as it performs badly with respect to lifetime due to being static. It serves as a representative for algorithms based on graph-theoretic abstractions and was expected to perform very well for delay due to its nice theoretical properties.

Static MD: This takes the same sink distribution as generated by our MD heuristic, but the sinks are not moving. Instead we run the MD strategy for a whole set of possible positions and choose the one, which has minimal delay. This obviously bad lifetime competitor is included to show both, how the lifetime of the network is increased by mobility as well as its negative effect on delay.

D. Experimental Set Up

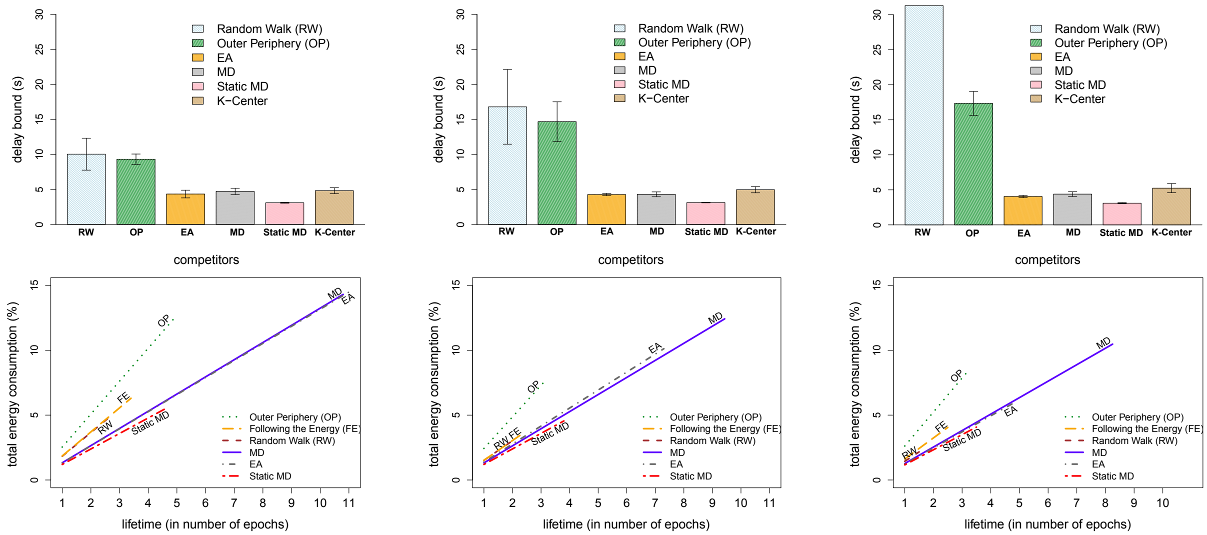
Using discrete-event simulations, we evaluate the worst-case delay and the lifetime performance of our heuristic framework. In the experiments, nodes are uniformly distributed over a circular field with radius R. The respective network radii are chosen such that always a node density of $\frac{1}{100\,m^2}$ is achieved. A $20\,m$ disc-based transmission range is used under a shortest path routing for the sink trees. Token-bucket arrival curves and rate-latency service curves are considered for SNC operations. In particular, for the service curve we use a rate-latency function that corresponds to a duty cycle of $1\,\%$ and it takes $5\,ms$ time on duty with a $500\,ms$ cycle length which results in a latency of $0.495\,s$[1]. The corresponding forwarding rate becomes $2500\,bps$. Initially, the nodes are set to an initial battery level of $0.1\,joule$. Packets of $100\,bytes$ length are sent to the corresponding sinks. Apart from static sinks, all others move synchronously to their next position between epochs. The MD and EA methods use a movement angle of $\theta = 10°$. To compute the energy consumption (Equation 1), we use the following data based on [1, 18]. The current consumption is $8.5\,mA$ with$-25\,dBm$ for distances up to $12.5\,m$, and $9.9\,mA$ for distances between $12.5\,m$ and $23\,m$ with $-20\,dBm$. For receiving a data packet, a $1\,\%$ duty cycle is considered with a current of $19.7\,mA$. A constant voltage of $3\,V$ is used. A transmission data rate of $250\,Kbps$ is used, which takes $t_{tx} = 3.2\,ms$ for a $100\,byte$ packet.

