Biased Sink Mobility with Adaptive Stop Times for Low Latency Data Collection in Sensor Networks

Athanasios Kinalis*, Sotiris Nikoletseas*, Dimitra Patroumpa*, Jose Rolim†
*University of Patras and Computer Technology Institute, Patras, Greece
†Centre Universitaire d' Informatique, Geneva, Switzerland

***Abstract*—Collecting sensory data using a mobile data sink has been shown to drastically reduce energy consumption at the cost of increasing delivery delay. Towards improved energy-latency trade-offs, we propose a biased, adaptive sink mobility scheme, that adjusts to local network conditions, such as the surrounding density, remaining energy and the number of past visits in each network region. The sink moves probabilistically, favoring less visited areas in order to cover the network area faster, while adaptively stopping more time in network regions that tend to produce more data. We implement and evaluate our mobility scheme via simulation in diverse network settings. Compared to known blind random, non-adaptive schemes, our method achieves significantly reduced latency, especially in networks with non-uniform sensor distribution, without compromising the energy efficiency and delivery success.**

I. Introduction

The collected data in wireless sensor networks is usually disseminated to a *static* control center (called data sink) in the network, using node to node – *multi-hop* data propagation. Such settings have increased implementation complexity and sensor devices consume significant amounts of energy, since a distributed routing protocol for disseminating data towards the sink is executed in each sensor node. Also, in the area around the control center, nodes need to heavily relay the data from the entire network, thus a hotspot of increased energy consumption emerges and failure, due to strained energy resources, of these nodes leads to a disconnected and dysfunctional network [1]. Towards a more balanced and energy efficient data collection sink mobility can be used.

A. Sink Mobility: Opportunities and Challenges

In recent years, a new category of important sensor networks applications emerges, where motion is a fundamental characteristic. In such applications sensors may be attached to vehicles, animals or people that move around large geographic areas, while robotic elements may be present as well. Data exchange between individual sensors and infrastructure nodes will drive applications such as traffic and wild life monitoring, smart homes and pollution control.

Motivated by these developments, a new approach has been introduced that shifts the burden of acquiring the data, 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 region the sensor network is deployed, in close proximity to a (usually small) subset of the sensor devices, collecting the recorded data from the sensor nodes at very low energy cost.

Partially supported by the IST Programme of the European Union under contract number IST-2005-15964 (AEOLUS).

This data collection paradigm has many attractive properties. A mobile agent that moves closer to the nodes can help conserve energy since data is transmitted over fewer hops. Connectivity of the network is not required, thus sparse networks can be better handled, and additionally, fewer sensor devices may be deployed, to reduce 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, thus saving energy.

However, many apparent difficulties arise as well since traversing the network in a timely and efficient way is critical. Failure to visit some areas will result in data loss, while infrequently visiting some regions will result in high delivery delays. Also, routing and localization problems in the case of mobile sinks become more difficult to cope with.

Additionally, critical issues arise in node to sink communications. For single hop communication, the sink should eventually come within range of every sensor. Furthermore, to successfully complete the communication, it must remain within the range of the transmitter for the entire period that the message is transmitted. This problem can be severe when there is high density of sensors in an area or when some sensors have recorded a significant amount of data. In such cases, the communication time between the nodes and the sink is not enough to upload their data, thus they need to wait for the sink to return. This results in high delivery delays or even data loss when the nodes have limited buffers. This problem can be mitigated or even completely eliminated if the sink pauses the network traversal in order to collect the data. In our work we investigate sink mobility strategies that follow this approach, since we introduce adaptive stop times which are proportional to the local data traffic.

B. Our Contribution

We propose biased sink mobility with adaptive stop times, as a method for efficient (with respect to both energy and latency) data collection in wireless sensor networks. We assume a weak model of a single mobile sink and propose a strategy for network traversal, which serves nodes in a balanced manner. The traversal is performed on a per region basis: the sink visits regions one after another, stopping at each region for an appropriate interval to collect data.

