Optimizing the Lifetime of Sensor Networks with Uncontrollable Mobile Sinks and QoS Constraints

FRANCESCO RESTUCCIA and SAJAL K. DAS, Missouri Univesity of Science and Technology

In past literature, it has been demonstrated that the use of mobile sinks (MSs) increases dramatically the lifetime of wireless sensor networks (WSNs). In applications where the MSs are humans, animals, or transportation systems, the mobility of the MSs is often uncontrollable and could also be random and unpredictable. This implies the necessity of algorithms tailored to handle uncertainty on the MS mobility. In this article, we define the lifetime optimization of a WSN in the presence of uncontrollable sink mobility and Quality of Service (QoS) constraints. After defining an ideal scheme (called *Oracle*) which provably maximizes network lifetime, we present a novel *Swarm-Intelligence-based Sensor Selection Algorithm* (SISSA), which optimizes network lifetime and meets predefined QoS constraints. Then we mathematically analyze SISSA and derive analytical bounds on energy consumption, number of messages exchanged, and convergence time. The algorithm is experimentally evaluated on practical experimental setups, and its performances are compared to that by the optimal *Oracle* scheme, as well as with the IEEE 802.15.4 MAC and TDMA schemes. Results conclude that SISSA provides on the average the 56% of the lifetime provided by *Oracle* and outperforms IEEE 802.15.4 and TDMA in terms of yielded network lifetime.

CCS Concepts: • **Networks → Sensor networks**

Additional Key Words and Phrases: Algorithms, TelosB, implementation, TDMA, 802.15.4

ACM Reference Format:
Francesco Restuccia and Sajal K. Das. 2016. Optimizing the lifetime of sensor networks with uncontrollable mobile sinks and QoS constraints. ACM Trans. Sen. Netw. 12, 1, Article 2 (March 2016), 31 pages.
DOI: http://dx.doi.org/10.1145/2873059

1. INTRODUCTION

Wireless sensor networks (WSNs) have become an affordable technology, able to support a wide variety of applications such as urban sensing and target tracking [Akyildiz and Vuran 2010]. As wireless sensors are tiny, energy-constrained devices, designing energy-efficient algorithms for reliable data gathering becomes crucial to optimize network lifetime. To this end, past literature has demonstrated that the use of *mobile sinks* (MSs) dramatically reduces the energy consumption of sensors, therefore extending the network lifetime [Di Francesco et al. 2011]. Specifically, MSs are special nodes that visit the WSN regularly to gather sensed data, eliminating the need of energy-expensive multihop communication [Yu et al. 2014]. Mobile sinks are also often employed whenever multihop transmission is not feasible, for example, due to sparse deployment of sensors [Restuccia et al. 2014].

This work was supported by NSF Grants No. CNS-1545037, CCF-1533918, DGE-1433659, CNS-1355505, and IIS-1404673.

Authors' addresses: F. Restuccia and S. K. Das, Computer Science Department, Missouri University of Science and Technology, 500 West 15th Street, 325 Computer Science Bldg., Rolla, MO 65409 USA; emails: {frthf, sdas}@mst.edu.

DOI: http://dx.doi.org/10.1145/2873059

The use of controllable sink mobility has been extensively studied in the literature to optimize the lifetime of WSNs [Gao et al. 2011; Liu et al. 2012; Xu et al. 2012; He et al. 2013; Gu et al. 2013; Tashtarian et al. 2015]. However, in a significant number of real-world applications [Campbell et al. 2008; Zhang et al. 2004; Haas and Small 2006; Chakrabarti et al. 2003] the MSs may present *uncontrollable* and *random* mobility. This means that (i) the MS may not be able to arbitrarily stop its motion for data collection and (ii) the trajectory, speed, and arrival time of the MS are unknown a priori [Khan et al. 2014]. Such mobility applies to many relevant scenarios where the MSs are pedestrians [Campbell et al. 2008]; animals, for example, zebras in the *ZebraNet* project [Zhang et al. 2004] or whales in the *SWIM* project [Haas and Small 2006]; or public transportation systems, where mobility heavily depends on current traffic conditions [Chakrabarti et al. 2003].

In the examples mentioned above, the time of the next MS visit and the actual time available for data transmission at each MS visit are uncertain and may be hard to predict [Yu et al. 2014]. Furthermore, in scenarios where sensors are deployed in challenging environments, for example, underneath the ground [Tooker and Vuran 2012] or on top of streetlight poles [Cenedese et al. 2014], extending the network lifetime becomes fundamental, as substituting/recharging the battery of sensors is cumbersome (or impossible). In addition, such applications may also require Quality of Service (QoS) constraints to be satisfied, for example, on throughput and data reliability. To the best of our knowledge, the problem of lifetime optimization in WSNs satisfying QoS constraints and allowing uncontrollable sink mobility is yet to be defined and solved. This motivates our work and the following novel contributions.

—After defining the network scenario under consideration, we define the problem of lifetime optimization of a WSN in the presence of uncontrollable sink mobility yet guaranteeing QoS constraints on data reliability and throughput. For comparison reasons, we formulate a scheme, called the *Oracle*, which is provably optimal (i.e., maximizes the network lifetime as defined in this article).

—To solve the lifetime optimization problem, we propose the *Swarm-Intelligence-based Sensor Selection Algorithm* (SISSA), which optimizes the sensor network lifetime and meets the desired QoS requirements without the need of any synchronization between the sensor nodes. We also develop an analytical model of SISSA and derive analytical bounds on the number of messages exchanged, energy consumption, and convergence time. We also provide an approximate formula to estimate the network lifetime as provided by SISSA.

—We validate the analytical model and evaluate the performance of SISSA on an experimental testbed composed of 40 TelosB [Crossbow 2014] sensor nodes, in both indoor and outdoor setups. To further evaluate our algorithm, we analytically compare the network lifetime yielded by SISSA with that of *Oracle*. Analytical and experimental results demonstrate that SISSA is highly scalable and energy efficient and provides on average 56% of the network lifetime given by the optimal scheme in every parameter set under consideration.

—Finally, we compare the performance of SISSA with respect to the IEEE 802.15.4 carrier sense multiple access/collision avoidance (CSMA/CA) medium access control (MAC) protocol [Society 2006], as well as with respect to the time division multiple access (TDMA) scheme, in terms of energy consumption, throughput, and yielded network lifetime. Results conclude that SISSA outperforms 802.15.4 and TDMA schemes, as it achieves desired QoS contraints with significantly lower energy consumption.

The rest of the article is organized as follows. Section 2 summarizes the related work, while Section 3 introduces the considered system model and the related assumptions. Section 4 defines the problem of lifetime optimization under uncontrollable sink mobility and QoS constraints and presents the ideal *Oracle* scheme, which provably maximizes network lifetime. Section 5 describes in detail the SISSA algorithm, whereas Section 6 derives its analytical model. Section 7 and 8 respectively validate the proposed model of SISSA and present the lifetime optimization results, including the comparison between SISSA and existing literature. Finally, Section 9 draws conclusions.

2. RELATED WORK

In this section, we summarize relevant work related to the lifetime optimization in WSNs in the presence of MSs and QoS constraints. For excellent surveys on WSNs with sink mobility, the readers may refer to Di Francesco et al. [2011], Khan et al. [2014], Tunca et al. [2014], Yu et al. [2014], and Gu et al. [2015]. Hereafter, we will use the terms "sensor node," "sensor," and "node" interchangeably.

A significant amount of research has been devoted to improve the lifetime of WSNs by designing energy-efficient routing protocols from the sensor nodes to the MSs in case the mobility is controllable, constrained, or unconstrained [Khan et al. 2014; Tunca et al. 2014]. As far as unconstrained mobility is concerned, in Li et al. [2012] the authors propose a routing scheme called ILSR, which is based on geographic routing and aims to ensure guaranteed packet delivery to a MS. More recently, in Shi et al. [2013] an efficient Data-Driven Routing Protocol (DDRP) for WSNs was proposed, which reduces network control overhead in route discovery/maintenance and improves data delivery performance. In this article, we consider WSNs in which multihop communication to the MS is not feasible. Therefore, such solutions are not applicable to the problem defined in this article, which is designing an algorithm for energy-efficient data transmission with QoS constraints to the MS.

A considerable amount of the literature has also exploited controllable sink mobility to optimize the lifetime of WSNs. For example, in Gao et al. [2011] the authors studied the problem of limited time available for data communication when path-constrained MSs are exploited. They proposed a data collection scheme that increases network throughput and decreases energy consumption of sensor nodes by balancing the load of appropriately chosen subsinks that relay the traffic of farther nodes to the MS. As regards to maximizing network lifetime along with QoS constraints, the authors in Xu et al. [2012] aimed at finding a trajectory for the MS, subject to constraints on the potential sojourn locations of the MS and maximum delay on data delivery. Similarly, in He et al. [2013], Gu et al. [2013], and Tashtarian et al. [2015] the authors study the problem of controlling sink mobility to achieve maximum network lifetime. The maximum throughput and lifetime of a WSN was studied in Liu et al. [2012], in which the data collection is performed using controllable mobility of MSs.

Despite the soundness of the above approaches, they do not optimize network lifetime when the assumption of controllable sink mobility no longer holds and, therefore, a different strategy is needed. In contrast, the proposed SISSA algorithm optimizes the lifetime of WSNs with QoS even though the MS mobility is uncontrollable, random, and eventually, unpredictable.

Alongside, adaptive algorithms have also been proposed in the literature to optimize network lifetime when the MS mobility is uncontrollable, yet can be *predictable* [Shah et al. 2011; Kondepu et al. 2012]. For example, in Shah et al. [2011] the authors proposed a learning-based technique to predict the arrival time probability and thus adapt the duty cycle of sensors based on the next estimated arrival time. On the other hand, other approaches exploit hierarchical schemes to improve network lifetime in case the mobility is *not* predictable. In Restuccia et al. [2012], the authors analyzed a hierarchical MS discovery protocol and evaluated its performance considering

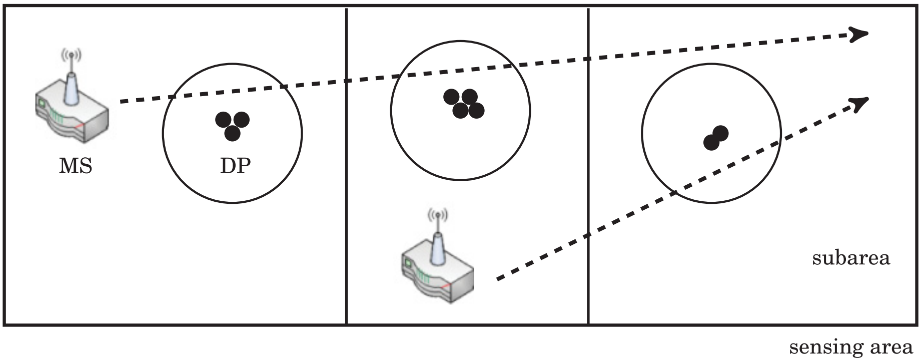

Fig. 1. Example of a deployment scenario with three subareas.

uncontrollable and random MS mobility. The work was extended in Restuccia et al. [2014] by considering more realistic MS mobility and packet loss models.

However, the main limitation of Shah et al. [2011] and Restuccia et al. [2014] lies in the assumption of sparse network scenario (one sensor only). When sensors are in the transmission range of each other, however, sensors must compete for a free channel after the MS discovery in order to start packet transmission. In this article, we consider such case and therefore deal with both the multiple access control (MAC) *and* MS discovery. Furthermore, the study in Restuccia et al. [2014] is limited to the constrained sink mobility, which means the MS is bound to follow a predefined trajectory with a given speed. In contrast, we relax this assumption and consider a more general scenario in which the MS mobility is unrestricted and may be unpredictable. Finally, we consider data reliability as a QoS constraint, which neither Shah et al. [2011] nor Restuccia et al. [2014] took into account.

3. SYSTEM MODEL AND ASSUMPTIONS

This section first describes the sensing scenario under consideration and the related assumptions and then defines the lifetime optimization problem with QoS.

3.1. Application Scenario

In this article, we concentrate on applications where the sensing area can be divided into *subareas*, which are monitored by one or more sensors deployed close to each other at a location of interest called *deployment point* (DP). By *close to each other* we mean that every node is the transmission range of each other and that the packet loss between them is close to zero. The nodes inside the same subareas are considered equivalent in terms of the data generated, which means that every device in the same subarea collects the same sensor readings. In practical scenarios, DPs may be, for example, streetlight poles, as in Cenedese et al. [2014], and/or strategic locations underneath the ground, as in Tooker and Vuran [2012]. Figure 1 shows a deployment example supposing the sensing area has been divided into three subareas having 3, 4, and 2 sensors in each DP, respectively. The transmission range of sensors has been depicted as a circle, while each DP has been represented by a black dot.

