

# Heuristic Approaches for Transmission Scheduling in Sensor Networks with Multiple Mobile Sinks

DAMLA TURGUT AND LADISLAU BÖLÖNI\*

*School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL  
32816–2450, USA*

*\*Corresponding author: lboloni@eecs.ucf.edu*

**A large part of the energy budget of traditional sensor networks is consumed by the hop-by-hop routing of the collected information to the static sink. In many applications it is possible to replace the static sink with one or more mobile sinks that move in a sensor field and collect the data through one-hop transmissions. This greatly reduces the power consumption of the nodes, which can be further reduced by choosing the appropriate moment of transmission. In general, the transmission energy increases quickly with the distance, and thus it makes sense for the nodes to transmit when one of the mobile sinks is in close proximity. Seeing the node as an autonomous agent, it needs to choose its actions of transmitting or buffering the collected data based on what it knows about the environment and its predictions about the future. The sensor agent needs to appropriately balance the following two objectives: the maximization of the utility of the collected and transmitted data and the minimization of the energy expenditure. We introduce the cumulative policy penalty as an expression of this interdependent pair of requirements. As a baseline, we describe a graph-theory-based approach for calculating the optimal policy in a complete knowledge setting. Then, we describe and compare three heuristics based on different principles (imitation of human decision making, stochastic transmission and constant risk). We compare the proposed approaches in an experimental study under a variety of scenarios.**

*Keywords: agents; sensor networks; mobile sink*

*Received 30 July 2009; revised 30 July 2009*

*Handling editor: Alex Rogers*

## 1. INTRODUCTION

Traditional sensor networks are composed of a set of low-power sensor nodes that collect environmental data and forward it by hop-by-hop routing to one or more sinks. Sinks are assumed to have much more computational power and energy resources than the sensor nodes. The traditional vision of a sensor network assumed both the sinks and the sensor nodes to be static. Because of the low-power resources of the sensor nodes, energy conservation is an important factor. Most of the energy of the node is spent for the wireless transmissions. Nodes in the proximity of the sink need to forward the transmissions of the nodes from the exterior part of the network. For such a node, the energy expenditure of forwarding can be several times larger than the cost of transmitting the node's own observations.

An alternative approach, more economical in terms of consumed power would be to collect the data with a set of *mobile sinks*, which are periodically visiting the vicinity of each node. The sensor nodes are collecting and buffering their observations, and occasionally transmitting them to the closest sink. Naturally, there will be moments when there is no mobile sink in the transmission range of the node. Even when a sink is in the transmission range, it might be so far that the transmission can happen only with large energy consumption. This creates a new problem for the sensor node: should it send the data now, or wait until a sink gets closer, allowing the transmission of the data with a lower power consumption? Given that the necessary transmission power increases very quickly with distance (in certain cases, for nodes close to the ground, it can be as much as the fourth power of distance),

the right choice of the transmission moment can be of major importance. Of course, if a sensor node waits too long, it might be forced to transmit at the moment when its memory buffer is full, while bypassing previous, better opportunities. Even worse, if there is no mobile sink in the transmission range when the buffer is full, some amount of observations will be lost. Those interested in energy consumption within sensor networks can refer to an extensive survey presented by Anastasi *et al.* [1].

In this paper, we describe and compare three practically implementable heuristic algorithms to control the transmission behavior of the nodes in the presence of mobile sinks. To provide a baseline for comparison, we also describe an optimal algorithm based on the transformation of the problem into a graph theory problem. The latter, however, has a limited practical applicability due to its information requirements (it requires advance knowledge of the movement patterns of the mobile sinks) and its considerable computational complexity.

The remainder of this paper is organized as follows. The transmission scheduling problem, its applications and possible strategies are described in Section 2. Related work in the domain of sensor networks with mobile sinks (MSSN) is presented in Section 3. In Section 4 we present the Oracle Optimal (OrOpt) algorithm, an algorithm which calculates the optimal schedule of transmissions providing that the movement schedule of the sinks is known ahead of time. While this requirement, together with the high memory and computational cost makes it less suitable for deployment on a sensor node, the algorithm will serve as a reference for the more realistic algorithms we present in the next sections. In Section 5 we present the three heuristic algorithms for transmission scheduling. In Section 6 we present the results of an experimental study. We conclude in Section 7.

## 2. THE TRANSMISSION SCHEDULING PROBLEM

The transmission scheduling problem for MSSN is centered on the decisions of the node whether or not to transmit its currently collected set of observations to a mobile sink at a particular moment in time.

MSSN have applications in areas ranging from environmental data collection to battlefield surveillance. The transmission scheduling problem appears in most of these deployments, although in slightly different formulations. For instance, our assumption is that data transmission is initiated by the sensor node, and thus the transmission scheduling problem needs to be solved by the node itself. If a certain architecture requires the data transmission to be initiated by the sink, the transmission scheduling problem must be solved by the sink. The only scenario when the transmission scheduling problem is irrelevant is when the mobile sink visits the sensor nodes regularly and positions itself at a predetermined location for receiving data.

In this paper, we make the following assumptions about the deployment scenario of the network:

- (i) The mobile sinks visit every sensor node; all the nodes will be eventually visited by a sink. This does not necessarily mean that all the data collected by the node can be transmitted to the sink; it is possible that the time interval between two visits is so large that even with an optimal strategy some data will be lost.
- (ii) The data transmission always happens between the sensor node and the closest mobile sink.
- (iii) The sink does not move during the transmission.
- (iv) The nodes have a finite buffer of constant size and collect observations with a constant bit rate.
- (v) The information does not become obsolete in the time it takes to fill the buffer. That is, the node can wait until its buffer is full and transmit the full buffer in a single transmission, without incurring penalties for delay. Of course, the node can also transmit when the buffer is only partially full.

Naturally, the algorithms proposed in this paper need to be appropriately modified if deployed under different assumptions, and the relaxation of some of these assumptions, such as allowing the sink to move during data transmission, adds additional complexity to the problem.

The sensor node, acting as an autonomous agent, needs to balance two separate goals as follows: the transmission of the observations to the mobile sink and minimization of the energy consumption. Neither of these two objectives alone would yield the desired behavior. Considering only the energy minimization criterion would create a sensor that does not transmit any observation. Considering only the goal to minimize data loss would create a system that will transmit at every available opportunity.

To consider both goals, we will require the agent to minimize a *cummulative policy penalty* (CPP) function which represents over a period of time the consumed energy and a weighted penalty for data loss. The *cummulative* aspect of this function is important, because of the delay between a decision and the moment when the associated penalty occurs. For instance, the decision of not transmitting at a certain moment will incur a data loss penalty only later, if no other opportunity to transmit occurred before the buffer was full. Unfortunately, this *cummulative* aspect means that the CPP cannot be used directly as a decision function, as it can only be evaluated retrospectively. The *policy* aspect of this function reflects the fact that the balance of the two components can be set arbitrarily by the beneficiary of the system.