When moving in a random manner, we propose an efficient biased random choice method that favors less visited and more dense areas. Also, our method locally determines the stop time needed to serve each region with respect to some global network resources. More specifically, we estimate an upper bound for the available total pause time, based on the initial energy reserves of the nodes and hence the expected lifetime of the network. We disperse the total pause time, based on local, at each region, criteria, stopping for a greater time interval at regions with higher density, and hence more traffic load. In this way, we achieve accelerated coverage of the network as well as fairness in the service time of each region. Besides randomized mobility, we also propose an optimized deterministic trajectory without visit overlaps, including direct (one-hop) sensor-to-sink data transmissions only.

We evaluate our methods via simulation, in diverse network settings and comparatively to related state of the art solutions. Our findings demonstrate: a) for both network traversal methods (e.g. the randomized and deterministic) the introduction of stop times (both constant and adaptive) reduces latency a lot, while keeping high (or even increasing) the delivery success rate, and also reducing the energy consumption, b) especially in the case of adaptive stop times, the latency improvements are very significant; in fact, our adaptive random walk outperforms the optimized deterministic traversal.

II. Related Work and Comparison

In mobile settings, the protocols and findings of previous research on static wireless sensors networks can not be (at least directly) applied. Efficient solutions in the state of the art become inefficient or even inoperable. Even well studied algorithms need to be redesigned; as an example, [2] proposes a leader election algorithm suitable for mobile networks. Also, new problems arise due to the high dynamics, e.g. maintaining system integrity [3] and security [4] becomes more difficult.

Recently, applications that motivate mobility in wireless sensor networks appeared. For networks of highly mobile sensors (e.g. when the sensors themselves move dynamically) [5] proposes adaptively redundant data dissemination strategies as replacement for connectivity e.g sensors moving fast throughout the network are favored for message ferrying, while less data is forwarded under high mobility.

Relevant research is also presented in [6]. The authors propose and evaluate experimentally two algorithms for adaptive movement of a data sink, that moves back and forth on a straight line at given speed levels. The algorithms achieve low energy consumption and satisfy an arbitrary time constraint, set by the network operators, for the round trip time of the sink. In our work we examine the problem in the more general case where the sink moves either deterministically or randomly in a two dimensional setting. We also impose stops on the sink movement but the total stop time, and hence the cover time, is constrained by the energy reserves of the nodes. We examine equally distributed, fixed stop times as well as adaptive ones, based on the observed local density in the area where the sink is visiting. Our approach aims at reducing the overall delay in collecting the data from the network.

Another relevant idea is presented in [7]. In a two dimensional setting, multiple mobile sinks that move with constant speed on precomputed linear trajectories are examined. Our work is also based on a two dimensional setting but we assume a single sink only and propose mobility methods that cover the whole network area in both deterministic and randomized ways. Also, we propose adaptive movement in order to better handle areas with increased load.

The effects of docking the sink in different points in the network are investigated in [8]. The authors propose a multihop routing scheme where a single sink stops at certain anchor positions while collecting data. Our protocols assume only knowledge of the initial energy reserves of the sensor nodes and operate using only local knowledge; we do not acquire any global knowledge about the network conditions; also, we focus on minimizing latency.

Finally, in [9], [10] the authors present methods for scheduling the movement of mobile elements in a sensor network to avoid data loss due to buffer overflow.

III. The Model

Sensors in our model do not move. The network area $\mathcal{A}$ is a flat square region of size $D \times D$. 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*. However, in several important scenarios the network operators 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 with high sensor node 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 $n_P = d_P \cdot \mathcal{A}_P$. Likewise, the number of sensors contained in the rest of the network is $n_n = d_n(\mathcal{A} - \mathcal{A}_P)$. Thus the total number of sensors is $n = n_P + n_n$.

Sensor devices are equipped with hardware monitors that measure environmental conditions of interest. Each device has a *broadcast* (digital radio) *beacon mode* of fixed transmission range $\mathcal{R}$, 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 $\mathcal{E}_i$ the *initial available energy supplies* of sensor i. 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, $E_R(k) = \epsilon_{recv} \cdot k$, where $\epsilon_{trans}, \epsilon_{recv}$ are constants that depend on the radio module and the transmission range $\mathcal{R}$.

For the idle state, we assume that the energy consumed for the circuitry is constant for each time unit and equals *eidle*.

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. Here, we assume that the sink is *mobile*. 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 accurately calculate its position (e.g. by using navigational equipment, such as GPS) and is aware of the dimensions and boundaries of the network area.