We chose such deployment scenario since it applies to a significant number of applications where fine-grain sensing is not needed (e.g., air/soil pollution monitoring), yet lifetime optimization may be critical. Specifically, when sensors are deployed in challenges environments, battery substitution/recharging may be difficult or impossible. In such cases, concentrating sensors only at strategic locations eases maintenance and deployment costs and helps to increase accuracy and network lifetime [Di Francesco et al. 2011]. We point out that this deployment scenario is very similar to the well-known sparse sensor network deployment strategy [Di Francesco et al. 2011] and has been recently employed in practical WSNs implementations, among others [Cenedese et al. 2014; Tooker and Vuran 2012]. Note that this scenario may be applied to several network configurations (ring network configuration, among others).

Other remarkable advantages of such sensing scenario are that (i) each network in each subarea cannot become partitioned, since every node in each subarea is in the transmission range of each other, (ii) the hidden node problem is absent, (iii) packet loss between sensors in each subarea is decreased, if not absent, and (iv) the MS enters in the transmission range of all sensors in the same subarea almost at the same time. These assumptions will be verified by experimental analysis in Section 7.

Owing to the fact that each subarea is independent from each other, multihop communication to a sink node becomes unfeasible. Thus, we assume one (or more) MSs are employed to collect the data acquired by the sensors in each subarea. The overall MS data collection process (i.e., MS visit and data collection) is called an *MS tour*. Hereafter, we will refer to *communication area* as the area in which the communication between the MS and the sensors can take place via receiving or sending beacons/packets.

As anticipated earlier, we assume the mobility of the MSs could be *uncontrollable and random*. We chose to analyze such mobility given it applies to most sensing scenarios in which the MSs are pedestrians, vehicles, or animals [Campbell et al. 2008; Zhang et al. 2004; Haas and Small 2006; Chakrabarti et al. 2003]. Specifically, such assumption implies that:

—The arrival time a_j of the MS during the j-th MS tour is not known a priori;
—The maximum time c_j available for data transmission during the j-th MS tour is also not known a priori.

We note here that uncontrollable and random mobility does *not* imply a_j and c_j are not to some extent *predictable*. Indeed, when the MS mobility presents patterns, energy consumption and efficiency can be further optimized (e.g., as in Shah et al. [2011]). For the sake of generality, in this article, we do not make any assumption about the regularity of a_j and c_j and assume the mobility pattern can be eventually unpredictable. The reader is referred to Khan et al. [2014] for additional discussions on different MS mobility models.

3.2. Data Collection Scenario

The states in which a sensor node can be at any time are *MS-Discovery*, *Data-Transfer*, and *Sleep*; the related transitions are shown in Figure 2. While in the *MS-Discovery* state, the sensor wakes up periodically (i.e., using a duty cycle[1]) in order to check for possible beacons from the MS. Upon the reception of a beacon, the sensor node transits to the *Data-Transfer* state. In such state, the sensor node tries to access the channel and transmit its data depending on the particular medium access control (MAC) protocol being used. After transferring all its data, the sensor node transits to the MS-Discovery state again. However, if the sensor node has a (even partial) knowledge about the mobility pattern of the MS (i.e., the next arrival time a_j can be predicted with some uncertainty), it can enter a *Sleep* state in which the radio is put in sleep mode to save energy.

In this case, the sensor node will wake up and switch to the MS-Discovery state W time units before a_j, where W is defined as the *waiting time*. The waiting time W is a quantity that expresses the *uncertainty* about the next arrival time of the MS to the

[1] By respectively defining T_{ON} and T_{SL} as the active and inactive times of the radio, the duty cycle δ is defined as the ratio between T_{ON} and $T_{ON} + T_{SL}$.

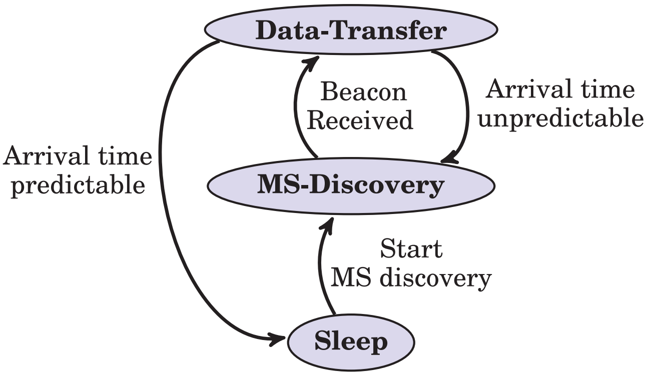

Fig. 2. State diagram of a sensor node.

sensing area. To give an example, let us assume that the MS is estimated to arrive at a particular time t. A waiting time $W = 100$s means that the sensor node must switch to the MS discovery state (i.e., wake up from sleep state and start duty-cycling) 100s before the predicted arrival time of the MS. This is because the MS could arrive in the time window $[t - 100\text{s}, t + 100\text{s}]$. In case the waiting time is equal to W = 0s, it means that there is no uncertainty about the next arrival time of the MS, and therefore the sensor nodes do not have to enter the discovery phase but just switch to the Data-Transfer state when the MS will arrive to the sensing area (i.e., the MS arrives periodically to the sensing area). In other words, the more uncertain the prediction of the next MS arrival is, the longer the waiting time will be.

In case it is not possible to predict the next arrival time of the MS, the sensor node enters the MS discovery state as soon as the MS exits from the sensing area and stays in discovery phase until the next MS arrival. Given that mechanisms to estimate the waiting time W have already been proposed (for example, Shah et al. [2011]), the computation of W is out of the scope of this article. In Section 8, we will consider different values of W to estimate the energy consumption of sensors and ultimately the network lifetime.

Figure 3 provides an example of MS tour and data collection, in which we show the relationship between W, a_j, and c_j. Furthermore, Figure 4 illustrates the MS discovery process by a sensor node, where $c(t)$ and $r(t)$ are, respectively, the wireless channel and the sensor node status (i.e., active/inactive). The MS is discovered as soon as the sensor becomes active and receives a beacon packet of duration T_{bd}, transmitted every T_{bi} time units by the MS.

4. PROBLEM DEFINITION

Before formally defining the network lifetime optimization problem, we define the QoS constraints considered in this article. In the following, we will consider the problem of maximizing the lifetime of a specific subarea. Since each subarea is independent from each other, the lifetime of each subarea can be maximized by applying the same strategy to each subarea.

Definition 1. The *QoS constraints* are defined as

(1) the number of sensors k_{des} transmitting their sensed data to the MS at each visit;
(2) the minimum time θ_{des} available for data exchange to each of the k_{des} nodes during each MS visit.

Here we discuss some important points regarding Definition 1.

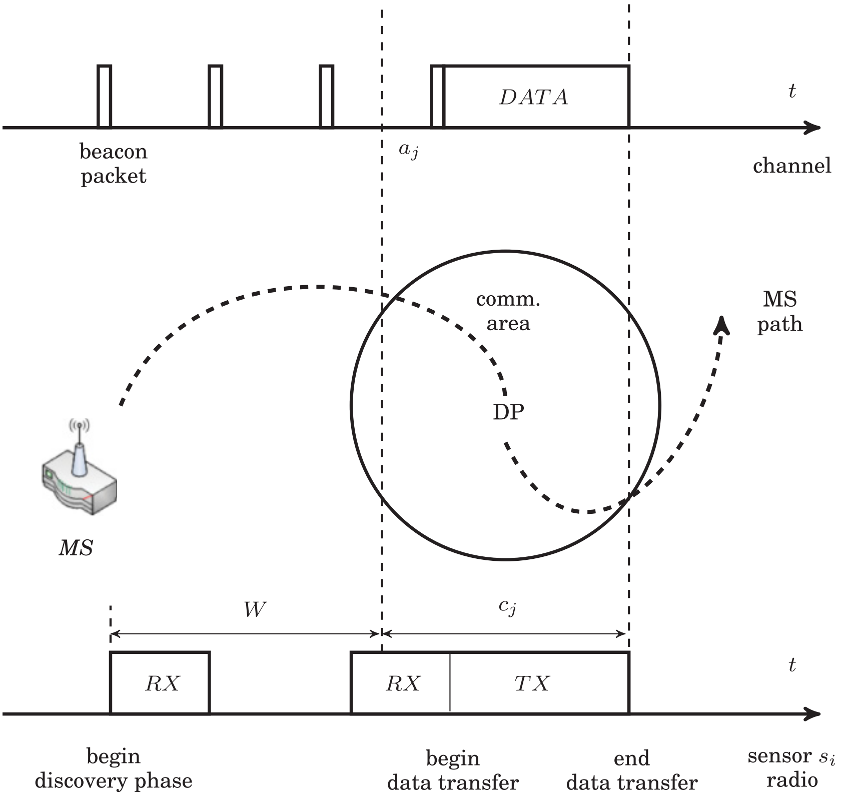

Fig. 3. Example of MS tour and data collection.

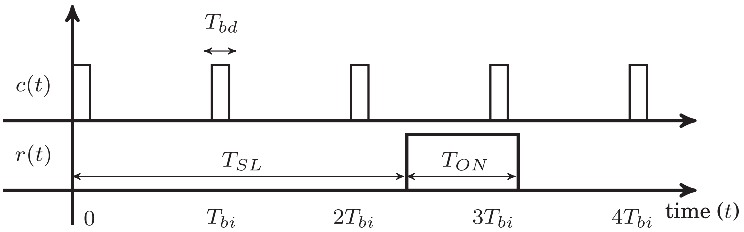

Fig. 4. The MS discovery process by a sensor node.

—We express the QoS constraints with (1) and (2) because they define, respectively, a constraint on *reliability* (more than one source of data) and on *efficiency* (maximization of data transfer per MS tour). Other QoS constraints, such as maximum end-to-end delay, and so on, have already been investigated in other contexts [Xu et al. 2012] and therefore are considered out of the scope of this article. Furthermore, note that such constraints are defined *for each subarea* composing the WSN. This is because each subarea may present *different* QoS requirements. This might happen, for example, because of budget constraints or because more stringent QoS requirements are needed in a subarea instead of another.

—We chose to express the reliability constraint as k_{des} because if more sources available for data collection are available, *fault-detection* techniques (e.g., exclude outliers in the case one or more sensors are faulty) can be implemented to verify data reliability.

For example, let us assume five sensors have been deployed in a subarea. Let us also suppose that after some time one sensor becomes faulty. If $k_{des} = 1$, then it will be impossible for the application to detect the fault because no other data source will be available. As k_{des} increases (maximum value of 5), detecting the fault will be easier as more data sources will be available to help recognizing the fault. Clearly, there is a tradeoff between the value of k_{des} and θ_{des} and the energy consumption of the sensors. Specifically, if more sensors transmit data during each MS tour, then the overall energy consumption will be higher, which ultimately leads to decreased network lifetime. Therefore, the k_{des} parameter must be carefully chosen by the application to allow a good tradeoff between desired fault tolerance and desired lifetime. The same consideration applies to θ_{des}; more data transmitted at each MS tour translates to decreased network lifetime.

—Defining θ_{des} as the amount of time allowed for transmission is a *more general* definition than the minimum amount of data. This is because the amount of data transferred is dependent from factors such as packet loss rate and packet header size. Conversely, our definition is independent from channel conditions and MAC protocols and can be used as an absolute measurement of the efficiency constrained of the sensing application. Furthermore, we point out that the goal of this paper is not to provide a mechanism for reliable data transmission to the MS, as this has already been explored in existing literature [Borgia et al. 2013]. To this end, we wanted the definition of the efficiency constraint to be independent from assumptions related to wireless transmission (e.g, packet loss rate, MAC protocol, signal modulation, and so on).

—In order to guarantee *feasibility* of the solution of the optimization problem, we assume that the MS and sensors remain in the transmission ranges of each other for at least $C_{min} = \theta_{des} \cdot k_{des}$ time units during each MS tour. In other words, C_{min} is sufficiently large to guarantee that θ_{des} time units of channel time will be available to k_{des} sensor nodes. In real-world applications, this condition might be met, for example, by constraining somehow the MS mobility or by choosing the deployment points of each subarea according to the MS mobility constraints (i.e., trajectory and speed). We point out that this assumption is necessary to guarantee that the QoS constraints imposed by the application can be satisfied and are independent from the solution of the optimization problem. Also, we point out that this assumption does not imply that the mobility of the MS is predictable.

—The value range of k_{des} is from $1 \leq k_{des} \leq S$, where S is the number of sensors deployed in the subarea. Furthermore, the range of θ_{des} depends on the value of k_{des} and the tour number j. For example, if $k_{des} = 1$, the range of θ_{des} will be from C_{min} to c_j (i.e., maximum time available for data transmission during the j-th MStour, see Figure 3). In general, θ_{des} may range between $C_{min}/k_{des} \leq \theta_{des} \leq c_j/k_{des}$.

We now define the concept of network lifetime that will be used throughout the paper.

Definition 2. The *network lifetime L* is the number of MS tours elapsed from the network deployment to the tour in which the first sensor depletes its energy.