Let us assume that the sensor network operates over a timespan  $[0, T]$  during which the considered node performs  $k$  transmissions, with transmission  $i \in 1 \dots k$  being performed at time  $t_i$  with the buffer content being  $b_i$  and the distance to the mobile sink  $d_i$ . We denote by  $b_{nt}$  the amount of data that was not transmitted to the mobile sink. The CPP can be expressed

as follows:

$$\text{CPP} = \sum_{i \in 1 \dots k} b_i \cdot p_{\text{tx}}(d_i) + w_p \cdot b_{\text{nt}}. \quad (1)$$

If the data gathering rate is  $r$ , we have

$$rT = \sum_{i \in 1 \dots k} b_i + b_{\text{nt}}. \quad (2)$$

The transmission energy is fully determined by the physical factors. We use the following model for the energy dissipation used for communicating one bit of data [2]:

$$p_{\text{tx}}(d) = (\alpha_{11} + \alpha_2 d^n), \quad (3)$$

where  $p_{\text{tx}}$  is the power dissipated when the node is transmitting to the mobile sink,  $d$  is the distance to the sink and  $n$  is the path loss index, and  $\alpha_{11}$  and  $\alpha_2$  are positive constants. The path loss index varies between 2 and 4 depending on the environment and the position of the node. In general, for sensor networks deployed on the ground, the path loss index is higher. In our experimental study, we assume a path loss index of 4. Typical values of the parameters are  $\alpha_{11} = 45$  nJ/bit and  $\alpha_2 = 0.001$  pJ/bit/m<sup>4</sup> (for  $n = 4$ ).

Let us now consider the choice of the data loss penalty parameter  $w_p$ . At first sight, this is a free policy parameter, which can be set arbitrarily by the customer of the sensor network. Unfortunately, such free parameters are a significant headache for the users, because we lack the intuition of what a particular setting of this parameter means and subtle changes in the parameter can trigger unexpected emergent behavior in the system. The solution is to *anchor* this parameter to intuitive physical properties of the system. In the following we present two approaches for this anchoring, both of them relying on bringing the two components of the CPP to the same dimensionality.

## 2.1. Express the consumed power in terms of future data loss

Let us assume that the sensor node starts with an initial energy reserve  $P_{\text{total}}$ , and the life expectancy of the sensor network is  $T_{\text{exp}}$ . Excessive energy consumption in the present would make the node exhaust its energy before the expected lifetime, losing all the data which would have been collected afterward. We thus need to balance the data loss currently  $b_{\text{nt}}$  with the data loss at the end of the lifecycle  $b_{\text{eol}}$ .

The average energy consumption per unit of time is given by

$$p_{\text{avg}} = \frac{\sum_{i \in 1 \dots k} b_i \cdot p_{\text{tx}}(d_i)}{T}. \quad (4)$$

Assuming that the node will consume energy at its current rate, the lifetime of the node will be as follows:

$$T_{\text{life}} = \frac{P_{\text{total}}}{p_{\text{avg}}} = T \frac{P_{\text{total}}}{\sum_{i \in 1 \dots k} b_i \cdot p_{\text{tx}}(d_i)} \quad (5)$$

and the data loss at the end of the life cycle is given by

$$b_{\text{eol}} = \min(0, r(T_{\text{exp}} - T_{\text{life}})). \quad (6)$$

Now we can design a variant of the CPP with a version expressed only in terms of data loss as follows:

$$\text{CPP}_{\text{dataloss}} = b_{\text{eol}} + b_{\text{nt}} \frac{\min(T_{\text{exp}}, T_{\text{life}})}{T}. \quad (7)$$

In this model, the free parameter is the expected lifetime of the network  $T_{\text{exp}}$ . This parameter is intuitive and easy to set based on the functional specifications of the deployment.

## 2.2. Express the data loss in terms of alternative forwarding techniques

The previous approach requires us to set the expected lifetime of the network. In the case when the deployment model is a traditional sensor network with a static sink *augmented* with several mobile sinks (a reasonable assumption), we can express the cost of the data not collected by the mobile sinks by the cost of transmitting the data using traditional hop-by-hop routing to the static sink. Let us assume that the hop count to the static sink is  $n_{\text{hops}}$  and the average distance between nodes is  $d_{\text{avg}}$ . Then, we can express the penalty component of the CPP as follows:

$$w_p \cdot b_{\text{nt}} = b_{\text{nt}} \cdot n_{\text{hops}} \cdot p_{\text{tx}}(d_{\text{avg}}) \quad (8)$$

that is

$$w_p = n_{\text{hops}} \cdot p_{\text{tx}}(d_{\text{avg}}). \quad (9)$$

In this case, there is no free variable at all in the expression of the CPP, which can be calculated directly from the physical properties of the deployment. Under these assumptions, the CPP will differ from node to node. Sensor nodes close to the static sink would have a relatively low penalty for data transmitted through the hop-by-hop method. Sensor nodes many hops away from the static sink would need to take more effort to send their data through the mobile sinks.

## 3. RELATED WORK

The traditional view of wireless sensor networks was based on the assumption of fixed sinks and multi-hop routing in which every sensor node participates. However, forwarding other nodes' packets puts a very significant load on the limited energy resources of the sensor nodes. Significant research effort was spent on routing techniques that reduce and balance the energy consumption of the sensor network.

Recently, several research groups proposed approaches based on the assumption of mobile sinks. Whenever their deployment is possible, mobile sinks can greatly extend the lifetime of the sensor network. In the best case, the mobile sinks periodically visit the vicinity of every sensor; in these conditions, all the

communication happens in a single hop between the node and the mobile sink.

Naturally, the use of mobile sinks opens a number of new research challenges. In the following, we review some of these efforts grouped by the research problems on which they concentrate.

### 3.1. Routing toward mobile sinks

These types of networks assume that only a subset of sensor nodes are visited by the sinks. The nodes that do not have direct access to the sink are using hop-by-hop routing either toward the mobile sink or toward sensor nodes that are periodically visited by a sink.

The MULE (Mobile Ubiquitous LAN Extension), by Shah *et al.* [3], architecture has the following three tiers: (i) a top tier of WAN connected devices, (ii) a middle tier of mobile transport agents and (iii) a bottom tier of fixed wireless sensor nodes. The mobile transport agents, which are the equivalents of mobile sinks, are opportunistic agents capable of short range wireless communication with the sensors and wireless access points. The agents use Markov chain theory to determine the average values of the entities of interest.

Anastasi *et al.* [4] consider the transport layer implications of a sensor transmitting data to a moving mobile sink. They start from the observation that the lowest error rate occurs at the point when the sink is closest to the sensor, which should be at the midpoint of the contact interval. The proposed adaptive data transfer protocol schedules the transfer symmetrically around the estimated midpoint of the contact interval and uses the automatic repeat request mechanism to recover from the transmission errors.

Kim *et al.* [5] introduced scalable energy-efficient asynchronous dissemination which is a distributed self-organizing protocol that reduces the energy consumption by the construction of a dissemination tree (*d-tree*) and the dissemination of the data to the mobile sinks.

Baruah *et al.* [6] aims to deliver sensor data toward a mobile sink over multiple hops by hybrid learning-enforced time-domain routing. The mobile sink does not query for data but rather passively listens for data ‘pushed’ by the source sensor. The sensor nodes are forwarding their observations towards the *moles*, sensor nodes located within the sink’s path.