Finally, we assume that a specific, high-level, application is executed by the sensors. This application is characterized by the same message generation rate λ, for all sensors . The approach of the uniform message generation rates, however, is meant to address the following: **a)** In some applications it is $\lambda_i = \lambda$, indeed and traffic is captured by the heterogeneous density **b)** even if λ_i's are different, we may not know their exact values but only an average; even if known, we may take λ as a gross, average value in each neighborhood, since the sink each time adapts on all sensors it can listen to (see IV.*B*).

IV. Data Collection with Adaptive Stopping Times

As the mobile sink traverses the network, the subset of sensors that it is able to communicate with changes very frequently. To collect all the recorded data, the sink may need to visit each sensor several times, since data is generated continuously and not all of the data may be collected in a single session with a sensor. In the latter case, this means that some data will be delivered extremely delayed since the sink will have to roughly complete a trip around the network before visiting again. This effect can be mitigated if the sink pauses to collect more data where available. The approach we follow here is focused on two complementary directions. First, we suggest efficient ways to traverse the network area and then we propose efficient methods to determine for how long should the sink remain in each region.

A. Network Traversal

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, called cells. The center of each cell is considered as a vertex in a graph that is connected with unidirectional edges only to the four vertices corresponding to adjacent cells. Thus, a lattice graph $G_o = G(V, E)$ is created which is overlayed over the network area as depicted in Figure 1. We set $j = \lceil D/\sqrt{2}\mathcal{R} \rceil$, thus when the sink is located at the center of a cell, it can communicate with every sensor node within the cell area. By reducing the walk in an overlay graph we can perform some optimizations more easily; also, our mobility schemes can be deployed in areas of arbitrary topologies as long as we can abstract them by an overlay graph. Initially, the mobile sink is positioned on one of the nodes of G_o. We define two traversal methods of G_o.

Deterministic Walk. In order to cover the whole network area in an efficient way we first suggest a deterministic walk that spans the entire network. The sink visits cells from left to right and vice versa when a boundary is reached, thus forming a trajectory seemingly composed of connected horizontal linear segments as seen in Fig. 1. By moving on this trajectory, the sink will be, at some point in time, within the communication range of each node in the network. This walk assumes some global network knowledge at least at some stage (e.g. the sink knows the boundaries of the network) and since it avoids visit overlaps and multihop communication it represents some kind of optimality with respect to the time needed to cover the network and the energy spent.

Biased Random Walk. However, in the general case it may not be feasible to traverse the network in a deterministic way as the one we already presented. The presence of obstacles may hinder the movement of the sink while it may be desirable to move in an unpredictable way to avoid mischiefs from adversaries. Also, the network topology may not be known to the sink or may change dynamically. We thus propose a form of random walk that uses probabilistic transitions between the cells. The next position of the sink is determined by selecting (with some appropriate probability) one of the neighbors of the current vertex/cell in the graph.

In this walk, the sink associates a counter c_u for every vertex u; initially $c_u = 0 \ \forall u \in V$. When the mobile sink enters the area corresponding to vertex u it increases the associated counter c_u by 1. Thus, the frequency of visits of each area can be estimated and maintained by the sink. The selection of the next area to visit is done in a biased random manner depending on this variable. 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 probability $p(f)_v$ 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 regions are favored when the sink is located at a nearby region.

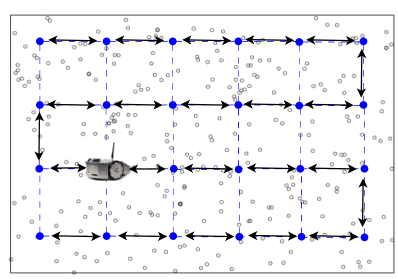


Fig. 1. Example of the overlay graph G_o and deterministic sink movement

B. Calculation of Stop Time

For each one of the network traversal methods (deterministic or random) proposed in the previous subsection, we additionally introduce stop times i.e. as the sink traverses the overlay graph G_o, it pauses its movement in each vertex of the graph (i.e. in each network region) for a certain amount of time. In particular, we propose constant (the same for all vertices) and adaptive stop times (the stop time depends on some local parameters e.g. density), and also compare them to the non-stop movement (e.g. the stop time is zero at each vertex). We present these schemes in next paragraphs.