Note that Definition 2 yields a result which is, in general, lower w.r.t. the actual network lifetime. This is because, in general, the subarea will be able to deliver sensed data with QoS to the MS for additional MS tours. However, we point out that the notion of network lifetime is usually *conventional*, and it is used to give an indication of the long-term performance of the network in delivering its service. Indeed, different definitions of lifetime may in general lead to different values of lifetime. In this paper, we chose such network lifetime definition to be coherent with the relevant related literature [Liu et al. 2012]. We also chose such definition for the sake of simplicity, as the mathematical analysis necessary to derive the network lifetime becomes significantly more tractable with such definition.

Furthermore, we point out that the number of MS tours directly indicates the number of times the subarea will be able to deliver sensed data to the MS with desired QoS constraints. Note that if the frequency of MS tours is known, then thte network lifetime expressed in time units can be obtained by dividing the number of MS tours by their frequency.

Let us now define the lifetime optimization problem. Let $E_d^{i,j}$ and $E_c^{i,j}$ define the energy spent by sensor s_i during the j-th MS tour in the discovery and communication phases, respectively. Then the total energy $E_{tot}^{i,j}$ will be equal to $E_d^{i,j} + E_c^{i,j}$. By noting that minimizing the energy consumption of every sensor node in every subarea of the WSN translates in the network lifetime maximization, the lifetime L of the entire WSN is consequently optimized. We define the network lifetime optimization problem as follows.

Definition 3. *Lifetime optimization problem.*

$$\begin{array}{c}\text{For every sensor } s_i,\\ \text{and for every MS tour } j,\end{array} \begin{cases}\text{Minimize } E_{tot}^{i,j}\\ \text{subject to}\\ \\ \theta \geq \theta_{des,p}\\ k \geq k_{des,p} \geq 1\end{cases}. \qquad (1)$$

We want to highlight that the solution of the optimization problem is *independent* from the definition of network lifetime. More specifically, the target of the lifetime optimization problem in Equation (1) is to minimize the energy consumption for every sensor s_i and for every MS tour j. Although such minimization may lead to different network lifetime values if different definitions are used, the result of the optimization (i.e., the duty-cycle value to be used by the sensors in the subarea, computed in Equation (8)) is independent from the network lifetime yielded by using the solution of the optimization problem. Indeed, note that the definition of network lifetime does not appear on the lifetime optimization problem.

4.1. The Oracle Scheme

We now propose an ideal scheme, hereafter referred to as the *Oracle* scheme, and prove that Oracle maximizes the lifetime of WSNs according to Definition 2. Henceforth, to simplify the mathematical notation, we will consider one subarea only and refer to θ_{des} and k_{des} as the QoS constraints for subarea. 2.

The Oracle scheme is designed as follows. Assuming a sensor s_i has perfect knowledge of each MS arrival time a_j for each MS tour j, no beacon packets are emitted by the MS, since the discovery phase is not necessary. We also assume that for the Oracle scheme, the sensor nodes know each other's energy level, and, additionally, the nodes are synchronized with each other. As a consequence, at time a_j, the k_{des} sensors having the highest residual energy budget will wake up from the sleep state and start transmitting data to the MS back to back until they have used the channel for θ_{des} time units. The selection of the k_{des} nodes at each MS tour is based on the IDs, such that for two nodes with the same energy budget, priority will be given to nodes with lower ID.

Figure 5 illustrates the Oracle scheme where $k_{des} = 2$, $S = 4$, and $\theta_{des} = 1$. In this example, the sensors with highest residual energy budget at $t = a_3$ are s_1 and s_2. Therefore, they start transmitting their data back to back as soon as the MS enters the communication area, and they stop transmitting as soon as they have used the channel for one time unit.

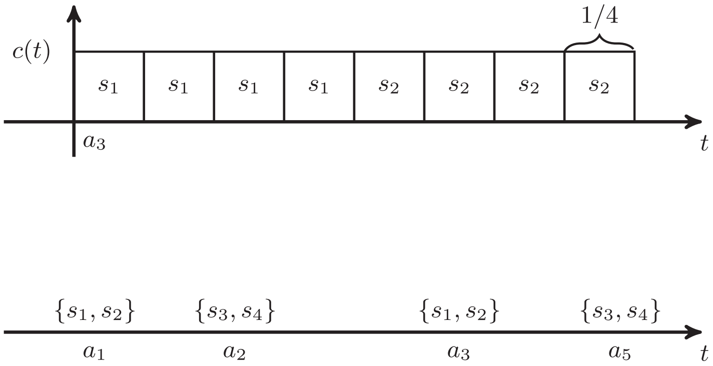

Fig. 5. Illustration of the Oracle scheme.

Let us now derive a simple formula to characterize the network lifetime provided by the Oracle scheme. Let $R = S/k_{des}$ define the *redundancy ratio* of the WSN, which measures the reliability of the network.[2] Also, let E_b define the initial energy budget of a sensor. Therefore, the energy E_{cp} spent by a sensor while communicating with the MS is given by $E_{cp} = P_{TX} \cdot \theta_{des}$, where P_{TX} is the packet transmission power of the sensor's radio. Then the following two lemmas hold.

LEMMA 1. *The network lifetime L_{or} provided by the Oracle scheme is given by*

$$L_{or} = E_b \cdot R/E_{cp}. \tag{2}$$

PROOF. By definition, the S sensors in the Oracle scheme are divided into R groups made up of k_{des} sensors (groups $\{s_1, s_2\}$ and $\{s_3, s_4\}$ in Figure 5). Each group will send data to the MS once every R number of MS tours in a round-robin scheme. Therefore, every sensor will spend E_{cp} units of energy every R tours. This implies the sensors in the first group (i.e., sensors s_1, s_2 in Figure 5) will deplete their energy budget after a number of MS tours equal to $E_b \cdot R/E_{cp}$. □

LEMMA 2. *The Oracle scheme maximizes the network lifetime according to Definition 2 while guaranteeing the QoS constraints as in Definition 1.*

PROOF. While using the Oracle scheme, sensors know exactly each MS's arrival time a_j, hence, no MS discovery is needed. This implies $E_d^{i,j} = 0$ for all $1 \leq i \leq S,\ j \geq 0$. Recall that (i) the only energy spent by the sensors is due to packet transmissions; (ii) channel access is contentionless; (iii) only k_{des} nodes transmit at each MS tour j and they use the channel for exactly θ_{des} time units. Therefore, we conclude that $E_c^{i,j}$ is minimized for all $1 \leq i \leq S,\ j \geq 0$ and the QoS constraints are satisfied. □

It is worth pointing out that Oracle is only an ideal scheme and not implementable in reality. This is because it assumes each sensor has exact knowledge about every MS's arrival time a_j, which is impossible under the hypothesis of uncontrollable and random MS mobility.

5. THE SISSA ALGORITHM

This section describes the SISSA that aims to solve the lifetime optimization problem as formulated in (1). To increase lifetime and at the same time guarantee the required QoS constraints, SISSA schedules during each MS visit a contention-free channel access

[2] Henceforth, without loss of generality we will assume that R is integer.

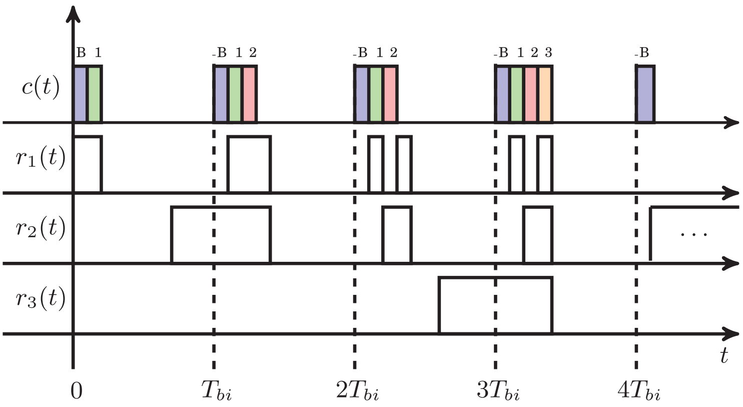

Fig. 6. A possible evolution of the SISSA swarm phase.

scheme such that only the k_{des} nodes with the *highest* residual energy levels among all S nodes transmit. Thereby, SISSA allows the remaining $S - k_{des}$ nodes to save energy and the selected k_{des} nodes to increase their channel access time at each MS tour. This ultimately allows each sensor to decrease dramatically its duty cycle and thus save energy and optimize the lifetime. Details on how to find the duty cycle δ^* to guarantee the desired throughput θ_{des} to the k_{des} sensors will be provided in Section 6. The proposed SISSA scheme consists of two different phases, the swarm phase and communication phase, as described below.

5.1. The Swarm Phase

The swarm phase is aimed at ensuring that each of the S nodes will be aware of the residual energy level[3] of the remaining $S - 1$ nodes after a certain amount of time, hereafter referred to as the *convergence time* of the swarm phase. In particular, as soon as the MS is discovered, each sensor node starts transmitting periodically what is called a *swarm agent*, a packet containing information about the residual energy level of the sensor node along with its ID.

Each swarm agent is transmitted using a time offset from the beacon derived from the sensor's ID, so the transmission of swarm agents is contention free, which does *not* require any synchronization between the sensors. At the same time, each sensor discovering the MS starts listening to possible swarm agents emitted by other sensors. The sensor nodes stop listening when every node has received a swarm agent from the remaining $S - 1$ sensors or a timeout occurs. This timeout depends on the duty cycle and will be derived in Section 6.

Figure 6 illustrates a possible evolution of the swarm phase, supposing that three sensor nodes s_1, s_2, and s_3 are competing for highest residual energy level. In this figure, $r_i(t)$ represents the state (i.e., ON or OFF) of the radio of sensor s_i, $1 \leq i \leq 3$, while $c(t)$ represents the channel status in terms of beacons/packets sent (B is the beacon packet).

In this example, we assume that the MS is discovered by s_1, s_2, and s_3 by means of the first, second, and fourth beacons, respectively, and that s_2 has the highest residual energy level among those three nodes. Since s_1 is the first node to discover the MS (at $t = 0$), it starts advertising periodically its swarm agent, while listening to possible swarm

[3] Note that, by knowing the initial energy budget, it is easy for each sensor node to estimate its residual energy by simply keeping track of the radio operations (i.e., transmissions, receptions, and computations) performed in the past.

agents from s_2 and s_3. Note that after receiving the swarm agent from s_2 (respectively, s_1), the node s_1 (respectivley, s_2) switches its radio off in correspondence with the transmission time of the swarm agent sent by s_2 (respectively, s_1) to save energy. Ultimately, the swarm phase converges at $t = 3 \cdot T_{bi}$, where T_{bi} is the beacon emission period, when all sensors have received a swarm agent from other nodes. Therefore, according to this example, the convergence time of the swarm phase is given by $T_{ct} = 4 \cdot T_{bi}$ (rounded to the next beacon transmission). Since s_2 is the node with the highest energy consumption, it starts transferring its data beginning at $t = T_{ct}$, while the other nodes resume their duty cycle and wait for another MS visit. The transmission phase ends as soon as the MS leaves the sensor network, implying that the MS and sensors are too far from each other to communicate.

Let us now point out some properties of the SISSA algorithm with the help of the above example.

—The SISSA swarm phase cannot converge until *every* sensor receives a swarm agent from *all* other sensors. This implies that each sensor node will terminate the swarm phase at the *same time*. Therefore, without any global information, intelligence, or synchronization, each sensor knows the swarm phase is completed only with the help of the knowledge provided by the "swarm intelligence" [Engelbrecht 2006];

—The sensor radio remains active only during the instants of swarm transmission/receptions. This allows the swarm phase to be energy efficient (the energy performance will be investigated in Section 6);

—SISSA does assumes neither a homogeneous initial energy budget nor a homogeneous sensor platform. This gives additional flexibility to the algorithm, which can indeed be implemented by using different platforms inside the same WSN.

—The swarm phase does *not* flood the network with swarm agents, as the communication is not multihop, and it occurs only once every MS tour. In the following section, we will derive strict analytical bounds on energy consumption, convergence time, and the number of swarm agents exchanged during each swarm phase.

—Although we assumed the most general case of uncontrollable and random MS mobility, SISSA is able to function also in case of controlled MS mobility.

In case one (or more) nodes fail, every sensor will reach timeout and stop emitting swarm agents. Therefore, every node in the WSN will know if any node s_f has failed by not receiving swarm agents from s_f during the swarm phase. Note that this is accomplished *without* the need of central synchronization. After every node stops emitting swarm agents, it will update the current value of S and the list of nodes still alive. Then the communication phase will start as described next.

5.2. The Communication Phase

At the beginning of the communication phase, each sensor is aware of the energy levels of other sensors thanks to the swarm agents received by them. As a result, each sensor is able to *autonomously* determine the k_{des} sensor nodes having the highest residual energy levels and whether it is allowed to transmit its data.

Thus, on one hand, k_{des} sensors recognize themselves as the "winners" of the competition and start transmitting their sensed data. On the other hand, the remaining $S - k_{des}$ nodes recognize themselves as the "losers" of the competition and return to their operating duty cycle (or sleep mode), waiting for the next MS visit. If two or more nodes have the same energy level, for tie breaking the node(s) with the highest ID(s) will be selected. Since by assumption the S nodes monitor the same event, the fact that the remaining $S - k_{des}$ sensors do not transmit their data does not affect the functionality of the application.