### 3.2. Mobility models of the sinks

The mobility of the sink can be categorized into the following three types: random, predictable and controllable. In case of *random mobility*, the sink travels through the network in a random walk fashion. In the case of *predictable mobility*, the sensor nodes can learn the mobility pattern of the sink and therefore can predict the location of the sink at any given point in time. In the case of *controlled mobility* the sink mobility

is adaptively controlled based on specific parameters of the network and/or the deployed applications.

A model for controlled mobility is presented in Basagni *et al.* [7]. A linear optimization model is used to determine which nodes the single mobile sink visits and for how long. The authors find that the energy depletion was more balanced across the network and the network lifetime was extended up to five times compared with a network with a static sink.

The data collection process is modeled as a queuing system in Chakrabarti *et al.* [8] to measure the impact of predictable observer mobility (where the observers correspond to mobile sinks). The network uses only single-hop communication. The authors show that predictable mobility can save communication power in the sensor network. Knowing the path of the sink can help the sensor and the sink find positions where they can exchange data with the lowest possible power.

Vincze *et al.* [9] proposes two sink location algorithms to minimize the overall energy consumption in the network (mintotal) as well as to minimize the energy consumption of the most loaded sensor (minmax).

The SENSOR NETWORKS WITH MOBILE AGENTS (SENMA) architecture, by Mergen *et al.* [10], is suitable for power-constrained large-scale dense sensor networks. SENMA relies on one-hop transmission between the sensor nodes and mobile agents. For communication, the system uses a slotted time division duplexing system with opportunistic ALOHA. The opportunistic ALOHA turns off the sensor automatically when the mobile agent is no longer in the proximity of the sensor.

The goal of the two-tier data dissemination (TTDD) protocol, by Luo *et al.* [11], is to provide scalable and efficient data delivery to multiple mobile sinks. TTDD uses a grid structure in which only the sensors placed in the grid points are required to obtain information for forwarding. Nodes nearby the grid points (dissemination nodes) receive queries from the mobile sink. The queries travel through the grid and data is forwarded back to the sinks by tracing the reverse path. As TTDD forwards data only to a fraction of the sensor nodes, it allows a lower control overhead.

### 3.3. Mobility and routing

This category combines projects which consider not only the mobility of the sink, but also routing of the sensed data towards the sink.

The mobile enabled wireless sensor networks (mWSN) architecture, introduced by Chen *et al.* [12], uses multi-hop forwarding to form a cluster around the expected position of the mobile sink. mWSN has two operational modes: local and remote sensing. In local sensing, once a mobile sink receives a response to a query sent to the fixed sensors, the collected data is transferred to the base station for interpretation. The query result will then be returned to the mobile sink. In the remote sensing case, multiple mobile sinks help gather the data

of interest. In this protocol, the sink trajectory is not controlled but rather it can be estimated or learned.

Kansal *et al.* [13] proposed the use of controlled and coordinated motion of network elements to alleviate resource limitations and improve system performance by adapting to deployment demands.

In Gandham *et al.* [14], multiple mobile stations are deployed to extend the lifetime of the sensor network, which is divided into equal periods of time known as *rounds*. Base stations are mounted on unmanned remote-controlled vehicles to be moved from one location to another and they can be located only at specific places called ‘feasible sites’. At the beginning of every round, the location of the base stations is determined using an integer linear programming model.

Wang *et al.* [15] investigate various combinations of networks with mobile sinks and/or mobile relays. The paper describes a performance study comparing different routing algorithms in three cases as follows: (i) the network consists of static nodes only; (ii) there exists a single mobile sink; and (iii) there exists a single mobile relay. A joint mobility and routing algorithm is described which requires the entire network to know the current location of the mobile node. The algorithm was then enhanced such that only a small portion of the nodes were needed to be aware of the location of the mobile node while still achieving the same performance as the previous algorithm.

A combination of base station mobility and multi-hop routing strategy are proposed by Luo and Hubaux [16] to maximize network lifetime. The paper shows that data collection protocols can be optimized, for instance for a better load balancing among the nodes in the network, by considering the mobility of the base station and multi-hop routing. The authors find that the most desirable mobility pattern for the base station is to follow the periphery of the network. The simulation results have demonstrated that highly loaded nodes reduced their load by a factor of five and the joint mobility and multi-hop strategy improved the network lifetime by 500%.

The MobiRoute architecture [17], is a sensor network with mobile sinks where the mobility is controlled and predictable and the sinks have long pauses in their movement called *epochs*. In a typical scenario, nodes send data via multi-hop communication toward the mobile sink, which changes its location based on route traces. A routing protocol forwarding data toward a sink must carry out the following processes: (i) inform the node when its communication link to the sink is broken due to mobility; (ii) alert the entire network of any topological variations; and (iii) reduce the packet loss during the time when the sink moves to a different position.

In Olariu *et al.* [18], the authors design, an AutoNomous netWorked sEnsoR system. The architecture assumes static sensor nodes and (possibly mobile) aggregation and forwarding nodes (AFNs). An important role of the AFNs is to organize the sensors in their immediate vicinity into a dynamic virtual infrastructure that depends on the current task. The AFN can

perform controlled mobility that balances the benefits of getting closer to the nodes recording a certain action with the risks of getting too close to potentially dangerous environments or agents.

### 3.4. Transmission scheduling

The process of determining when to transmit the buffered data.

Song and Hatzinakos [19] proposed a wireless MSSN that utilizes a single-hop transmission to avoid expensive multi-hop transmissions in sparsely deployed sensor networks. The proposed transmission scheduling algorithm (TSA-MSSN) uses the  $\lambda$  parameter to maintain the balance between maximizing the probability of successful data retrieval while keeping the energy consumption at a minimum. TSA-MSSN is a centralized algorithm and executed only at the sink in order to make use of the available resources.

Harras and Almeroth [20] makes use of a dedicated set of messengers to deliver message bundles between different routes (also defined as regions that are not connected). The regions and the nodes in the regions may be stationary or mobile. The authors have developed the following two messenger ownership schemes: regional and independent. In addition, the three message scheduling schemes introduced are periodic, storage-based and on-demand.

## 4. THE ORACLE OPTIMAL ALGORITHM FOR COMPLETE KNOWLEDGE TRANSMISSION SCHEDULING

Let us consider the transmission scheduling problem in a complete knowledge environment. As one of our assumptions we have stated that the transmission always occurs between the sensor node and the closest sink. Thus, we can characterize the mobility pattern of the mobile sinks from the point of view of a node through the vector  $D = (d_{t_{\text{start}}} \dots d_{t_{\text{stop}}})$ , where  $d_t$  represents the distance of the closest sink at time  $t$ .

A *transmission schedule* is a set of  $k$  time points, such that  $A = \{t_{\text{start}} < a_1, a_2, \dots, a_k = t_{\text{stop}}\}$ ,  $a_i < a_{i+1}$  and  $d_{a_i} \leq d_{t_r} \forall i$ , where  $d_{t_r}$  is the transmission range of the sensor node. The problem is to select the transmission schedule such that it minimizes the CPP.

As the number of distinct schedules is  $2^{(\text{stop}-\text{start})}$ , a naive solution-state search would have an exponential complexity.