First, in order to decide the stopping times for each case, we define the variable T_{total_stop}, which represents the maximum total amount of time the sink can remain stationary. T_{total_stop} is calculated as follows:

$$\begin{aligned} E_{total} &= \text{total\_messages\_sent} \cdot E_{msg} \\ &= n \cdot \lambda \cdot T_{total\_stop} \cdot E_{msg} + E_{idle} \end{aligned} \tag{1}$$

where E_{total} is the total initial energy of all the sensors in the network, n is the number of sensors of the network and λ is the average event generation rate. Assuming that the maximum length of a message sent is bounded, let E_{msg} be the maximum amount of energy consumed when sending a message. In this way, we calculate the maximum stop time the sink can use until the whole energy of the network is depleted. It is:

$$E_{total} = n \cdot \mathcal{E}_i = d \cdot \mathcal{A} \cdot \mathcal{E}_i$$

where $\mathcal{A}$ is the size of the network area, d the density of sensor deployment and $\mathcal{E}_i$ the initial energy of each sensor i. Also

$$E_{idle} = eidle \cdot n \cdot T_{sim} - eidle \cdot n \cdot T_{total\_stop}$$

where $eidle$ is the energy spent when the sensors remain idle, since $E_{idle} = n \cdot T_{idle} \cdot eidle$ and $T_{idle} = T_{sim} - T_{total\_stop}$, where T_{idle} is the total time the sensors remain idle and T_{sim} is the total time that the experiment is performed. Then

$$\begin{aligned} T_{total\_stop} &= \frac{E_{total} - eidle \cdot n \cdot T_{sim}}{n \cdot \lambda \cdot E_{msg} - eidle \cdot n} \\ &= \frac{d \cdot \mathcal{A} \cdot \mathcal{E}_i - eidle \cdot n \cdot T_{sim}}{n \cdot \lambda \cdot E_{msg} - eidle \cdot n} \end{aligned}$$

That is, T_{total_stop} is a function f of d, $\mathcal{A}$ and λ: $f(\frac{d \cdot \mathcal{A}}{\lambda}) = f(n/\lambda)$. We assume that the algorithm evolves in r rounds, where *round* is the time needed for the sink to visit all the nodes. This interval is stored in the variable T_{round}. If the total stopping time is equally shared among the rounds, then for each round $T_{total\_stop\_round} = \frac{T_{total\_stop}}{r}$. Thus, $T_{total\_stop\_round}$ is the maximum amount of time that the sink will remain static in each round. For the first round, from standard random walk theory, the value of T_{round} is $number\_of\_cells^2 \cdot a$, where a depends on the distance between the centers of two adjacent cells and the speed of the sink (in our case it is $a = (D/j)/s_{max}/10$). After the first round, T_{round} takes its actual value. We now propose two different types of stopping:

Constant stop time. Here we disperse $T_{total\_stop\_round}$ equally in each cell. The constant time the sink pauses at each cell *i* is:

$$T_{cell_i} = \frac{T_{total\_stop\_round}}{number\_of\_cells}$$

Adaptive stop time. Here we *adaptively* calculate the stopping time, with respect to the local density of the network and thus the expected traffic in the cell. Let T_{adap_i} the adaptive pause interval for cell i:

$$T_{adap_i} = \frac{d_i}{d} T_{cell_i}{}^{1}$$

where d_i is the local density in cell i and $0 \leq \frac{d_i}{d} \leq \beta$ (β can be > 1).

In all cases, data is collected in a passive manner. The sink periodically broadcasts beacon messages. Nodes that receive a beacon start transmitting the data stored in their memory to the sink. The maximum time the sink can stay in cell *i* is T_{adap_i}. The messages the nodes send to the sink include a flag reporting whether their memory emptied or not. If the sensors empty their memory before T_{adap_i} expires, the sink leaves the cell, to avoid spending a lot of time in an area without collecting data. Otherwise, it leaves at the end of T_{adap_i}, even if there is more data in the sensors' memory to be sent.

Also, at each cell, the sink waits for messages to arrive for a submultiple of T_{adap_i}, $\frac{T_{adap_i}}{c}$; if no messages arrive during that time it leaves the cell (in particular, we take *c = 10*). Especially for the case of deterministic network traversal, after the first round the sink does not stop at all in a cell where no nodes were discovered.