Table I. List of Major Symbols

Symbol	Meaning
$\mathcal{A}_i$	Automaton of sensor s_i
$\mathcal{Q}_i$	Local descriptor of $\mathcal{A}_i$
$\mathcal{Q}$	Global SaN descriptor
t_j	Transmission time of the j-th beacon
T_{bi}, T_{bd}	Beacon period, duration
T_{sa}	Swarm agent duration
T_{ON}, T_{SL}, T_P	Active, inactive, total time of the radio
r_i	Radio residual time at $t = 0$
$SR(t)$	Radio state at time t
$\mathcal{S}_Q$	SaN state space
Γ, γ^j	SISSA convergence time, distribution
D^i, λ_i^j	MS Discovery time by s_i, distribution

In order to guarantee the fastest channel access, each of the k_{des} nodes is allowed to transmit its data only in a specific time slot between the emission of two consecutive beacons. The transmission order is ID-wise, in the sense that the node with the lowest ID is the first to transmit, and so on. The data transfer phase ends as soon as the MS leaves the WSN, that is, the k_{des} sensor nodes do not receive any other beacon from the MS in a window of W_{max} seconds. The duration of the time slot assigned to each node depends on k_{des} and will be derived in the next section.

6. ANALYSIS OF SISSA

This section derives a comprehensive analytical model of the SISSA algorithm. In particular, we derive the MS discovery process and the sensors' radio models, followed by the swarm phase convergence time, some properties of the SISSA algorithm, and the energy consumption analysis. Finally, we derive an approximate formula to estimate the network lifetime yielded by SISSA.

Table I enumerates the list of major symbols used in the analysis.

6.1. MS Discovery Process

Before characterizing the MS discovery process, it is worthwhile to first introduce the concept of stochastic automata network (SaN) [Plateau and Atif 1991], representing a mathematical abstraction that models the interactions between a number of individual stochastic automata. Each automaton $\mathcal{A}_i$ is represented by a *local descriptor* $\mathcal{Q}_i$, which is a matrix representing the possible automaton states with transition probabilities. The *entire* system $\mathcal{A}$ is represented by a *global descriptor* $\mathcal{Q}$, which is a matrix obtained as a function of the local descriptors. In our analysis, we will model a sensor s_i through an automaton $\mathcal{A}_i$ and its local descriptor $\mathcal{Q}_i$, while the entire WSN composed of S sensors will be modeled by the global descriptor $\mathcal{Q}$ of the SaN.

In the following, we assume that the sensing rate is set in such a way that every sensor node is able to store in its memory at least $B \cdot \theta_{des}$ bytes, where B is the transmission rate of the sensor node's radio in bytes per second. Also, we assume that the transmission power of swarm agents is high enough to guarantee error-free transmission. This assumption is sound since the sensor nodes are assumed to be closely deployed to each other.

In order to derive the local descriptor $\mathcal{Q}_i$ associated with the automaton $\mathcal{A}_i$ for a given sensor s_i, it is necessary to model first the MS discovery process. Let $t = 0$ be the time at which the MS enters the WSN, that is, the time from which the beacons can be received by the sensor nodes. Figure 7 depicts the finite-state machine describing the evolution of the automaton $\mathcal{A}_i$ for sensor s_i during an MS visit. In particular, s_i

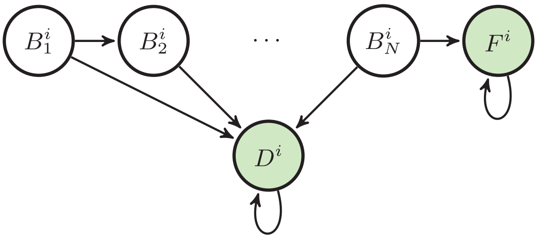

Fig. 7. Finite-state automaton describing the sensor s_i.

is in state B_j^i if the previous $j-1$ beacons sent by the MS since $t=0$ were missed, where $j \in [1, N]$ and $N = \lfloor \frac{C_{min}}{T_{bi}} \rfloor$ is the minimum number of beacons received during an MS visit. Therefore, the automaton $\mathcal{A}_i$ evolves from state B_j^i to B_{j+1}^i if s_i has not yet discovered the MS within time t_j of the j-th beacon transmission. Conversely, the automaton transits from state B_j^i to the absorbing state D^i if the MS is discovered by means of the j-th beacon and from state B_N^i to the absorbing state F^i if the MS has not been discovered during this visit.

Since the reception of the j-th beacon depends on the initial state of the sensor's radio at time $t=0$, each state B_i^j is in reality a macrostate that keeps track of all possible radio states of the i-th node just before receiving the j-th beacon. The composition of such macro states, the transition probabilities between the states and the number of possible radio states are provided below.

Let us now derive the values of t_j, for $1 \leq j \leq N-1$. Assuming that time is discretized with slots of duration Δ, the time t_0 of the first beacon transmission into the WSN is a random variable that can assume all possible values in the set $\Lambda \equiv \{0,\ \Delta, \ldots,\ n \cdot \Delta\}$, where $n = \lfloor \frac{T_{bi}}{\Delta} \rfloor$. For this reason, the time t_j of the j-th beacon transmission is given by $t_j \triangleq \{t_0 + j \cdot T_{bi},\ 1 \leq j \leq N-1\}$. For the sake of simplicity, we consider a generic $t_0 \in \Lambda$. This constraint t_0 will be relaxed later.

In order to derive the radio model, we first define $T_P \triangleq T_{ON} + T_{SL}$ as the period of the sensor node's radio, where $T_{SL} = \frac{(1-\delta) \cdot T_{ON}}{\delta}$ and δ is the duty-cycle *ratio*. Henceforth, we will assume that T_{ON} will be equal to the minimum value $T_{bi} + T_{bd}$ such that a sensor node will be able to receive at least one beacon packet in the active mode. To characterize the radio state of a generic sensor at beacon reception time t_j, we introduce the function $SR(t)$ that assumes the value ON or OFF if the sensor radio is active or inactive at time t. Let us indicate by r_i the residual time of the initial radio state $SR(0)$, starting from time $t=0$. Since the time is discretized in slots of duration Δ, the residual time r_i of the radio state can assume uniformly $M = \lceil \frac{T_P}{\Delta} \rceil$ different values. Given r_i, and defining $t'_j = t_j \bmod T_P$, the state $SR(t_j)$ of the radio of sensor s_i at time t_j is given by

$$\begin{array}{c} SR(t_j) \\ SR(0) = ON \end{array} = \begin{cases} ON & \text{if } t'_j \in [0, r_i] \cup [r_i + T_{SL}, T_P) \\ OFF & \text{if } t'_j \in [r_i, r_i + T_{SL}) \end{cases},$$

$$\begin{array}{c} SR(t_j) \\ SR(0) = OFF \end{array} = \begin{cases} OFF & \text{if } t'_j \in [0, r_i) \cup [r_i + T_{ON}, T_P) \\ ON & \text{if } t'_j \in [r_i, r_i + T_{ON}) \end{cases}.$$

The above equations can be justified as follows. Since the time of the first beacon transmission t_0 is assumed to be known, and because the radio evolves deterministically with a period T_P, it also follows that $SR(t_j)$ always evolves in a deterministic and periodic manner. Therefore, it is possible to derive $SR(t_j)$ by comparing t'_j against the initial residual time r_i.

Once the radio state at beacon reception times and the MS discovery process have been fully characterized, we can derive the transition probabilities of the automaton $\mathcal{A}_i$. Equation (3) below depicts the descriptor $\mathcal{Q}_i$ of $\mathcal{A}_i$, derived from the finite-state machine depicted in Figure 7. Let $m_{k,j},\ 1 \leq k \leq M$ be the set of all possible radio states at time t_j, obtained by using the $RS(t)$ function calculated at time t_j and by considering all possible M initial radio states. Given the initial residual time can assume M values, blocks $\mathcal{P}^i_{xy}$ in matrix $\mathcal{Q}_i$ have size $1 \times M$ and keep track of all possible transition probabilities from the generic state B^i_x to the generic state B^i_y (respectively to F^i or D^i if $x = N$):

$$\begin{array}{c@{}c} & \begin{array}{cccccc} B^i_1 & B^i_2 & \cdots & B^i_N & D^i & F^i \end{array} \\ \begin{array}{c} B^i_1 \\ B^i_2 \\ \vdots \\ B^i_N \\ F^i \\ D^i \end{array} & \left(\begin{array}{cccccc} \mathcal{P}^i_{12} & 0 & \cdots & 0 & \mathcal{P}^i_{1D} & 0 \\ 0 & \mathcal{P}^i_{23} & 0 & \cdots & \mathcal{P}^i_{2D} & 0 \\ & \vdots & & & \vdots & \\ 0 & 0 & 0 & 0 & \mathcal{P}^i_{ND} & \mathcal{P}^i_{NF} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \end{array} \tag{3}$$

By pointing out that a sensor discovers the MS independently of others, it follows that the transition probability at time t_j depends only on that node's radio. Therefore, the M subblocks of the block $\mathcal{P}^i_{xy}$ are derived as follows:

$$p^{(m_{k,j})}_{B_j B_{j+1}} = \begin{cases} 1 & m_{k,j} = OFF \\ 0 & \text{otherwise} \end{cases} \quad p^{(m_{k,j})}_{B_j D^i} = \begin{cases} 1 & m_{k,j} = ON \\ 0 & \text{otherwise} \end{cases}$$

$$p^{(m_{k,N})}_{B_N F^i} = \begin{cases} 1 & m_{k,N} = OFF \\ 0 & \text{otherwise} \end{cases}.$$

Once the description of the single automaton is completed, we can now derive the SaN global descriptor $\mathcal{Q}$. By definition, the state space of the SaN can be obtained by the Cartesian product of the state space of each automaton corresponding to a sensor s_i. Let $\mathcal{S}_Q$ define the set containing the state space of the SaN. For example, for $S = 5$ sensors, a state $\xi \in \mathcal{S}_Q$ can be obtained as $\xi = \{B^1_5, B^2_5, D^3, B^4_2, D^5\}$. Since the overall SaN is composed of S autonomous and independent automata, following the arguments in Plateau and Atif [1991], Claim 1 below holds.

CLAIM 1. *Given S independent stochastic automata $\mathcal{A}_1, \ldots, \mathcal{A}_S$, with associated descriptors $\mathcal{Q}_1, \ldots, \mathcal{Q}_S$, the global descriptor $\mathcal{Q}$ can be derived as $\mathcal{Q} = \otimes^S_{i=1} \mathcal{Q}_i$, where $\otimes$ is the tensor product operator (see Itskov [2007] for a definition of tensor product).*

Although $\mathcal{Q}_i$ is a sparse tensor, its dimensions may become prohibitive for realistic values of S. However, to keep the model scalable, we can derive the following corollary from the above claim.

COROLLARY 1. *By defining $\pi^{(j)}_i$ as the state probability vector associated with time t_j of the j-th beacon transmission by the i-th automaton (for sensor s_i), the distribution*

$\pi_Q^{(j)}$ *associated with the tensor* $\mathcal{Q}$ *can alternatively be derived as* $\pi_Q^{(j)} = \otimes_{i=1}^{S} \pi_i^{(j)}$ *for* $0 \leq j \leq N+1$.

The resulting model is now *scalable*, as each $\pi_i^{(j)}$ can be calculated separately and then combined with others with the tensor product to obtain $\pi_Q^{(j)}$.

6.2. Convergence Time

In this subsection we derive the SISSA swarm phase convergence time. Hereafter, the notation $\mathbb{P}\{e\}$ will denote the probability of occurrence of event e. First, H^j defines a discrete-time random process that denotes the total number of automata discovering the MS at time $t \leq t_j$, given that the first beacon transmission occurred at $t = t_0$. Furthermore, let us define by $h_{jk}^{t_0}$ the probability mass function of H^j, such that

$$h_{jk}^{t_0} \triangleq \mathbb{P}\{H^j = k \mid t_0\}, \text{ for } 0 \leq k \leq S.$$

Let us define $\mathcal{C}_k$ as the subset of tuples in $\mathcal{S}_Q$ indicating that exactly k nodes have discovered the MS. For example, a tuple in the set $\mathcal{C}_3$ is $\{B_2^1, D^2, D^3, B_2^4, D^5\}$. By definition of $\mathcal{C}_k$, it follows that

$$h_{jk}^{t_0} = \sum_{\xi \in \mathcal{C}_k} \pi_Q^{(j)}(\xi), \text{ for } 0 \leq k \leq S, \tag{4}$$

where $\pi_Q^{(j)}(\xi)$ is the component of the vector $\pi_Q^{(j)}$ corresponding to the state $\xi \in \mathcal{C}_k$.