In the following, we describe a graph-theoretic solution for the transmission scheduling problem. Exploiting the additiveness of the CPP, we construct a graph where the edge weights correspond to the components of the CPP sum. We create a directed acyclic graph  $G = (V, E)$  as follows. The set of vertices  $v_{\text{start}}, v_{\text{end}}$  correspond to the timepoints in the interval  $t_{\text{start}} \dots t_{\text{stop}}$ , vertices for which  $d_t > d_{t_r}$  have no incoming or outgoing edges (except the last vertex). For every edge  $e_{ij}$  with

$i < j$ , we associate a weight as follows:

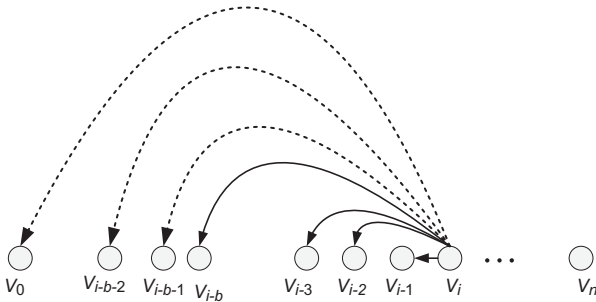
$$w_{ij} = \begin{cases} (j - i) \cdot r \cdot p_{\text{tx}}(d_j) & \text{if } i + t_{\text{bf}} > j, \\ t_{\text{bf}} \cdot r \cdot p_{\text{tx}}(d_j) + (j - i - t_{\text{bf}})p_w & \text{otherwise.} \end{cases} \quad (10)$$

If the last vertex  $v_{\text{stop}}$  has the associated distance  $d_{\text{stop}} > d_{\text{tr}}$ , it will exceptionally have incoming edges with the weight  $w_{i,\text{stop}} = p_w(\text{stop} - i)$ . This corresponds to the situation when the considered time interval finishes at a moment when no mobile sink is in the transmission range, and thus the data since the last transmission would be lost. The construction of this graph is illustrated on Fig. 1.

With this construction, the weight of an edge  $w_{ij}$  is the CPP cost incurred by the node if it made a transmission at time  $j$  such that the previous transmission was at  $i$ . The graph is acyclic (as all the edges follow increasing indexes). The graph is not fully connected, as vertices without any mobile sink in the transmission range have neither outgoing nor incoming links (exception the last vertex which has incoming edges even if it corresponds to a situation where no sink is in the transmission range). We can simply remove these unconnected vertices, obtaining a fully connected graph.

The solution of the transmission scheduling problem is the shortest path from vertex  $v_{\text{start}}$  to  $v_{\text{stop}}$ . The vertices in the shortest path are the timepoints where transmission is made; the cost of the shortest path is the total CPP of the solution.

The shortest path problem can be solved with Dijkstra's algorithm. The generated graph is *not* sparse, and thus the complexity of the algorithm is  $O(|E|) = O(n^2)$ , where  $n$  is the number of timepoints where there is a mobile sink in the transmission range. Although this is relatively low complexity, the practical deployment of the algorithm is dependent on the advance knowledge of the schedule of future visits by the mobile sinks.



Q4 **FIGURE 1.** The construction of the solution graph for the transmission scheduling problem. Due to the density of the full graph, only the connections of a single vertex  $v_i$  are represented. The edges represented by continuous lines have a weight of the form  $(j - i) \cdot r \cdot p_{\text{tx}}(d_j)$  while the edges represented by dotted lines have weights of the form  $t_{\text{bf}} \cdot r \cdot p_{\text{tx}}(d_j) + p_w \cdot (j - i - t_{\text{bf}})$ .

## 5. THREE HEURISTICS

In the following, we propose three heuristic algorithms for making the transmission decision. In contrast to the OrOpt algorithm, these algorithms make their decision based on very simple calculations and do not assume advance knowledge of the movement patterns of the mobile sinks. In the remainder of this section we shall use the following notation:  $M$  is the current buffer content,  $M_{\text{full}}$  is the size of the buffer,  $r$  is the data collection rate,  $d_{\text{tr}}$  is the transmission range, and  $d$  is the current distance of the closest mobile sink.

### 5.1. H1: Human-inspired simple heuristics

The first heuristic was designed based on the observation of several humans who were asked to play the transmission scheduling problem as a game, and then asked to describe their strategy. We found that humans are not comfortable doing calculations during the game. The strategies deployed by the human subjects were based on simple triggers based on the levels of the buffer and the current distance of the mobile sink. The subjects never described their approach as being stochastic, although in the practical game they did not adhere rigidly to the stated strategy. When asked directly whether they would base their strategy on a ‘coin toss’, all the interviewed people said that that does not appear to be a good strategy.

The heuristics is based on three parameters as follows:

- $d_{\text{opt}}$  optimal distance. This is the distance that, for the human, represents the intuition that the mobile sink is ‘as close as it gets’ to the node. Note that this might not be the absolute minimum of the distances, only a value that is hit with a relative certainty during the maximum collection interval.
- $M_{\text{L}}$  buffer too low to transmit. Represents the level at which the amount of collected data is too low to justify its transmission.
- $M_{\text{H}}$  buffer emergency level. This represents an amount of collected data that puts the system in danger of losing data. At values higher than this, the system will transmit at the next available opportunity, regardless of the distance.

The transmission rule of the system is shown in Algorithm 1. Obviously, the approach needs to be calibrated by choosing values for the  $d_{\text{opt}}$ ,  $M_{\text{L}}$  and  $M_{\text{H}}$  parameters. These values depend on many factors. For instance, if the transmission overhead is 0, that  $M_{\text{L}} = 0$ . If in the system there is always a mobile sink in the transmission range of the sensor, then  $M_{\text{H}} = M_{\text{full}}$  (the buffer capacity).

For cases when there is a chance that there is no mobile sink in the range of the sensor, we need to fall back to our intuition in determining the buffer emergency level. By interviewing our experimental subjects, we found that humans start to become alarmed about a possible data loss when the buffer reaches 90% capacity. Thus we chose the value  $M_{\text{H}} = 0.9 \cdot M_{\text{full}}$ .

---

**Algorithm 1.** H1: Human inspired simple heuristics.

---

**Require:**  $d, M, M_L, M_H, d_{opt}$   
**if**  $M > M_L$  **then**  
  **if**  $d < d_{opt}$  **then**  
    **return** transmit  
  **else if**  $M < M_H$  **then**  
    **return** transmit  
  **else**  
    **return** wait  
  **end if**  
**else**  
  **return** wait  
**end if**

---

## 5.2. H2: Stochastic transmission

The stochastic transmission heuristics transmits randomly with a probability distribution that is affected by the buffer level and the distance to the mobile sink. If the buffer is empty, this contributes a 0 value, while if the buffer is full, it contributes a 1 value to the probability; which is given by

$$p_b = \frac{M}{M_{full}}. \quad (11)$$

The second factor is the distance of the mobile sink. We assume that there is a minimum distance  $d_{min}$ , such that any distance smaller than that cannot be translated into an energy advantage. If the distance of the mobile sink is smaller than  $d_{min}$ , it contributes 1 to the probability of the transmission, while if it is  $d_{tr}$  or larger, it contributes 0. One expression that achieves these values is as follows:

$$p_d = \frac{d_{tr} - \max(d, d_{min})}{d_{tr} - d_{min}}. \quad (12)$$

We express the probability of the transmission as a weighted mixture of these two probabilities as follows:

$$p = w \cdot \frac{M}{M_{full}} + (1 - w) \cdot \frac{d_{tr} - \max(d, d_{min})}{d_{tr} - d_{min}}. \quad (13)$$

The overall algorithm for the stochastic transmission method is described in Algorithm 2.