V. Performance Evaluation

We implement our schemes in $ns-2$ [11], using the extension framework **TRAILS** [12] which aims at better handling networks of high dynamics due to e.g. mobility, failures and obstacles. In addition to the protocols we propose, we implemented two well known schemes, for comparison. The first is a protocol without stop times, that in the case of random movement is the classic random walk on the overlay graph G_o. Also, we adapted, to operate in our two dimensional setting, and implemented **SCD** (Stop to Collect Data), one of the algorithms proposed in [6]. In **SCD**, the mobile sink stops when receiving new data, for an interval inversely proportional to the number of newly discovered sensor nodes.

A. Simulation Setup

We considered different simulation setups for various network settings and mobility parameters. We here present the results for the set of experiments that consider several mobility parameters. In particular, the size of the network area is $200m \times 200m$ and 300 sensor nodes are deployed. We consider a deployment of nodes in one pocket of radius r_p = $2\mathcal{R}$. The ratio of the number of nodes in pockets over the number of

[1] $\sum T_{adap_i} = \frac{T_{cell_i}}{d} \sum d_i = \frac{T_{cell_i}}{d} \cdot d \cdot number\_of\_cells = number\_of\_cells \cdot T_{cell_i} = T_{total\_stop}$, i.e. the stop times sum up to the total stop time as they should.

nodes in the rest of the network is $\frac{n_p}{n_n} = 9$. The transmission range of the sensors and the sink is set to $\mathcal{R}$ = 15m.

We evaluate our methods considering different values for the speed of the mobile sink, $s \in \{4, 8, 10, 20\}$m/s. The initial energy reserves of the nodes are 5.5 Joules. The values of ϵ_{trans}, ϵ_{recv} and *eidle* were set to match as close as possible the specifications of the mica mote platform. We assume a high level periodic monitor application executed by the sensors; the application is triggered at the beginning of the simulation and registers data about the network region. The data is generated at random times at an average rate of 1 message/10 sec. Each sensor device transmits 100 messages before the monitored phenomenon ends and the simulation lasts 7000*sec*.

Conducting these experiments, we measure several metrics depicting the behavior of the protocols. We call *success rate* the percentage of data messages received by 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. *Latency* is defined as the time interval between the creation of a message and the time it is delivered to the sink.

For each result we executed 100 runs. The findings demonstrated high concentration around the average values, which are depicted in the figures below.

B. Findings

The performance of the methods we proposed when the network traversal is performed deterministicly is shown in Fig. 2. In Fig. 2a, we observe that increased mobility speed affects positively the performance of both the adaptive and the constant schemes, since when speed increases, the latency of the adaptive scheme is reduced, while the latency of the constant scheme initially reduces and then remains almost the same. On the contrary, increased speed leads to increased latency in the case of the deterministic walk with no stop times. The high speed in the case of the adaptive scheme balances the effect of stopping in different network regions, thus it achieves better latency. The increase of speed is beneficial for the **SCD** algorithm, too, since when speed increases its latency reduces. In Fig. 2b, we can see that the success rate of the adaptive scheme is close to 100% for all the mobility speed values. The success rate of the constant scheme is also very high (98%-100%), while that of the deterministic walk with no stop times is between 94% and 99.5%. The success rate of the **SCD** algorithm is over 99%, too. In the cases of the adaptive and the constant schemes, the energy consumption is very high (Fig. 2c), because almost all messages are delivered (success rate close to 100%). The same stands for the **SCD** algorithm, too. On the other hand, when no stop times are used, the energy consumption is very high even if the success rate is not close to 100%; this happens because the sink moves in a way that it cannot collect all the data of a region when it visits it (especially when it moves very fast), so there is an increased need of retransmissions.

Comparing the results of the adaptive scheme with the results of the **SCD** algorithm, we observe that when the mobility speed is low, the adaptive scheme gives a higher latency than the **SCD**, but also achieves higher success rate and lower energy consumption. When speed increases the difference in the latency between the two strategies becomes very small, the success rate remains higher in the case of the adaptive scheme and energy consumption is lower or almost the same with the energy consumption of the **SCD** algorithm.