Now let Γ define the random variable describing the SISSA swarm phase convergence time. Recalling that $h_{jS}^{t_0}$ is the probability that exactly S nodes have discovered the MS at time t_j, the distribution γ^{j,t_0} of Γ restricted to specific t_0 value can be derived as follows:

$$\gamma^{j,t_0} = \begin{cases} h_{jS}^{t_0} & j = 0 \\ h_{jS}^{t_0} - h_{(j-1)S}^{t_0} & 1 \leq j \leq N \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

Finally, in order to eliminate the dependency of γ^{j,t_0} from t_0, we consider all possible values of such variables and the corresponding probabilities. Since every $t_0 \in \Lambda$ has the same uniform probability $\frac{\Delta}{T_{bi}}$ of occurrence, the distribution γ^j of the SISSA swarm phase convergence time, Γ, is given by

$$\gamma^j = \sum_{t_0 \in \Lambda} \gamma^{j,t_0} \cdot \mathbb{P}\{t_0\} = \frac{\Delta}{T_{bi}} \cdot \sum_{t_0 \in \Lambda} \gamma^{j,t_0}. \tag{6}$$

By using a similar procedure (not reported here due to lack of space), we can derive the distribution λ_i^j of the MS discovery time D_i by sensor s_i.

6.3. Properties and Analytical Bounds

Let us now prove some important properties and analytical bounds yielded by the SISSA algorithm. The first property is related to the maximum convergence time of the swarm phase.

Let us define T_{SL}^* as the sensor radio inactivity time associated with the use of duty-cycle ratio δ^*. Let t_j^* be the minimum t_j such that $t_j \geq T_{SL}^*$. Then the following claim holds.

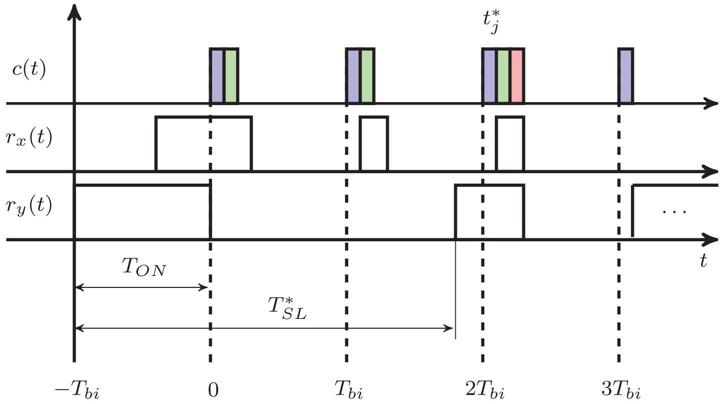

Fig. 8. Worst-case convergence time of SISSA.

Claim 2. *In case of no node failure(s), the SISSA swarm phase convergence time is* at most t_j^* *when using a duty-cycle ratio* δ^*.

Proof. A sensor node using a duty cycle δ_j^* will wake up at most T_{SL}^* time units after the emission of the first beacon into the WSN. This means that, in the worst case, the sensor node will discover the MS by means of the j^*-th beacon. Therefore, the SISSA swarm phase will converge in at most at $t = t_j^*$ time units. □

In Figure 8 is exemplified the worst-case convergence time of SISSA. In this example, sensor node x discovers the MS by means of the first beacon (emitted at time $t = 0$), while the node y will discover the MS at time $t = t_j^*$.

The above implies that the maximum convergence time of the swarm phase of SISSA depends not on the number S of sensors in the WSN but on the duty-cycle ratio δ^* only. This gives scalability to the SISSA algorithm. Let us now derive some interesting corollaries from this claim.

Corollary 2. Timeout value. *The timeout has to be set to* $\tau = t_j^*$ *for each node, since* t_j^* *is the maximum convergence time of the swarm phase.*

Corollary 3. Maximum number of swarm agents. *The number of swarm agents (i.e., the number of messages) a sensor node emits during the swarm phase is* $O(\frac{t_j^*}{T_{bi}}) \equiv O(1)$.

Corollary 4. Maximum energy consumption during swarm phase. *Given the number of swarm agents received by each node is* $S - 1$, *we can derive an upper bound on the energy consumption by the sensor nodes during the swarm phase. In particular, by defining* P_{RX} *as the reception power of the sensors' radio and* P_{TX}^{sa} *as the transmission power of swarm agents, the maximum energy* E_{max}^{sp} *each node spends during the swarm phase by each node is*

$$E_{max}^{sp} = P_{TX}^{sa} \cdot \frac{t_j^*}{T_{bi}} \cdot T_{sa} + P_{RX} \cdot T_{sa} \cdot (S - 1).$$

Corollary 5. Minimum channel time. *Recalling that* C_{min} *is the minimum available contact time, then SISSA* guarantees *that* at least $C_{min} - t_j^*$ *time will be available to the* k_{des} *sensors for data communication. In particular, defining* $T_k = \frac{(T_{bi} - T_{bd})}{k_{des}}$ *as the slot duration of each node selected for data transfer, the minimum channel time available* *to each of the k_{des} nodes during an MS visit is obtained as*

$$\theta_{min} = T_k \cdot \left\lfloor \frac{C_{min} - t_j^*}{T_{bi}} \right\rfloor . \tag{7}$$

From Equation (7), we can solve the following optimization problem and derive the *optimum* duty-cycle ratio δ^*_{opt} that allows each of the k_{des} nodes to attain a communication time θ_{min} of at least θ_{des}. Thus,

$$\delta^*_{opt} = \min_{\delta^* \in (0,1]} \left\{ \delta^* : \theta_{min} \geq \theta_{des} \right\} . \tag{8}$$

The solution can be calculated by simply discretizing the continuous interval $(0, 1]$ of the duty-cycle ratio δ^* and applying an exhaustive search.

6.4. Energy Consumption Analysis

In this subsection, we derive the distribution of energy consumption by each sensor node as a summation of the energy consumed during the MS discovery process and the swarm and communication phases.

Let us now compute the distribution of the energy E^i_{dp} spent by the sensor s_i during MS discovery. Assuming that sensor nodes use a duty-cycle ratio δ^*, the energy consumption distribution $E^i_{dp}(j),\ 0 \leq j \leq j^*$, can be defined as

$$E^i_{dp}(j) = \mathbb{P}\{E^i_{dp} = P_{RX} \cdot \delta^* \cdot (W + t_j)\},$$

where W is the estimated waiting time and P_{RX} is the radio power in the receiving state. The sensor s_i begins the discovery phase W time units in advance of the (predicted) MS arrival into the WSN and then remains in the discovery process for additional t_j seconds (i.e., until the MS is discovered). Therefore, with a probability λ_i^j, the sensor consumes $P_{RX} \cdot \delta^* \cdot (W_m + t_j)$ units of energy, thus leading to the following probability distribution:

$$E^i_{dp}(j) = \lambda_i^j, \text{ for } 0 \leq j \leq j^*. \tag{9}$$

We now derive the distribution of energy E^i_{sp} spent by s_i during the swarm phase. Since this distribution depends on D^i (discovery time of the MS by s_i) and Γ, we define the distribution $E^i_{sp}(j, z, t_0),$ for $0 \leq j \leq z \leq j^*$, as

$$E^i_{sp}(j, z, t_0) = \mathbb{P}\{E^i_{sp} = E \mid D^i = t_j,\ \Gamma = t_z,\ t_0\}. \tag{10}$$

Next we derive the energy E used in the above equation. The amount of energy spent between t_j and t_{j+1} during the swarm phase by a sensor will be the sum of the energy spent for receiving the beacon and that spent for receiving the still missing swarm agents and transmitting its swarm agent. By pointing out that the distribution $\psi(j, k)$ of the still missing swarm agents at time t_j can be derived as the probability that exactly $S - k$ nodes have discovered the MS at time $t \leq t_{j-1}$, and by recalling the definition of $h^{t_0}_{jk}$ in Equation (4), the distribution $\psi(j, k)$ can be recursively derived as

$$\psi(j,k) = \begin{cases} 1 & j = 0,\ k = S - 1 \\ \hat{h}^{t_0}_{j-1,S-k} & 1 \leq j \leq z \\ 0 & \text{otherwise} \end{cases} ,$$

where $\hat{h}_{jk}^{t_0}$ is the distribution described in Equation (4) normalized from t_0 to t_z. Therefore, E is derived as

$$E = P_{TX}^{sa} \cdot T_{sa} + P_{RX} \cdot \left(T_{bd} + \sum_{k=0}^{S-1} \sum_{m=j}^{z} k \cdot T_{sa} \cdot \psi(m,k) \right),$$

where P_{TX}^{sa} is the transmission power of swarm agents and T_{sa} is the duration of a swarm agent. The joint distribution $J^i(j,z) \triangleq \mathbb{P}\{D^i = t_j, \Gamma = t_z\},\ 0 \le t_j \le t_z \le t_j^*$ and the distribution $E_{sp}^i(j,z,t_0)$ can be derived as

$$J^i(j,z) = \frac{\lambda_i^j \cdot \gamma^z}{\sum_{0 \le j \le z \le j^*} \lambda_i^j \cdot \gamma^z}$$

$$E_{sp}^i(j,z,t_0) = J^i(j,z),\ 0 \le j \le z \le j^*. \tag{11}$$

We can easily remove the dependency from t_0 by using the same procedure as in Equation (6).

Next we calculate the energy spent by one of the k_{des} nodes during the communication phase. Since each of these nodes accesses the channel in a slotted fashion, the energy spent during this phase is only for transmitting each data packet, without any additional energy overhead. Thus, the energy distribution $E_{cp}^i(z),\ \ 0 \le z \le j^*$ is defined as

$$E_{cp}^i(z) = \mathbb{P}\left\{ E_{cp}^i = P_{TX}^{msg} \cdot T_k \cdot \left\lfloor \frac{C_{min} - t_z}{T_{bi}} \right\rfloor \right\},$$

where P_{TX}^{msg} is the transmission power of data packets. Similarly to Equation (9), it follows that the distribution $E_{cp}^i(z)$ is given by

$$E_{cp}^i(z) = \gamma^z,\ 0 \le z \le j^*. \tag{12}$$

Equation (12) derives from the fact that γ^z is the distribution as a function of time t_z, where $0 \le z \le j^*$.

6.5. Lifetime Analysis

In the following, we derive a simple yet effective formula to approximate the total network lifetime L_{sis} provided by the SISSA algorithm. First, as in the ideal Oracle scheme, let us define *redundancy ratio* as the quantity $R = S/k_{des}$, where S is the number of sensors and k_{des} is the QoS parameter defined by the application. In addition, let us define $\overline{E}_{dp}$, $\overline{E}_{sp}$, and $\overline{E}_{cp}$, respectively, as the average energy spent during the MS discovery, swarm, and communication phases obtained by using Equations (9), (11), and (12).

To estimate the lifetime provided by SISSA, we assume that the set of k_{des} sensors will be selected during each MS tour following a round-robin fashion (similarly to the Oracle scheme). This assumption is sound due to the fact that sensors with the highest energy consumption will transmit at each MS tour, and approximately every sensor will spend the same amount of energy. Indeed, we also assume that each of the k_{des} nodes will spend the same amount of energy during each tour, which is $\overline{E}_{dp} + \overline{E}_{sp} + \overline{E}_{cp}$. Conversely, each of the $S - k_{des}$ nodes will spend $\overline{E}_{dp} + \overline{E}_{sp}$ units of energy, since they will not send their data to the MS. Therefore, the following lemma holds.

LEMMA 3. *By defining E_b as the initial energy budget of each sensor node as in the Oracle scheme, the network lifetime L_{sis} provided by SISSA is approximated by*

$$L_{sis} \approx \frac{E_b}{R \cdot (\overline{E}_{sp} + \overline{E}_{dp}) + \overline{E}_{cp}} \cdot R. \tag{13}$$

PROOF. Assuming that each sensor will be selected once every R tours, the energy consumption of a single sensor during a period of R different MS tours will be obtained as $\overline{E}_{sp} + \overline{E}_{dp} + \overline{E}_{cp}$ during one tour and $\overline{E}_{sp} + \overline{E}_{dp}$ for the remaining $R - 1$ tours. By dividing each sensor's energy budget E_b by the total energy spent for all R tours, we can estimate the number of times a sensor will be able to conclude a period of R tours. By multiplying this quantity by R, we can approximate the number of tours one sensor will be active before dying, which is the network lifetime by definition. □

We now compare the lifetime provided by SISSA against that provided by the ideal Oracle scheme. Let us define the lifetime approximation ratio (LAR) as the lifetime provided by SISSA divided by the lifetime provided by Oracle, that is, LAR $= L_{sis}/L_{ora}$. Applying expression (13), we derive

$$\begin{aligned} \text{LAR} &= \frac{E_b \cdot R}{R \cdot (\overline{E}_{sp} + \overline{E}_{dp}) + \overline{E}_{cp}} \cdot \frac{E_{cp}}{E_b \cdot R} \\ &= \frac{E_{cp}}{R \cdot (\overline{E}_{sp} + \overline{E}_{dp}) + \overline{E}_{cp}}. \end{aligned} \tag{14}$$

By definition of LAR, we conclude that SISSA approximates better the Oracle scheme as the energy spent in the discovery ratio decreases. This is because the SISSA algorithm includes in its execution the MS discovery, which is not considered in the Oracle scheme but necessary in real WSNs implementations. The performance of SISSA as a function of R and the duration of MS discovery process will be evaluated in details in Section 8.