---

**Algorithm 2.** H2: Stochastic transmission.

---

**Require:**  $d, M, d_{min}, w$   
 $p \leftarrow w \cdot \frac{M}{M_{full}} + (1 - w) \cdot \frac{d_{tr} - \max(d, d_{min})}{d_{tr} - d_{min}}$   
 $u \leftarrow U(0, 1)$  // generate random number  
**if**  $u \leq p$  **then**  
  **return** transmit  
**else**  
  **return** wait  
**end if**

---

## 5.3. H3: Constant risk

The reason for a node not to transmit is the hope that a better opportunity would appear in the future. Naturally, this decision carries a certain risk. The constant risk algorithm tries to estimate based on historical information how much risk a decision carries and then take decisions based on a constant risk factor. The goal is to prevent the algorithm from being too bold on one occasion and too cautious on others.

To implement this algorithm, the heuristics maintains a two-dimensional *risk estimation table*  $OP$ , where  $OP[t][d]$  is the fraction of the occasions in the history of the node when in a time window of size  $t$  the mobile sink came closer than distance  $d$ . We interpret this fraction as a future probability. The size of the array depends on the accuracy at which we quantize the distance and the buffer. Note that the measurements necessary to maintain this table do not depend on the behavior of the node.

The algorithm followed by the node is as follows. We choose a constant risk factor  $p_{risk}$  which we are willing to accept. Whenever we need to make a decision to send or not, we calculate the quantization of the current remaining time  $t_q$  and the current distance to the sink  $d_q$ , and look up the probability that a better opportunity will appear before we run out of buffer space  $OP[t_q][d_q]$ . If this probability is lower than  $p_{risk}$ , the node will send, otherwise it will wait.

Normally,  $p_{risk}$  should be chosen a value very close to 1 (we used 0.99 in our experiments). Note that even an  $OP$  value of 1 does not mean that there is no risk involved, but only that in the history of the node, those types of situations finished ‘well’, which might not be true in the future. The decision process is described in Algorithm 3.

---

**Algorithm 3.** H3: Constant risk.

---

**Require:**  $d, M, p_{risk}$   
 $t_q \leftarrow$  quantization of remaining time  
 $d_q \leftarrow$  quantization of distance to the sink  
**if**  $OP[t_q][d_q] < p_{risk}$  **then**  
  **return** transmit  
**else**  
  **return** wait  
**end if**

---

## 6. EXPERIMENTAL STUDY

### 6.1. Experimental setup and scenarios

We performed a series of experiments with a transmission scheduling scenario involving a field in which a number of mobile nodes are collecting the data from the sensor nodes using a one-hop transmission. The mobility pattern of the mobile sinks was random waypoint. We have assumed that the speed of the mobile sink was 1 m/s or 3.6 km/h. This is a realistic speed for a vehicle moving on rough terrain. We considered an area of  $400 \times 200$  m, with 4–20 mobile sinks. The transmission range

of the node was considered to be between 10–80 m, a realistic range for a sensor node. Finally, we assumed a 32 kB buffer and a data rate of 0.2 kB/s. The parameters of the simulation environment are summarized in Table 1.

Let us now consider the evolution of the mobile sinks in time. Even if the number of active mobile nodes is fixed, the

**TABLE 1.** The parameters of the simulation experiments.

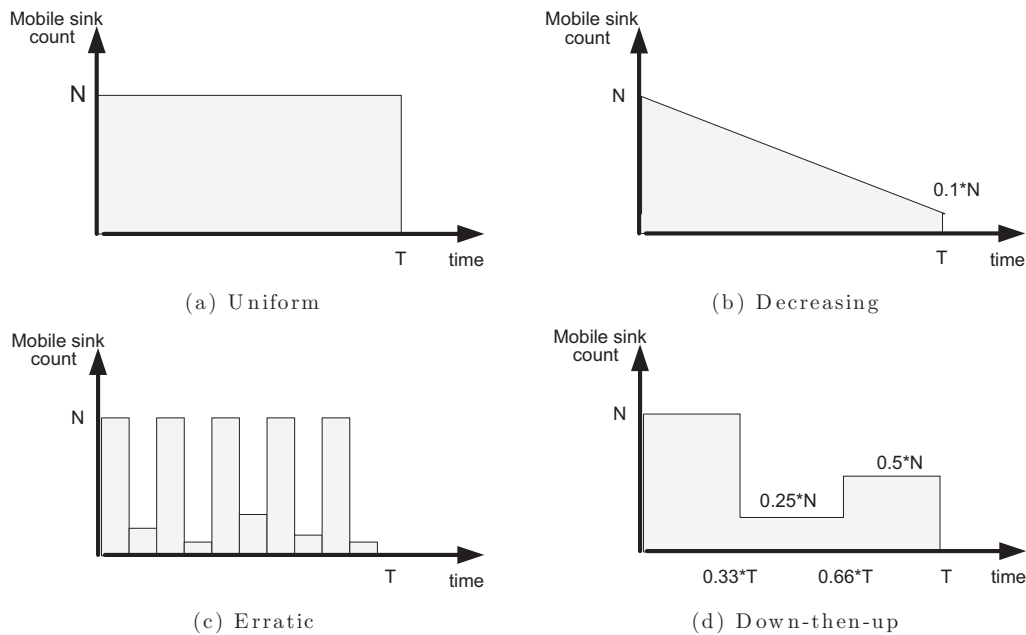
General settings	
Movement area	$400 \times 200$ m
Simulation time	10000 s
Mobile sinks	
Number	4–20 (10 default)
Velocity	1 m/s
Transmission range	80 m
Sensor nodes	
Buffer size	32 kB
Data rate	0.2 kB/s
Transmission range	10–80 m (50 default)
Transmission power model	
Path loss index $n$	4
$\alpha_{11}$	45 nJ/bit
$\alpha_2$	0.001 pJ/bit/m <sup>4</sup>
Policy	
Cost of data loss	equivalent to cost of transmitting at maximum power

randomness of their movement patterns implies a randomness in the distance of the closest mobile sink for a given sensor node. In addition, in a practical application, it is unlikely that the number of sinks would stay constant over a longer span of time. Even if the mobile sinks are dedicated data collection devices, they might get broken or reassigned to other tasks. In cases where the mobile sinks are piggybacked on mobile entities such as vehicles or humans, their number might vary drastically due to circumstances or tasks unrelated to the data collection task of the sensor network.