E. Results

We analyze the following three scenarios: 1500 nodes with 15 sinks, 5000 nodes with 50 sinks, and 10000 nodes with 100 sinks. So, we keep a constant node to sink ratio of 100 nodes/sink. For each scenario, we analyze the energy consumption per epoch, the lifetime and the worst-case delay. For all experiments, we performed 10 replications and present the average results from these. For the large majority of results, we obtained non-overlapping $95\,\%$ confidence intervals, so we do not show these in most of the graphs for reasons of legibility. The static MD and the K-center heuristic are static sink placements so that we compute the lifetime based on the overall number of packets transmitted and translate it into an equivalent number of epochs (using the results from the other methods).

1) Worst-Case Delay Evaluation: Figure 5 compares the delay performance of the four mobile sinks and two static sinks strategies. In all scenarios, the best delay performance is achieved by the static MD, closely followed by our mobile sinks strategies EA and MD. As already discussed in Section I, there is a price to pay for the prolonged lifetime by mobile sinks in terms of delay, yet as we see here that price is rather low. EA and MD perform almost equally well with a slight advantage for EA. More importantly, both of them achieve roughly the same delay performance across the different scenarios and are thus scalable with respect to delay.

[1]The values are calculated based on the TinyOS files CC2420AckLpl.h and CC2420AckLplP.nc.



(a) (b) (c)

Figure 5. Delay and lifetime performance comparison: (a) 1500 nodes and 15 sinks, (b) 5000 nodes and 50 sinks, and (c) 10000 nodes and 100 sinks.

For the outer periphery and the random walk, the assessment is very different: their delay performance is much worse and also the delay increases with growing network size, so they do not scale well with respect to delay. Somewhat surprisingly, the K-center heuristic, which requires a high computational effort and centralized information, is not doing particularly well and is actually slightly outperformed by the mobile trajectories EA and MD, which indicates again that their delay performance is very good.

2) Lifetime Evaluation: The simulation results for the lifetime performance of the competitors are shown in Figure 5. The graphs show the total energy consumption in the sensor field over the number of epochs, so the lengths of the lines indicates the lifetime performance of the respective method. Looking over all scenarios, MD turns out to be the clear winner with respect to lifetime. EA basically achieves the same lifetime in the 1500-nodes scenario, but cannot keep up with MD in the larger scenarios. All other competitors perform rather poorly: the random walk is a complete failure with a lifetime of 1.5 epochs in the largest scenario; the FE strategy also performs very bad and does not fulfil the hopes one could have in a state-aware trajectory (admittedly it is a simple strategy and more sophisticated state-aware trajectories could be doing better); the outer periphery strategy is a little bit better, but at the expense of a high overall energy consumption. Interestingly, the static MD does not perform too badly, it outperforms FE and the random walk, which shows that trajectory planning must be done with care otherwise one could do even worse than a good static strategy. On the other hand, we can see very clearly the lifetime prolongation effect of using mobile sinks when comparing static MD with the MD sink trajectory: for example, in the 1500-nodes scenario MD achieves 10.8 epochs whereas static MD achieves only 4.6 epochs.

3) Lifetime vs. Delay and Scalability: In this subsection, we somewhat wrap up the previous results by particularly looking at the combined lifetime vs. delay performance as presented in Figure 6. The x-axis represents the delay and the y-axis shows the corresponding lifetime performance in terms of the number of epochs. The shape and color of the symbols represents the different strategies and the size of the symbols encodes the scale of the experiment, i.e., the large symbols represent the experiments with 100 sinks, while the medium-sized and small symbols represent the experiments with 50 and 15 sinks, respectively. By following the path from small to large symbols one can see, how the strategies scale for larger WSNs. Clearly, the goal must be to stay within the upper left quadrant of this graph. Only MD achieves this goal, EA has a problem with respect to lifteime scalability. All other competitors do not really offer good lifetime-delay tradeoffs and are at best good in one of them.