7. EXPERIMENTAL MODEL VALIDATION

In this section, we first validate the analytical model of SISSA through experimental evaluation. Then we evaluate the impact of the transmission power level of swarm agents and the distance between sensors on the performance of SISSA.

7.1. Indoor Experiments

In the experiments, we set up an indoor experimental setup composed by 40 TelosB [Crossbow 2014] sensors deployed over a 4 × 10 grid (see Figure 9). This indoor setup was used to validate the assumptions of both the system model and the mathematical analysis. The data collection phase was carried out by a volunteer graduate student walking at a speed of about 2 m/s and holding in his hands another sensor acting as the MS. The remaining experimental parameters are summarized in Table II. All confidence intervals has been set to 95%. In each experiment we performed 50 tours of the MS.

Tables III and IV summarize the analytical and experimental results, respectively, of the average energy spent by sensor nodes during the swarm phase and the average swarm phase convergence time, as a function of both the number of nodes in the WSN and the duty-cycle parameter, δ. Results conclude that our analytical model accurately captures the performance of the SISSA algorithm in real implementations. As expected, Tables III and IV conclude that the energy spent during the swarm phase increases with the number of nodes in the WSN. This is because the number of swarm agents to be received by each sensor increases with the number of nodes. Furthermore,

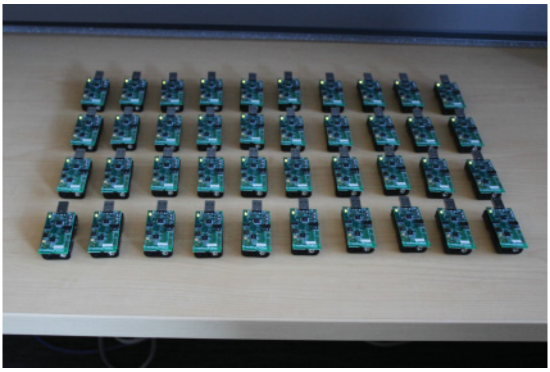

Fig. 9. Indoor experimental setup.

Table II. Parameters Used for Evaluating SISSA

Parameter	Value
Reception (RX) power of radio	56.4mW
Transm. (TX) power, beacon and messages	49.5mW
TX power, swarm agents	31.32mW
Beacon interval	200ms
Beacon duration	2ms
Swarm agent duration	2ms
T_{ON}	202ms
W	1s
Time slot (Δ)	10ms
Bitrate	250Kbps
Message size (B_m)	133bytes
Message duration (T_m)	4.256ms
Battery type	AA, 2500mAh

Table III. Swarm Phase Energy Consumption (S = Number of Sensors, δ = Duty-Cycle Ratio, CI = Confidence Interval)

	$\delta = 3\%$			$\delta = 5\%$			$\delta = 7\%$			$\delta = 9\%$		
S	Mod.	Exp.	CI	Mod.	Exp.	CI	Mod.	Exp.	CI	Mod.	Exp.	CI
5	2.26	2.36	±0.52	1.45	1.74	±0.18	1.07	1.17	±0.23	0.89	0.98	±0.44
10	3.58	3.77	±1.06	2.29	2.94	±0.32	1.71	1.95	±0.30	1.42	1.76	±0.60
20	6.21	5.47	±1.62	3.98	4.67	±0.60	2.97	3.29	±0.41	2.46	2.86	±0.95
30	8.85	9.28	±2.56	5.66	5.87	±0.68	4.23	4.43	±0.50	3.50	3.50	±1.15
40	11.50	11.00	±2.47	7.34	7.70	±0.90	5.49	5.81	±0.60	4.54	4.94	±1.84

Table IV. Swarm Phase Convergence Time (S = Number of Sensors, δ = Duty-Cycle Ratio, CI = Confidence Interval)

	$\delta = 3\%$			$\delta = 5\%$			$\delta = 7\%$			$\delta = 9\%$		
S	Mod.	Exp.	CI	Mod.	Exp.	CI	Mod.	Exp.	CI	Mod.	Exp.	CI
5	5.80	5.67	±0.20	3.60	3.78	±0.17	2.60	2.56	±0.34	2.2	2.60	±0.51
10	6.20	6.06	±0.30	3.60	3.64	±0.21	2.80	2.88	±0.40	2.2	2.40	±0.36
20	6.20	6.53	±0.35	4.00	4.12	±0.24	3.00	3.14	±0.55	2.4	2.46	±0.25
30	6.60	6.61	±0.15	4.00	3.95	±0.20	3.00	2.75	±0.50	2.4	2.57	±0.42
40	6.80	6.54	±0.28	4.00	4.06	±0.30	3.00	3.44	±0.50	2.4	2.85	±0.58

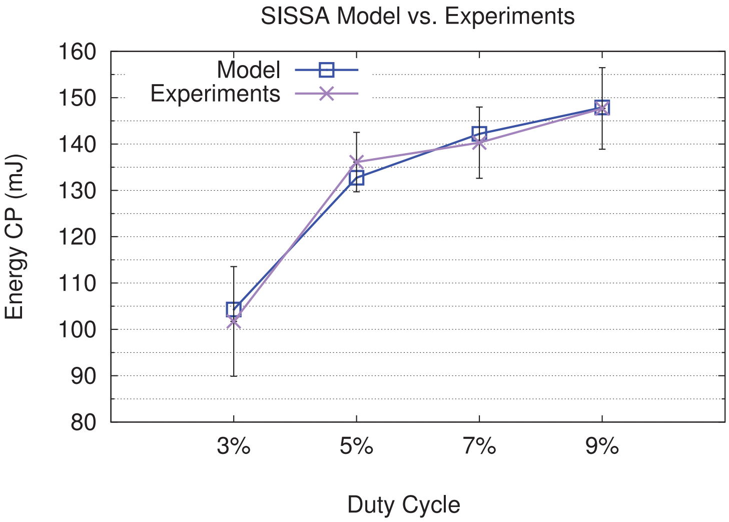

Fig. 10. Energy spent (mJ) during communication phase.

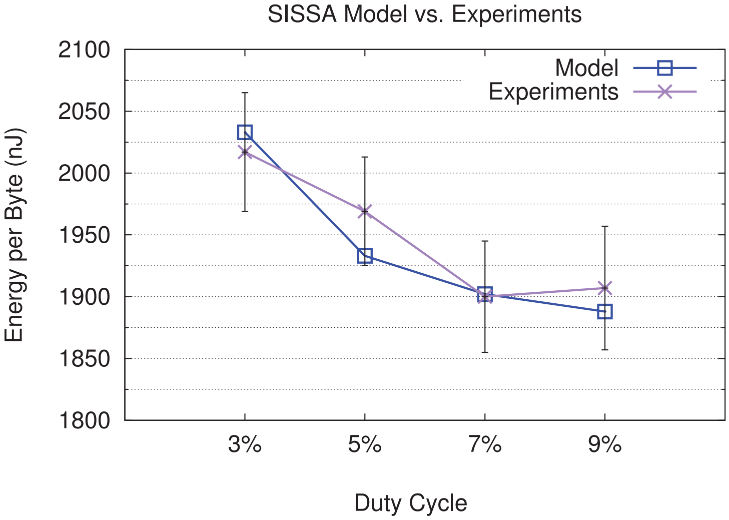

Fig. 11. Energy (nJ) per byte.

Table III suggests that the energy spent during the swarm phase increases as the duty-cycle decreases. This is because the maximum swarm phase convergence time increases as the duty-cycle decreases (as shown in Table IV) and so does the energy consumption. As expected, the convergence time of SISSA does not increase with the number of nodes and depends on the duty-cycle parameter only.

Finally, to determine the efficiency of SISSA swarm phase, Figure 10 and Figure 11 show respectively the average energy (measured in mJ) spent by each of the $k_{des} = 5$ sensors during the communication phase as a function of sensor nodes' duty cycle. Obviously, as the time available for data exchange increases, the energy spent by the k_{des} nodes increases. However, the energy per byte (measured in nJ) spent decreases as the duty cycle increases (result not shown here due to space limitations). Such results

Table V. Experimental Convergence Ratio in Function of TPL (Transmission Power Level) and R

	Transmission Powel Level											
R	6	CI	9	CI	12	CI	15	CI	18	CI	21	CI
7m	0.444	±0.091	0.974	±0.048	0.996	±0.045	0.996	±0.021	0.994	±0.032	0.996	±0.012
10m	0.190	±0.082	0.928	±0.056	0.906	±0.037	0.956	±0.012	0.984	±0.016	0.994	±0.015
13m	0.218	±0.079	0.894	±0.026	0.956	±0.089	0.960	±0.018	0.966	±0.043	0.990	±0.021
15m	0.150	±0.086	0.894	±0.075	0.982	±0.032	0.990	±0.009	0.990	±0.024	0.986	±0.040

also conclude that the majority of the energy consumption by the k_{des} sensors during the overall data collection process is due to the data transfer phase (e.g., 11 mJ vs. 105 mJ when the duty cycle is set to 3%). Therefore, the additional energy spent by the SISSA algorithm in the swarm phase is (much) less than the energy spent by the sensor node while communicating, thus making the algorithm highly scalable and energy efficient even with dense deployment.

7.2. Outdoor Experiments

The second set of experiments we conducted was aimed at evaluating the impact of the transmission power level of swarm agents and the distance between sensors on the performance of SISSA. In particular, in these experiments we set up an outdoor experimental setup composed by 10 TelosB sensors deployed in a ring network configuration with radius R. Sensors were deployed with an angle of about 36° between them. As in the first set of experiments, the mobile sink speed was about 2m/s, and 50 MS tours were performed. To measure the performance of SISSA, we define the *convergence ratio* as the number of MS tours in which SISSA converged (i.e., a timeout did not happen) divided by the total number of MS tours. A value of convergence ratio close to 1 indicates that SISSA converged most of the time in the given experimental setup. Table V summarizes the convergence values obtained in the outdoor experimental set with different values of the transmission power level[4] of the CC2420 radio transceiver [Chipcon 2004] and radius R of the ring network. As expected, Table V concludes that when the transmission power of swarm agents is low and the distance is relatively high, SISSA shows low convergence rate. However, as the transmission power increases, SISSA obtains very high convergence ratio. This concludes that the transmission power of swarm agents must be set accordingly to the distance between nodes to ensure convergence of the SISSA algorithm.

8. OPTIMIZATION AND COMPARISON RESULTS

In this section, we analytically investigate the network lifetime as provided by SISSA, and compare it with the ideal *Oracle* scheme defined in Section 3. Furthermore, we report the results obtained by comparing the performance of SISSA with respect to the IEEE 802.15.4 ZigBee carrier sense multiple access/collision avoidance (CSMA/CA) medium access control (MAC) protocol (hereafter referred to as *802.15.4*) [Society 2006], as well as with respect to the time division multiple access (hereafter referred to as *TDMA*) scheme. We chose 802.15.4 as it is currently the reference communication technology for wireless sensor networks (WSNs); TDMA has been chosen given its remarkable capability to efficiently achieving high throughput. If not specified otherwise, then the analytical parameters used are the same as listed in Table II. Without loss of generality, we have estimated the average contact time as 40s, corresponding to a radio communication range of about 40m and a linear speed of 2m/s (average human walking speed).

[4] The transmission power range of the CC2420 transceiver spans from 1 to 31, which respectively correspond to about 14.70mW and 31.32mW when the voltage is 1.8V.

Table VI. Optimum Duty-Cycle Values δ^*

Bound	$k_{des} = 1$	$k_{des} = 2$
$\theta_{des} \geq 5s$	0.6%	0.7%
$\theta_{des} \geq 10s$	0.7%	1.1%
$\theta_{des} \geq 12.5s$	0.75%	1.6%
$\theta_{des} \geq 15s$	0.85%	2.1%

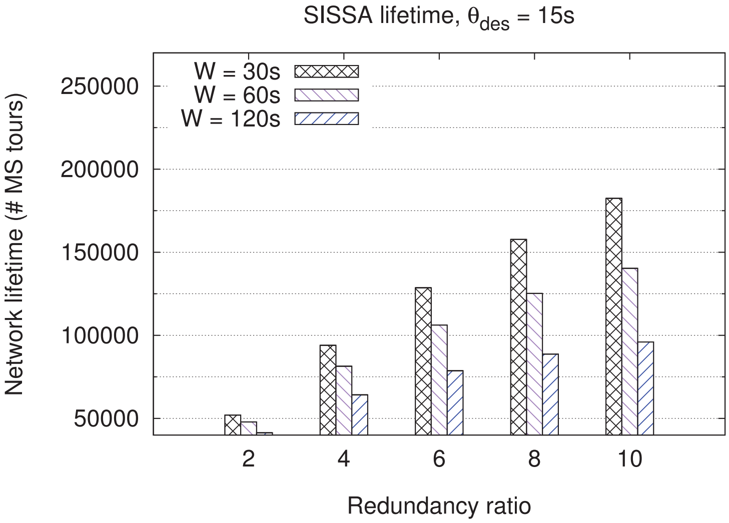

Fig. 12. Network lifetime, $\theta_{des} = 15s$.