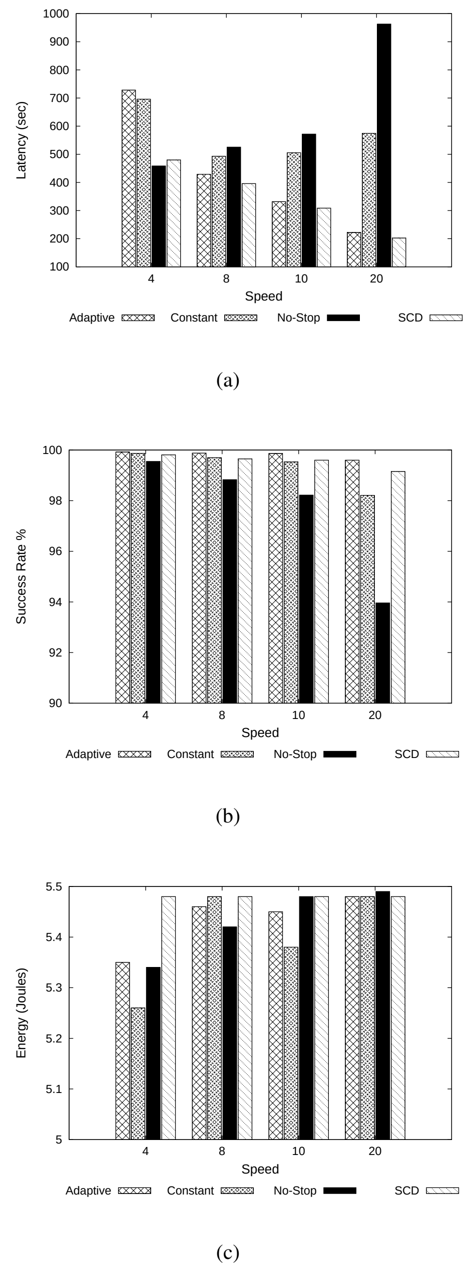


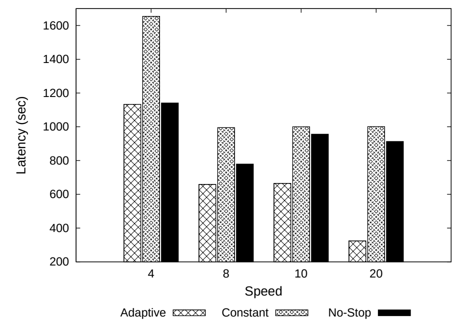
Fig. 2. Latency (a), success rate (b), energy dissipation (c) of the deterministic walk, for different speeds of the mobile sink.

Figure 3 depicts the performance of the proposed schemes when they are combined with the biased random walk. In Fig. 3a, we observe that by increasing the mobility speed, latency is reduced for the adaptive scheme. The constant scheme has high latency for low mobility speed, as the speed increases, it initially drops and then remains almost the same. The latency of the random walk with no stop times is high for low speed and slightly drops when speed increases. Besides, we see that the adaptive scheme achieves lower or almost the same latency for all the values of the mobility speed. Fig. 3b shows that the adaptive scheme achieves a very good success rate, almost 100%. The success rate of the constant scheme is also very good (95.5% - 99.5%). The success rate of the random walk with no stop times also increases as the mobility speed decreases. As shown in Fig. 3c, the energy consumption is high for all of the three cases; this is expected because the success rate of all the cases is high. Nevertheless, the adaptive scheme has a better behaviour in terms of energy consumption than the random walk with no stop times, because the latter consumes a lot of energy even when the succes rate is high (the same as in the deterministic walk explained above).

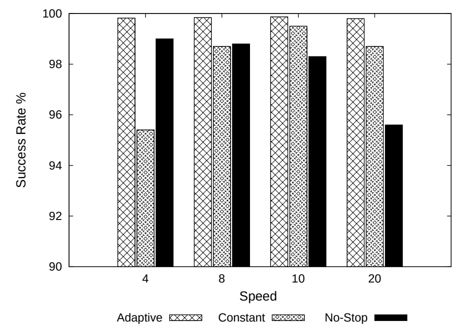
In more detail, the comparison between three main mobility strategies, our biased random walk with adaptive stop times, the random walk with no stop times and the deterministic traversal with no stop times, suggests that for mobility speed s = 20m/s, our adaptive scheme achieves significantly reduced latency; in particular the combination of adaptive stop times with biased network traversal achieves about 70% lower latency that the other two simple mobility strategies'. This implies that stop times and especially adaptation can accelerate network traversal and data collection a lot, since even when applied to the classic random walk (which is known to be slow) they lead to significant latency improvements (even compared to an optimized deterministic traversal). Thus, adaptation is very relevant, especially under weak models of very limited network knowledge (in such models, randomness is very suitable).