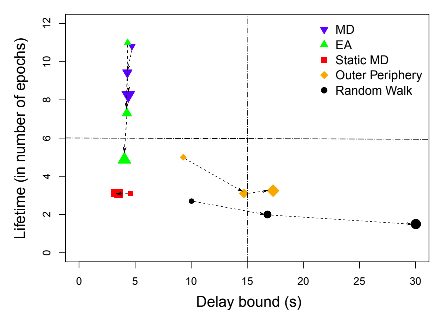


Figure 6. Lifetime-delay tradeoff among competitors.

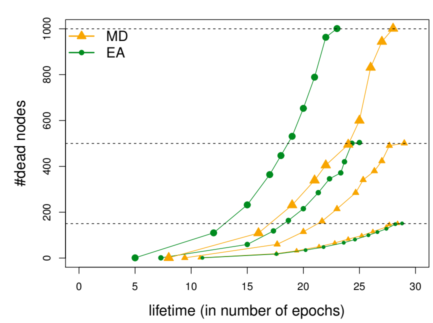


Figure 7. MD/EA lifetime performance under a different definition.

One may even become suspicious about MD for its scalability, because as can be observed in Figure 6, there is a certain degradation with respect to lifetime for it, too. However, the lifetime definition that we use here (when the first node dies) somewhat looses its usefulness with an increasing number of nodes, as it becomes more and more likely that some single node is in an unfortunate position where its battery is drained much quicker than for others. Therefore, we provide some more information on the "death" process of the nodes in the field when we continue network operation after the first node died in Figure 7 (again the size of the symbols represents the scale of the scenario). In particular, when we redefine lifetime as the time until which $10\,\%$ of the nodes have died then we see that MD scales very well, i.e., it achieves almost the same lifetime in all three scenarios. In comparison, EA still does not scale that well, though arguably it also benefits from this redefinition of the lifetime.

V. Conclusion and Outlook

In this paper, we have proposed a flexible heuristic framework to design the trajectories for multiple mobile sinks such that a good tradeoff between a high network lifetime and a low information transfer delay can be achieved in very large sensor networks. The framework uses an n-orbit model which is based on a geometric rationale that, in large sensor networks, cell areas and Euclidean distances between nodes and sinks are good proxy measures for lifetime and delay. Two instances of the framework are derived: one which focuses on the minimization of the maximal Euclidean distances (MD), and one which targets to equalize the area assignment and takes distance minimization as a secondary goal (EA). Both are compared with several competitors in detailed discrete-event simulations and show very good lifetime-delay tradeoffs. Especially, the MD strategy shows a very scalable behavior for its lifetime and delay performance when the number of nodes becomes large. In particular, in contrast to all other methods it keeps up the delay and lifetime values of smaller scenarios when scaled to larger scenarios under a constant node-to sink ratio. More abstractly, we believe to have provided strong evidence that the orbital sink trajectories provide for a natural scalability to very large sensor networks, if designed carefully.

To conclude the paper, we want to discuss some relaxations of the assumptions in our framework and how the geometric interpretation can still be helpful, thereby pointing out directions for future work: One assumption is the circular shape of the sensor field. Smaller distortions (linear transformations) of this would result in ellipsoid shapes, which can still be dealt with a similarly distorted orbit model (mapping the position of the sinks by the same transformation to the new sensor field). A change of the underlying distance norm would result in completely different shapes, we could think of a squared network area as a circle under the maximum norm $||\cdot||_\infty$ (similarly one could change to the $||\cdot||_1$-norm to handle a rhombus-shaped sensor field). Another assumption is the uniform distribution of nodes over the field. In some networks, there might be clusters of high density and regions with low density. Here more sinks are needed in the clusters, while the sparse areas can be handled by less sinks, this could be achieved by altering the distances the sinks move between the epochs in our orbit model. Slowing the sinks down, when reaching the clusters would result in accumulating sinks in that region and speeding them up again, when leaving the clusters, moves them fast through the sparse areas.