One of the questions we need to ask is whether different transmission scheduling algorithms behave differently under different patterns of variation of the number of mobile sinks. In particular, we should avoid relying on algorithms that show high performance under optimal conditions, but show significant drops in performance as a response to changing conditions. To investigate this, we designed four scenarios of the evolution of the number of mobile sinks. Each scenario starts with a *nominal number* of sinks  $N$ , which then evolves in time as follows:

*Uniform:* the number of sinks will remain at the nominal level throughout the scenario (Fig. 2a). Note that this does not imply a regularity of the visits, as the movement of the individual sinks is random waypoint. This corresponds to a well-maintained system in which the mobile nodes are replaced or repaired as necessary.

*Decreasing:* this scenario starts with the nominal number of nodes and decreases the number of nodes at a uniform pace, ending the scenario with 10% of the initial number (Fig. 2b). This corresponds to a system that is not maintained



**FIGURE 2.** Scenarios for the evolution of the number of mobile sinks over time.



after deployment, which is the case for systems deployed in unaccessible or hostile locations.

*Erratic*: the number of mobile nodes varies erratically, with sudden increases and drops over 10 time intervals (Fig. 2c). This corresponds to cases where the nodes are piggybacked over humans or vehicles whose presence is affected by factors unrelated to the data collection task.

*Down-then-up*: the number of mobile sinks drops suddenly to 25% in the second third of the scenario, but it is restored to half its nominal value in the last third (Fig. 2d). This pattern is characteristic for situations where the shifting interest of the beneficiaries requires the temporary reallocation of resources to other tasks.

Note that these scenarios are defined in terms of percentages over nominal values of mobile nodes and scenario length, and are adopted accordingly for varying number of initial nodes.

The environment and the proposed scheduling algorithms have been implemented in the YAES simulator framework [21].

## 6.2. Compared algorithms

In our experiments we compare four different sensor implementations:

*Oracle Optimal (OrOpt)*: This implementation has advance knowledge of the movements of the mobile sinks and calculates an optimal schedule which minimizes the given CPP. The implementation follows the description in Section 4.

As expected, the OrOpt algorithm always outperforms the other approaches, and as such, serves as a baseline to the level of performance that is possible for a given scenario. Note the fact that the algorithm is optimal does not mean that, it cannot lose data, as in certain scenarios the transmission of all the data is not possible.

*Human inspired (H1:HI)*: This is an implementation for the algorithm described in Section 5.1 and the pseudo-code in Algorithm 1. The free parameters were chosen as:  $M_L = 0.2$ ,  $M_H = 0.9 \cdot M_{full}$  and  $d = 0.5 \cdot d_{max}$ .

*Stochastic (H2:STO)*: This is an implementation for the algorithm described in Section 5.2 and the pseudo-code in Algorithm 2. The relative weight factor is chosen as  $w = 0.5$ .

*Constant risk (H3:CR)*: This is an implementation for the algorithm described in Section 5.3 and the pseudo-code in Algorithm 3. The constant risk factor  $p_{risk}$  was set to 0.99.

For each sensor model, we run the simulation with the same scenario and the same location of the sensor. The experiment was repeated 10 times, varying only the seeds of the random movement of the mobile sinks. We performed measurements of the total transmission energy and the data loss ratio, and retained the averages calculated over the 10 experiments.

## 6.3. Results: data loss

The first set of measurements concern the data lost. There are three components of the data loss: first, there is unavoidable data

loss due to the circumstances of the experiment. If no mobile sink comes into the transmission range for a duration longer than it takes to fill in the buffer, data will be unavoidably lost. The second source of data loss is the policy that strikes a balance between the cost of data lost and the cost of energy—even the optimal algorithm will choose to lose data if its cost is larger than the cost of transmitting. Finally, the third component is data loss due to imperfect decisions made by the node. This component does not exist for the OrOpt algorithm, and is responsible for the differences between the different algorithms.

Figure 3 shows the data loss function of the transmission range for the four considered scenarios. As expected, the data loss decreases with the transmission range, because a larger transmission range allows the mobile sinks to come into range more often for the same movement pattern. The OrOpt strategy, as expected, yields the best performance, followed by the H2:STO and H3:CR (almost indistinguishable) and H1:HI, at some distance. The differences between the various scenarios are significant in magnitude (the decreasing, erratic and down-then-up scenarios are harder than the uniform one), but very similar in pattern and the relative performance of the algorithms.

Figure 4 shows the data loss function of the nominal number of sinks in the various scenarios. Again, the overall shape is as expected, with the data loss decreasing with the increase in the number of sinks in each scenario, and the order of the performance remaining OrOpt as the best, followed by H2:STO and H3:CR (almost indistinguishable), and H1:HI at some distance. Overall, however, the differences in data loss are not significant in relative terms between the algorithms.

## 6.4. Results: energy consumption

The results for the data loss values give the impression that the transmission scheduling is an easy problem: the different algorithms appear to be very close to each other, close to the optimal algorithm and the different scenarios does not significantly affect the performance rankings. The results for the energy consumption, however, present a very different picture.

Let us first discuss our expectations about these values. A larger transmission range might allow the node to transmit to a farther sink, with a higher energy consumption. This can increase the energy consumption even in an optimal algorithm in situations when previously the choice was to lose packets. Once a full transmission is achieved, however, the energy consumption in an optimal algorithm would not depend on the transmission range. In a suboptimal algorithm, however, energy may be lost for unnecessary transmissions over longer distances.

Figure 5 shows the energy consumption function of transmission range. We find that the energy consumption increases with the transmission range for all the protocols, but the protocol shows a large divergence in value. The optimal protocol uses less than half the energy needed by the other protocols.

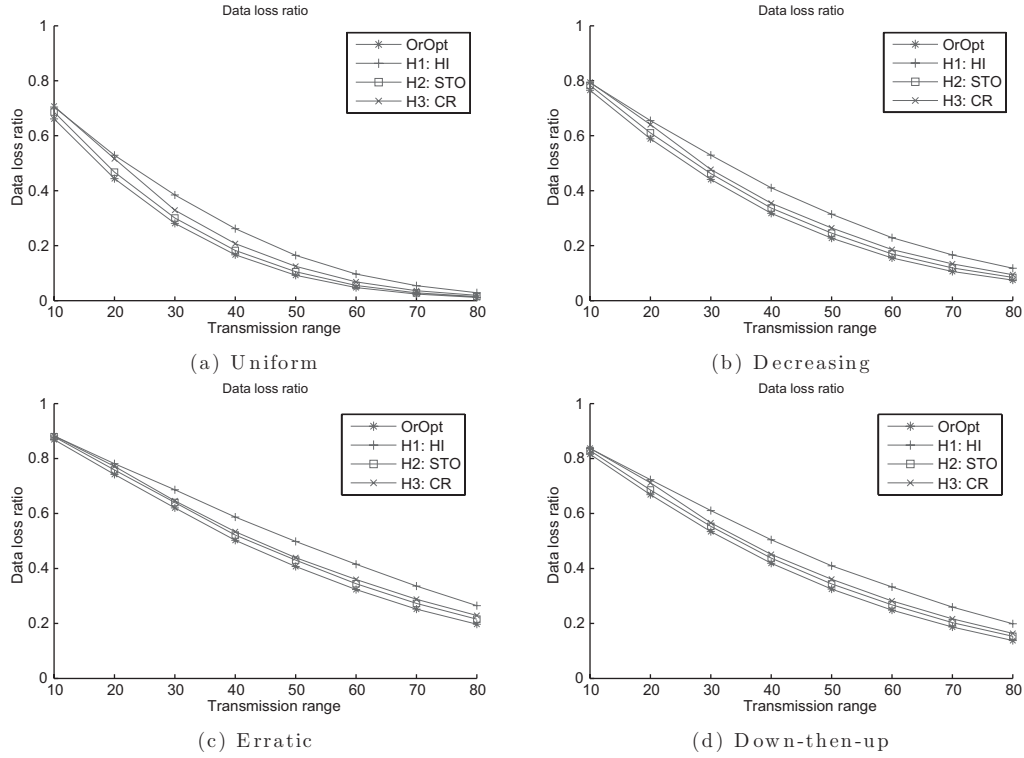


FIGURE 3. Data loss function of transmission range.