VI. Closing Remarks

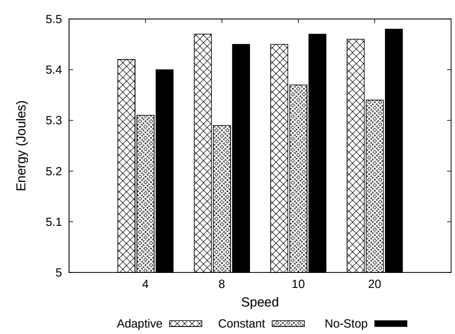
Our work points out the relevance of adaptively choosing the stopping times for sensory data collection using a mobile sink. In particular, in our method the stop times depend on the local density in each network region, towards balanced traversal, stopping more in regions with higher traffic. We propose both randomized mobility and optimized deterministic traversals. In all cases, the adaptive stop times lead to significantly reduced latency, especially for non-uniform sensor distribution, while keeping the delivery success rate and energy dissipation at satisfactory levels. We plan to continue this work by investigating additional stop strategies towards even better performance. Also, to combine with alternative suitable data dissemination mechanisms. Finally, we plan to consider adaptive loosely coordinated motion strategies for multiple mobile sinks.



(a)



(b)



(c)

Fig. 3. Latency (a), success rate (b), energy dissipation (c) of the biased random walk, for different speeds of the mobile sink.

References

[1] O. Powell, P. Leone, and J. D. P. Rolim, "Energy optimal data propagation in wireless sensor networks," *J. Parallel Distrib. Comput*, vol. 67, no. 3, pp. 302–317, 2007.

[2] A. Boukerche and K. Abrougui, "An efficient leader election protocol for mobile networks," in *In International Conference on Wireless Communications and Mobile Computing (IWCMC)*, ACM, Ed., New York, NY, USA, 2006, pp. 1129–1134.

[3] M. Elhadef, A. Boukerche, and H. Elkadiki, "Diagnosing mobile ad-hoc networks: two distributed comparison-based self-diagnosis protocols," in *4th ACM MobiWac*, 2006, pp. 18–27.

[4] A. Boukerche, Y. Ren, and Z. Zhang, "Performance evaluation of an anonymous routing protocol using mobile agents for wireless ad hoc networks," in *32nd IEEE LCN*, 2007.

[5] A. Kinalis and S. Nikoletseas, "Adaptive redundancy for data propagation exploiting dynamic sensory mobility," in *11th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. ACM Press, 2008, pp. 149–156. Also, in

the Journal of Interconnection Networks (JOIN), 2009

[6] 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.

[7] 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)*, 2005, pp. 244–257.

[8] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J.-P. Hubaux, "Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks," in *IEEE International Conference on Distributed Computing in Sensor Networks (DCOSS)*, vol. 4026, 2006, pp. 480–497.

[9] A. A. Somasundara, A. Ramamoorthy, and M. B. Strivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Transactions on Mobile Computing*, vol. 6, no. 4, pp. 395–410, April 2007.

[10] Y. Gu, D. Bozdağ, E. Ekici, F. Özgüner, and C.-G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks," in *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2005.

[11] "The network simulator – ns-2 http://www.isi.edu/nsnam/ns/." [Online]. Available: http://www.isi.edu/nsnam/ns/

[12] I. Chatzigiannakis, A. Kinalis, G. Mylonas, S. Nikoletseas, G. Prasinos, and C. Zaroliagis, "TRAILS, a toolkit for efficient, realistic and evolving models of mobility, faults and obstacles in wireless networks," in *41st ACM/IEEE Annual Simulation Symposium (ANSS)*, 2008, pp. 23–32.