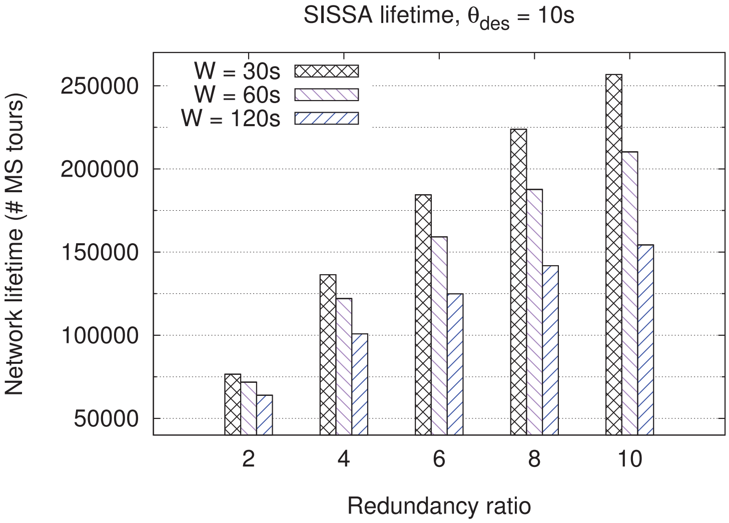

Fig. 13. Network lifetime, $\theta_{des} = 10s$.

Table VI shows the optimized duty cycles obtained from SISSA with varying number of k_{des} and θ_{des}. Furthermore, Figures 12 and 13 show the network lifetime with $\theta_{des} = 15s$ and $\theta_{des} = 10s$, respectively (both shown for $k_{des} = 2$). Results in these figures are shown for varying number of the redundancy ratio R of nodes and waiting time W,

Table VII. Lifetime Approximation Ratio Results

	$W = 30s$				
θ_{des}	R = 2	R = 4	R = 6	R = 8	R = 10
5s	75.80	61.16	51.21	44.05	38.65
10s	87.26	77.41	69.55	63.14	57.81
12.5s	89.01	80.20	72.79	66.94	61.93
15s	89.38	80.79	73.71	67.77	62.72
	$W = 60s$				
θ_{des}	R = 2	R = 4	R = 6	R = 8	R = 10
5s	70.76	54.75	44.65	37.69	32.61
10s	81.89	69.33	60.12	53.06	47.49
12.5s	82.58	70.33	61.24	54.23	48.66
15s	82.33	69.97	60.84	53.81	48.24
	$W = 120s$				
θ_{des}	R = 2	R = 4	R = 6	R = 8	R = 10
5s	62.32	46.18	36.38	30.02	25.55
10s	73.23	57.77	47.70	40.62	35.37
12.5s	72.15	56.43	46.34	39.31	34.13
15s	71.13	55.19	45.09	38.11	33.00

which is the time spent by the sensors in the discovery process before the MS enters the communication area. Both figures show that the lifetime provided by SISSA increases as θ_{des} decreases. This was expected, since we have shown in Equation (13) that lower energy spent in the communication phase corresponds to higher network lifetime. In addition, both figures conclude that SISSA is able to exploit very well the redundancy ratio to increase lifetime, especially when the waiting time becomes smaller. This is because the additional energy spent in the discovery phase impacts negatively on the overall energy consumptions of sensors, rendering the SISSA algorithm less effective.

However, Table VI points out that SISSA guarantees every θ_{des} constraint using a relatively low duty cycle (maximum 2.1%). The reasons behind these results are summarized as follows. First, by guaranteeing contention-free access, SISSA allows sensor nodes to have the same channel access time, irrespective of the number of nodes in the WSN. Second, since SISSA allows only a subset of nodes to communicate during each visit, the time slot allocated to each node becomes larger. Therefore, SISSA is able to guarantee stringent constraints on the θ_{des} value by using a relatively low duty cycle (not shown here for the sake of space). In other words, SISSA is energy efficient and optimizes network lifetime without compromising the QoS guarantees required by the sensing application. In addition, we would like to remark here that optimizing the energy consumption during the MS discovery process is not the primary target of this paper. In particular, other techniques (e.g., learning-based [Shah et al. 2011; Kondepu et al. 2012] or hierarchical discovery [Restuccia et al. 2012]) can be used on top of SISSA to further reduce the duty cycles adopted in the MS discovery and, therefore, further increase the network lifetime.

Table VII reports the LAR provided by SISSA as compared to the ideal, optimum Oracle scheme. Recall that the LAR was defined as the ratio between the lifetime provided by SISSA and that provided by Oracle. In Table VII the LAR has been expressed as a percentage value. Results in Table VII are shown with varying waiting time W and redundancy ratio R, considering $k_{des} = 2$. Overall, Table VII concludes that SISSA approximates better the ideal Oracle scheme when the MS discovery process takes shorter time and the redundancy ratio is lower. This is because Oracle does not include MS discovery, which is instead necessary in real-world WSNs implementations. Therefore, the LAR of SISSA decreases as the value of the redundancy ratio R and the waiting time W increases. Nevertheless, SISSA provides high LAR level in most of the considered network parameters, achieving an average of 56.9% for the set of network parameters considered.

Surprisingly enough, and conversely to what Figure 13 concludes, Table VII demonstrates that SISSA presents higher LAR when the constraint on θ_{des} is more stringent, especially for low W (e.g., 90.85 vs. 77.18 when $R = 2$ and $W = 30s$). This, however, can be explained as follows. When θ_{des} is higher and W is lower, the contribution to the energy consumption is due to the energy consumed in the communication phase. Therefore, in this case the SISSA algorithm approximates better the *Oracle* scheme, given the energy consumption in the discovery process is negligible. However, the *value* of the network lifetime is lower in the case of $\theta_{des} = 15s$, as Figure 12 points out.

Another interesting aspect exhibited by Table VII is that, when $W =$ 60s and $W =$ 120s, the LAR value decreases as θ_{des} increases from 12.5s to 15s, which does not happen when $W =$ 30s. This is explained by the definition of LAR presented in Equation (14). In fact, when the energy spent during MS discovery becomes significant, the increase in the energy spent in the communication phase due to the increase in the θ_{des} value is counterbalanced by the increase in the energy spent during MS discovery. In other words, higher θ_{des} implies higher LAR only when the waiting time is relatively small.

8.1. Comparison with IEEE 802.15.4 and TDMA

In this section, we show the results obtained by comparing SISSA with the 802.15.4 and TDMA schemes. Radio parameters are the same as listed in Table II. In these experiments, SISSA results were obtained from the analytical model, while 802.15.4 and TDMA were simulated. For the 802.15.4 parameters, we chose the standard parameter set (i.e., *macMinBE* = 3, *macMaxBE* = 5, *macMaxCSMABackoffs* = 4, *macMaxFrameRetries* = 3), as specified by the standard [Society 2006]. If not specified otherwise, then the value of the waiting time is $W =$ 5min. As in Section 8, we estimated the minimum contact time as 40s. In all experiments we performed at least 10,000 MS tours. Confidence intervals have been omitted when below 3%.

Figures 14 and 15, respectively, show the energy spent per byte successfully transmitted and throughput (calculated without considering the headers' overhead) obtained by SISSA, 802.15.4 and TDMA for varying number of nodes in the subarea. In order to evaluate the efficiency of SISSA with respect to different QoS constraints on θ_{des} and k_{des}, the different bars of SISSA were obtained by considering the optimal duty-cycle values in Table VI. By considering the parameters in Table II, the bounds of $\theta_{des} =$ 5s, 10s, and 12s correspond, approximately, to 100, 200, and 400kB, respectively. In addition, we considered the fixed duty-cycle values of 5%, 7%, and 10% for 802.15.4 and TDMA to allow sufficient available transmission time per MS tour. Figure 14 concludes that SISSA and TDMA are energy-efficient irrespective of the number of nodes, while 802.15.4 becomes inefficient as the network size assumes significant values. This is because SISSA and TDMA provide a contention-free channel access to the sensor nodes, allowing them to transmit their data efficiently regardless of the number of nodes considered.

However, as Figure 15 points out, both 802.15.4 and TDMA fail to provide any of the minimum throughput constraints required by the application (i.e., 100, 300, and 400kB), irrespective of the considered duty-cycle or the number of nodes. This is because, as anticipated, 802.15.4 suffers from the well-known "MAC unreliability problem" (discussed in detail in Anastasi et al. [2011]), so it becomes more and more inefficient as the network size increases. On the other hand, TDMA preallocates slots to *each* sensor in the subarea, and hence the time reserved by TDMA to each node decreases as the number of nodes increases, as so does the throughput per each node. Conversely, as shown in Table VI, SISSA guarantees all throughput constraints using

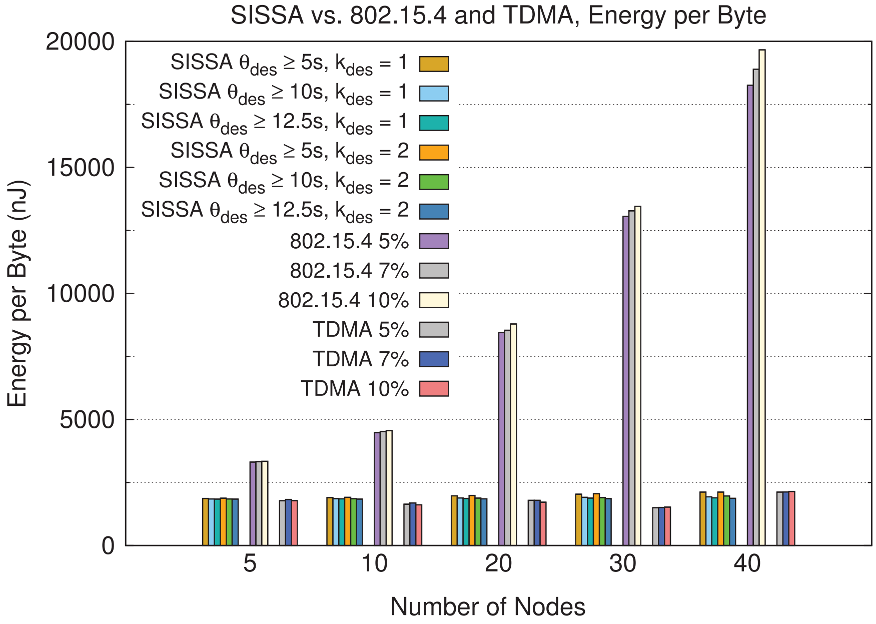

Fig. 14. Energy per byte, SISSA vs. 802.15.4 and TDMA.

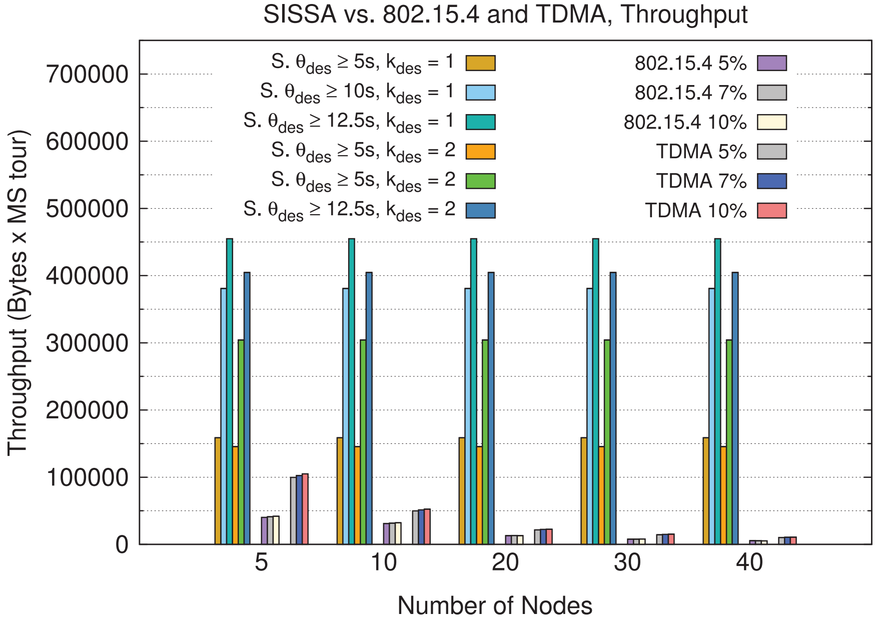

Fig. 15. Throughput, SISSA vs. 802.15.4 and TDMA.