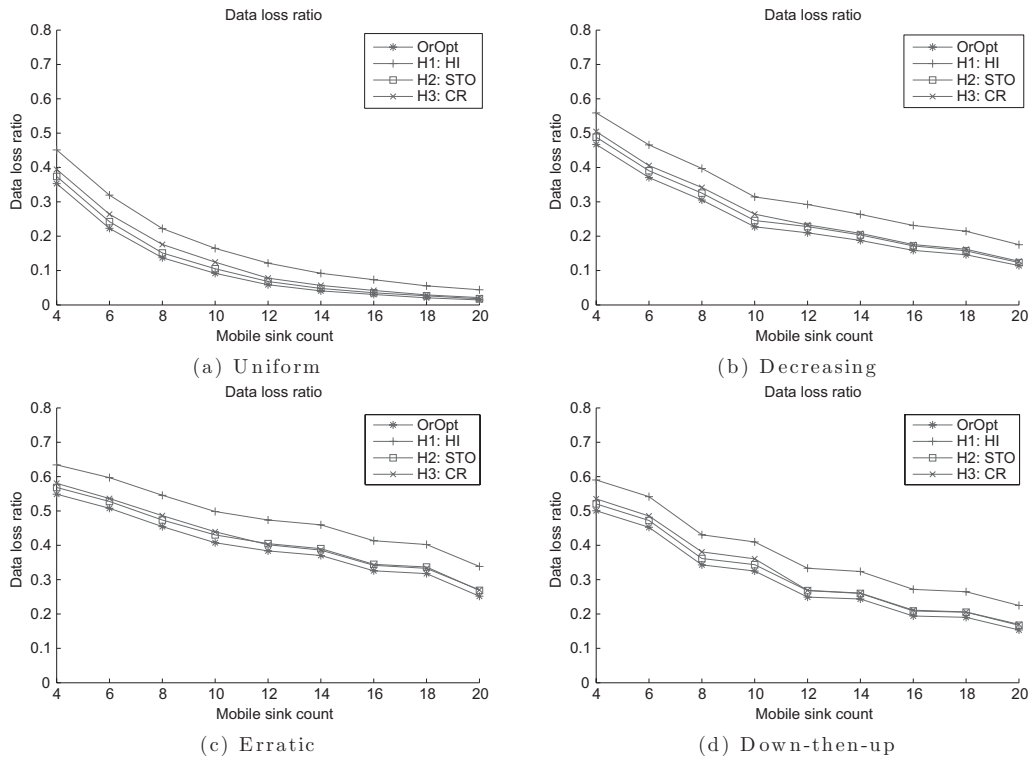


FIGURE 4. Data loss function of number of mobile sinks.

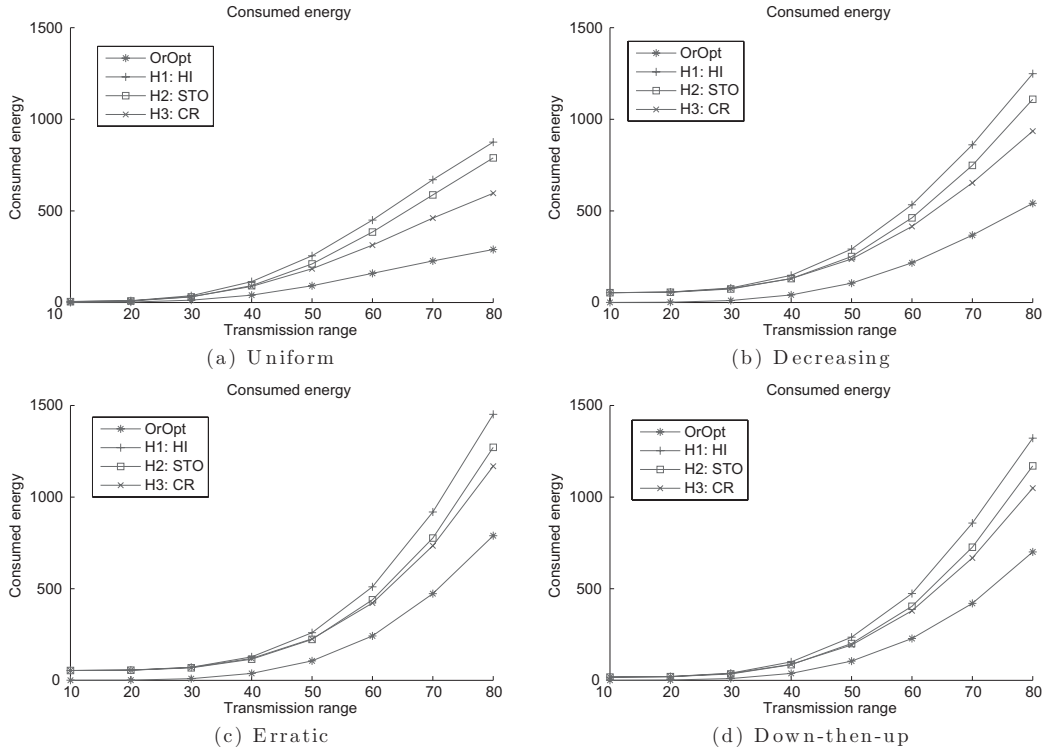


FIGURE 5. Energy consumption function of transmission range.