References

[1] Crossbow technology inc. mpr-mib users manual, June 2007.

[2] S. Basagni, A. Carosi, and C. Petrioli. Heuristics for lifetime maximization in wireless sensor networks with multiple mobile sinks. In *Proc. IEEE ICC*, June 2009.

[3] I. Chatzigiannakis, A. Kinalis, S. Nikoletseas, and J. Rolim. Fast and energy efficient sensor data collection by multiple mobile sinks. In *Proc. 5th ACM Int. Workshop. on Mobility Management and Wireless Access*, pages 25–32, 2007.

[4] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(3):180–184, May 1985.

[5] W-L. Hsu and G. L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1:209–215, 1979.

[6] Y. Liu, G. Zhou, J. Zhao, G. Dai, X.Y. Li, M. Gu, H. Ma, L. Mo, Y. He, J. Wang, M. Li, K. Liu, W. Dong, and W. Xi. Long-term large-scale sensing in the forest: recent advances and future directions of greenorbs. *Frontiers of Computer Science in China*, 4, 2010.

[7] J. Luo. *Mobility in Wireless Networks: Friend or Foe - Network Design and Control in the Age of Mobile Computing*. PhD thesis, School of Computer and Communication Sciences, EPFL, Switzerland, 2006.

[8] J. Luo and L. Xiang. Prolong the lifetime of wireless sensor networks through mobility: A general optimization framework. *In Theoretical Aspects of Distributed Computing in Sensor Networks*, pages 553–588, June 2010.

[9] M. Mudigonda, T. Kanipakam, A. Dutko, M. Bathula, N. Sridhar, S. Seetharaman, and J.O. Hallstrom. A mobility management framework for optimizing the trajectory of a mobile base-station. In *Proc. EWSN*, 2011.

[10] I. Papadimitriou and L. Georgiadis. Maximum lifetime routing to mobile sink in wireless sensor networks. In *Proc. SoftCOM*, 2005.

[11] W. Y. Poe, M. Beck, and J. B. Schmitt. Achieving High Lifetime and Low Delay in Very Large Sensors Networks using Mobile Sinks. Tech. Report 385/11, University of Kaiserslautern, Germany, July 2011.

[12] W. Y. Poe, M. Beck, and J.B. Schmitt. Planning the Trajectories of Multiple Mobile Sinks in Large-Scale, Time-Sensitive WSNs. In *Proc. IEEE DCOSS*, 2011.

[13] J. B. Schmitt and U. Roedig. Sensor Network Calculus - A Framework for Worst Case Analysis. In *Proc. DCOSS*, June 2005.

[14] J. B. Schmitt and F. A. Zdarsky. The DISCO Network Calculator - A Toolbox for Worst Case Analysis. In *Proceeding of the First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'06)*. ACM, November 2006.

[15] J. B. Schmitt, F. A. Zdarsky, and L. Thiele. A Comprehensive Worst-Case Calculus for Wireless Sensor Networks with In-Network Processing. In *IEEE Real-Time Systems Symposium (RTSS'07)*, pages 193–202, Tucson, AZ, USA, December 2007.

[16] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. In *Proc. IEEE SNPA*, pages 30–41, 2003.

[17] Y. Shi and Y. T. Hou. Theoretical Results on Base Station Movement Problem for Sensor Network. In *Proc. IEEE INFOCOM*, Phoenix, AZ,USA, 2008.

[18] R. Shokri, P. Papadimitratos, M. Poturalski, and J. P. Hubaux. A Low-Cost Method to Thwart Relay Attacks in Wireless Sensor Networks. Proj. Report IC-71, 2007.

[19] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava. Controllably Mobile Infrastructure for Low Energy Embedded Networks. *IEEE Transactions on Mobile Computing*, 5:958–973, 2006.

[20] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. In *Proc. 38th Hawaii Int. Conf. on System Sciences*, January 2005.