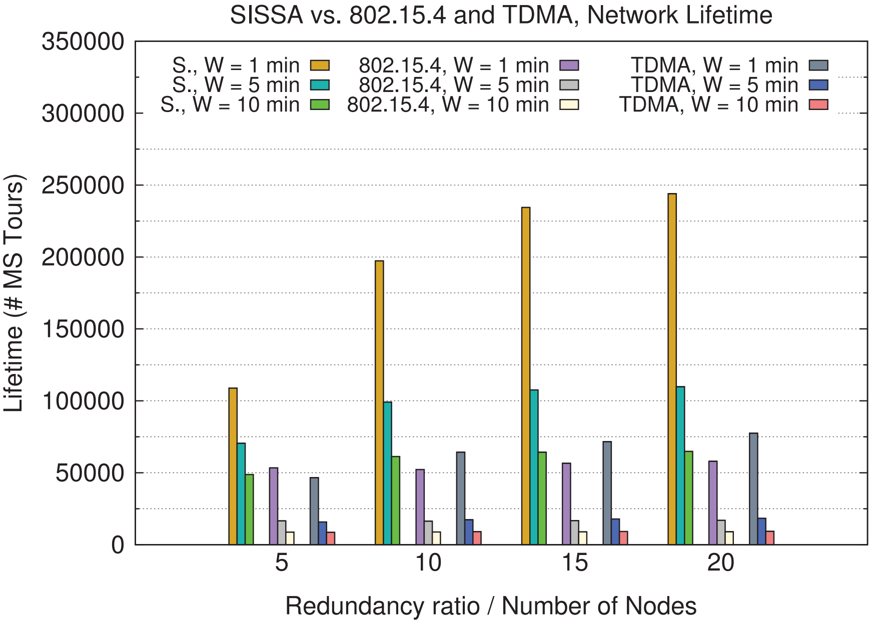

Fig. 16. Network Lifetime, SISSA vs. 802.15.4 and TDMA.

a (much) lower duty cycle than both TDMA and 802.15.4. The reasons behind these results are summarized as follows.

First, by guaranteeing contention-free access to selected nodes, SISSA allows sensor nodes to attain a very high throughput, irrespective of the number of nodes considered. Second, since SISSA allows just a subset of nodes to communicate during each tour, the time slot allocated to each node becomes larger as compared to TDMA. Therefore, SISSA is able to guarantee a much higher throughput while using a lower duty cycle (as shown in Table VI) when both $k_{des} = 1$ and $k_{des} = 2$.

The importance of having a very low duty cycle while in the discovery phase is exhibited by Figure 16, in which we report the network lifetime yielded by SISSA, 802.15.4, and TDMA. For the sake of space, we report only the case of $\delta = 0.6\%$, corresponding to $\theta_{min} \geq 5s$ and $k_{des} = 1$. Figure 16 depicts the results obtained with 802.15.4 and TDMA considering a duty cycle of 5%. These results conclude that SISSA outperforms both 802.15.4 and TDMA irrespective of the number of nodes and the waiting time.

From Figure 16 we observe that the lifetime improvement provided by increasing the redundancy ratio becomes more effective as the waiting time becomes smaller. This is derived from the fact that the maximum lifetime provided by SISSA is inversely proportional to the energy spent in the discovery phase. Therefore, as the waiting time increases, the energy spent during the discovery phase becomes predominant as compared to the energy spent during the communication phase; thus SISSA becomes less effective in order to extend lifetime. Note that this behavior is also experienced by 802.15.4 and TDMA.

Nevertheless, SISSA is able to improve network lifetime by several times in each considered scenario. For example, considering redundancy ratio of $R = 15$ and waiting time of $W = 10$min, SISSA increases the network lifetime by 6.03× and 5.80× with respect to 802.15.4 and TDMA, respectively. Even considering $\theta_{des} \geq 12.5$s, $k_{des} = 2$, and $S = 5$ (not shown in Figure 16 for space limitation), SISSA improves lifetime by

76% and 72% with respect to 802.15.4 and TDMA. It is noteworthy that SISSA achieves these enhancements of lifetime while guaranteeing a *much* larger throughput, whereas 802.15.4 and TDMA fail to guarantee any of these QoS constraints to the sensing application (see Figure 15). This is because SISSA uses a lower duty cycle while in the discovery phase, as shown in Table VI.

From Figure 16 it also appears that 802.15.4 and TDMA increase network lifetime when the network size grows. However, as depicted in Figure 15, this is simply related to the fact that the network throughput provided by 802.15.4 and TDMA decreases when the network size increases and so does the energy consumption. This misbehavior allows sensor nodes to consume less energy and then increase network lifetime.

9. CONCLUSIONS

In this paper, we have formulated the problem of optimizing the lifetime of a WSN in the presence of uncontrollable sink mobility and QoS constraints on throughput and data reliability. We first proposed the optimal *Oracle* scheme, which is proved to maximize the network lifetime. Next, we proposed SISSA, a novel algorithm based on the swarm intelligence which optimizes network lifetime. By deriving a comprehensive analytical model, we have derived performance parameters such as energy consumption and minimum throughput guarantees, we have proven bounds on energy consumption, number of swarm agents exchanged, and the convergence time. We have also derived an approximate formula to estimate the network lifetime yielded by SISSA under different network parameters. We have validated our model through experiments on a real sensor network experimental setup, as well as compared the performance of SISSA with respect to the *Oracle* scheme, as well as with the IEEE 802.15.4 and TDMA schemes. Results show that SISSA can effectively and energy-efficiently guarantee strict QoS constraints and approximates well the ideal scheme. This concludes that SISSA is suitable to most of the existing sensing scenarios in which the MS mobility is uncontrollable and yet QoS needs to be guaranteed.

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers for insightful comments which helped us improve the quality of the manuscript significantly.

REFERENCES

Ian F. Akyildiz and Mehmet Can Vuran. 2010. *Wireless Sensor Networks*. Vol. 4. John Wiley & Sons, New York, NY.

Giuseppe Anastasi, Marco Conti, and Mario Di Francesco. 2011. A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks. *IEEE Trans. Indust.* 7, 1 (February 2011), 52–65. DOI:http://dx.doi.org/10.1109/TII.2010.2085440

Eleonora Borgia, Giuseppe Anastasi, and Marco Conti. 2013. Energy efficient and reliable data delivery in urban sensing applications: A performance analysis. *Comput. Networks* 57, 17 (2013), 3389–3409. DOI:http://dx.doi.org/10.1016/j.comnet.2013.07.025

Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Hong Lu, Xiao Zheng, Mirco Musolesi, Kristóf Fodor, and Gahng-Seop Ahn. 2008. The rise of people-centric sensing. *IEEE Internet Comput.* 12, 4 (2008), 12–21.

Angelo Cenedese, Andrea Zanella, Lorenzo Vangelista, and Michele Zorzi. 2014. Padova smart city: An urban internet of things experimentation. In *Proceedings of the 2014 IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 1–6.

Arnab Chakrabarti, Ashutosh Sabharwal, and Behnaam Aazhang. 2003. Using predictable observer mobility for power efficient design of sensor networks. In *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN'03)*. Springer-Verlag, Berlin, 129–145. http://dl.acm.org/citation.cfm?id=1765991.1766001

Chipcon. 2004. 2.4 GHz IEEE 802.15.4/ZigBee-Ready RF Transceiver. (2004). Retrieved December 4, 2014 from http://inst.eecs.berkeley.edu/∼cs150/Documents/CC2420.pdf. Chipcon Products from Texas Instruments.

Crossbow. 2014. TelosB MOTE platform. (2014).

Mario Di Francesco, Sajal K. Das, and Giuseppe Anastasi. 2011. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Trans. Sensor Networks (TOSN)* 8, 1 (2011), 7.

Andries P. Engelbrecht. 2006. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, New York, NY.

Shuai Gao, Hongke Zhang, and Sajal K. Das. 2011. Efficient data collection in wireless sensor networks with path-constrained mobile sinks. *IEEE Trans. Mobile Comput.* 10, 4 (2011), 592–608.

Yu Gu, Yusheng Ji, Jie Li, and Baohua Zhao. 2013. ESWC: Efficient scheduling for the mobile sink in wireless sensor networks with delay constraint. *IEEE Trans. Parallel Distrib. Syst.* 24, 7 (July 2013), 1310–1320. DOI:http://dx.doi.org/10.1109/TPDS.2012.210

Yu Gu, Fuji Ren, Yushen Ji, and Jie Li. 2015. The evolution of sink mobility management in wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* (2015). DOI:http://dx.doi.org/10.1109/COMST.2015.2388779

Zygmunt J. Haas and Tara Small. 2006. A new networking model for biological applications of ad hoc sensor networks. *IEEE/ACM Trans. Networking* 14, 1 (2006), 27–40.

Liang He, Jianping Pan, and Jingdong Xu. 2013. A progressive approach to reducing data collection latency in wireless sensor networks with mobile elements. *IEEE Trans. Mobile Comp.* 12, 7 (July 2013), 1308–1320. DOI:http://dx.doi.org/10.1109/TMC.2012.105

Mikhail Itskov. 2007. *Tensor Algebra and Tensor Analysis for Engineers*. Springer, Berlin, 410–420.

Abdul Waheed Khan, Abdul Hanan Abdullah, Mohammad Hossein Anisi, and Javed Iqbal Bangash. 2014. A comprehensive study of data collection schemes using mobile sinks in wireless sensor networks. *Sensors* 14, 2 (2014), 2510–2548.

Koteswararao Kondepu, Francesco Restuccia, Giuseppe Anastasi, and Marco Conti. 2012. A hybrid and flexible discovery algorithm for wireless sensor networks with mobile elements. In *2012 IEEE Symposium on Computers and Communications (ISCC)*. 295–300.

Xu Li, Jiulin Yang, Amiya Nayak, and Ivan Stojmenovic. 2012. Localized geographic routing to a mobile sink with guaranteed delivery in sensor networks. *IEEE J. Select. Areas Commun.* 30, 9 (October 2012), 1719–1729. DOI:http://dx.doi.org/10.1109/JSAC.2012.121016

Wang Liu, Kejie Lu, Jianping Wang, Guoliang Xing, and Liusheng Huang. 2012. Performance analysis of wireless sensor networks with mobile sinks. *IEEE Trans. Vehic. Technol.* 61, 6 (2012), 2777–2788.

Brigitte Plateau and Karim Atif. 1991. Stochastic automata network of modeling parallel systems. *IEEE Tran. Software Eng.* 17, 10 (1991), 1093–1108.

Francesco Restuccia, Giuseppe Anastasi, Marco Conti, and Sajal K. Das. 2012. Performance analysis of a hierarchical discovery protocol for WSNs with mobile elements. In *2012 IEEE 13th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 1–9.

Francesco Restuccia, Giuseppe Anastasi, Marco Conti, and Sajal K. Das. 2014. Analysis and optimization of a protocol for mobile element discovery in sensor networks. *IEEE Trans. Mobile Comput.* 13, 9 (Sept 2014), 1942–1954. DOI:http://dx.doi.org/10.1109/TMC.2013.88

Kunal Shah, Mario Di Francesco, Giuseppe Anastasi, and Mohan Kumar. 2011. A framework for resource-aware data accumulation in sparse wireless sensor networks. *Comput. Commun.* 34, 17 (2011), 2094–2103.

Lei Shi, Baoxian Zhang, Hussein T. Mouftah, and Jian Ma. 2013. DDRP: An efficient data-driven routing protocol for wireless sensor networks with mobile sinks. *Int. J. Commun. Syst.* 26, 10 (2013), 1341–1355. DOI:http://dx.doi.org/10.1002/dac.2315

IEEE Computer Society. 2006. IEEE Standard for Information Technology, Part 15.4; Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). Available at https://standards.ieee.org/findstds/standard/802.15.4-2006.html.

Farzad Tashtarian, M. H. Yaghmaee Moghaddam, Khosrow Sohraby, and Sohrab Effati. 2015. On maximizing the lifetime of wireless sensor networks in event-driven applications with mobile sinks. *IEEE Trans. Vehic. Technol.* 64, 7 (2015), 3177–3189. DOI:http://dx.doi.org/10.1109/TVT.2014.2354338

John Tooker and Mehmet C. Vuran. 2012. Mobile data harvesting in wireless underground sensor networks. In *2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, Washington DC, 560–568.

Can Tunca, Sinan Isik, M. Yunus Donmez, and Cem Ersoy. 2014. Distributed mobile sink routing for wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* 16, 2 (2014), 877–897. DOI:http://dx.doi.org/10.1109/SURV.2013.100113.00293

Zichuan Xu, Weifa Liang, and Yinlong Xu. 2012. Network lifetime maximization in delay-tolerant sensor networks with a mobile sink. In *Proceedings of the 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 9–16.

Sheng Yu, Baoxian Zhang, Cheng Li, and Hussein T. Mouftah. 2014. Routing protocols for wireless sensor networks with mobile sinks: A survey. *IEEE Commun. Mag.* 52, 7 (2014), 150–157.

Pei Zhang, Christopher M. Sadler, Stephen A. Lyon, and Margaret Martonosi. 2004. Hardware design experiences in ZebraNet. In *Proceedings of the 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys)*. 227–238.

Received December 2014; revised October 2015; accepted January 2016