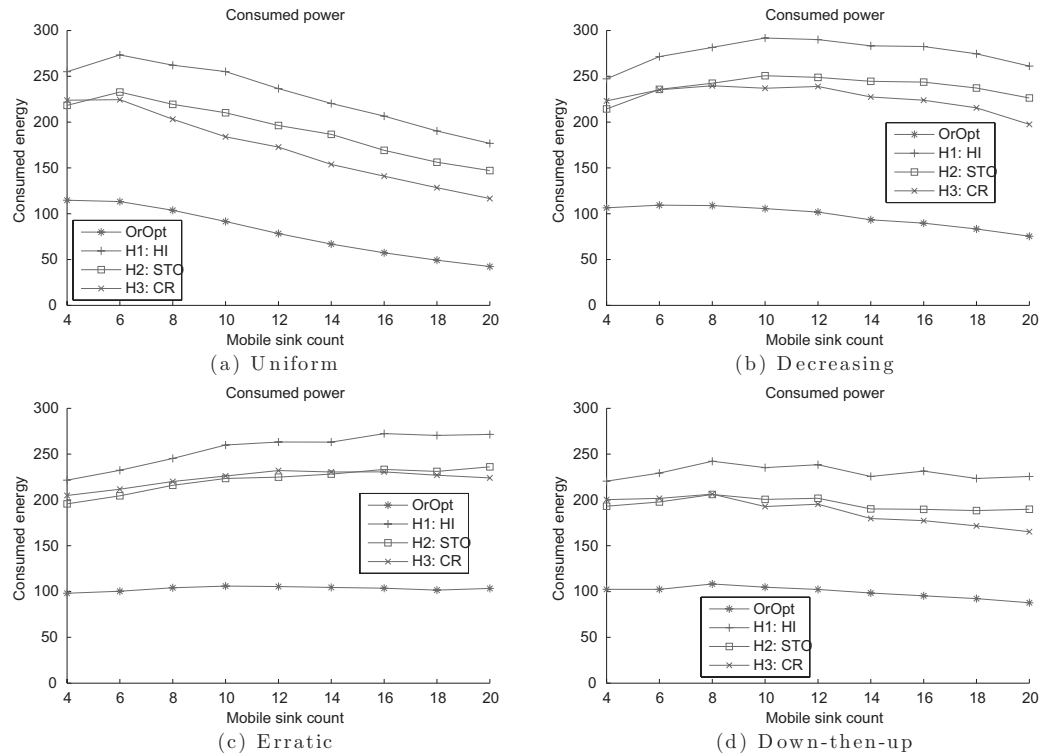


FIGURE 6. Energy consumption function of number of mobile sinks.

The next best is the constant risk algorithm H3:CR which has a roughly 30% advantage over the other two for the uniform scenario. However, we find that this advantage is much smaller for the decreasing scenario, and virtually inexistent for erratic and down-then-up. The reason for this phenomena is that the constant risk algorithm uses a risk table computed based on historical data in a uniform setting. Thus, the protocol underestimates the risk in an erratic or down-then-up scenario, yielding a comparatively lower performance.

Figure 6 shows the energy consumption function of the number of sinks. There is no clear trend in these graphs. Increasing the number of sinks might affect the energy consumption in the following two different ways: it can increase the energy consumption by letting the node transmit data that otherwise would be lost and it can decrease the consumption by providing more opportunities to transmit from a closer distance. In practice we find that in the uniform scenario the energy consumption decreases after the number of mobile sinks exceeds six. For the other scenarios, the more difficult is the scenario, the more the trend shifts toward the increase in the energy consumption with the number of sinks. As a note, if we would run these scenarios with a very large number of mobile sinks, we would eventually reach a point where these values would start to decrease as well.

Regarding the relative performance of the algorithms, we find that the OrOpt has a commanding lead, in average consuming 50% less power than the other algorithms. The next best is the constant risk algorithm H3:CR. Its advantage over H2:STO is especially significant in the ‘easy’ cases: in the uniform scenario as well as the decreasing and down-then-up scenario with a large number of mobile sinks. Its advantage is minimal for the ‘hard’ cases, such as the erratic scenario and the sections of the other scenarios with a low number of nodes. As discussed above, this is due to the fact that in these cases the H3:CR algorithm underestimates the risk, as its risk table was obtained in the uniform setting. The H1:HI algorithm is always the worst.

## 7. CONCLUSIONS

In this paper, we investigated the problem of transmission scheduling in sensor networks where nodes transmit their collected data with one-hop transmission to mobile sinks. Modeling the node as an autonomous agent, each node tries to maximize its utility by minimizing energy consumption and data loss. We described a graph-theory-based optimal algorithm that requires advance knowledge of the mobility patterns of the mobile sinks. Then, we introduced three heuristics that rely only on local and historical information.

Overall, the three heuristics managed to closely match the data loss performance of the optimal algorithm, but at the cost of a significantly higher power consumption. Consistently, the human-inspired algorithm H1:HI performed worse than the others. The constant risk-based algorithm H3:CR used

a risk table acquired from historical data. Using this, it usually outperformed the stochastic H2:STO algorithm in scenarios that closely matched the conditions in which the risk table was acquired. In scenarios where the number of mobile sinks were changing in time, the H3:CR and H2:STO algorithms performed essentially identically. The H2:STO algorithm appears to be a very good choice for implementation on devices with limited computing capabilities, due to its extreme simplicity, low memory requirements and the fact that it does not need any learning or data acquisition process.

Let us now outline some directions for future work. In this paper we took the approach of isolating the transmission scheduling problem from the sensing behavior, as well as the movement of the sinks—which we assumed to be random waypoint. We also assumed that the sensor node does not make any judgment on the value of the collected information. In a practical deployment, there are many possibilities of making more informed decisions. For instance, in the real world, truly random movements are rare. If the mobile sinks are dedicated data collection devices, their movement will be aligned with the needs of the data collection task. If the mobile sink is piggybacked on top of a vehicle or a human (such as a forest ranger) its movement, although not strictly regular, would still exhibit a certain structure that can be exploited. In addition, the sensor node might need to make other decisions beyond the simple send or wait considered in this paper. There is an inevitable evolution of sensor nodes from devices that simply collect and transmit data, to nodes which are aware of their environment and make autonomous decisions to further a higher-order goal.

## REFERENCES

- [1] Anastasi, G., Conti, M., Francesco, M.D. and Passarella, A. (2009) Energy conservation in wireless sensor networks: a survey. *Ad Hoc Netw.*, **7**, 537–568.
- [2] Rappaport, T. (1996) *Wireless Communications: Principles & Practice*. Prentice-Hall.
- [3] Shah, R., Roy, S., Jain, S. and Brunette, W. (2003) Data Mules: Modeling a Three-Tier Architecture for Sparse Sensor Networks. *Proc. 1st IEEE Int. Workshop on Sensor Network Protocols and Applications (SNPA-03)*, Anchorage, Alaska, May, pp. 30–41.
- [4] Anastasi, G., Conti, M., Monaldi, E. and Passarella, A. (2007) An Adaptive Data-Transfer Protocol for Sensor Networks with Data Mules. *IEEE Int. Symp. World of Wireless, Mobile, and Multimedia Networks (WoWMoM'07)*, Helsinki, Finland, June.
- [5] Kim, H.S., Abdelzaher, T. and Kwon, W. (2003) Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks. *Proc. 1st Int. Conf. Embedded Networked Sensor Systems (SenSys-03)*, Los Angeles, CA, USA, November, pp. 193–204.
- [6] Baruah, P., Urgaonkar, R. and Krishnamachari, B. (2004) Learning-Enforced Time Domain Routing to Mobile Sinks in Wireless Sensor Fields. *Proc. 29th Annual IEEE Int. Conf. Local*

- Computer Networks (LCN-04)*, Tampa, FL, USA, November, pp. 525–532.
- [7] Basagni, S., Melachrinoudis, A.C.E., Petrioli, C. and Wang, Z. (2008) Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wirel. Netw.*, **14**, 831–858.
- [8] Chakrabarti, A., Sabharwal, A. and Aazhang, B. (2003) Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks. *Proc. 2nd Int. Workshop on Information Processing in Sensor Networks (IPSN-03)*, Palo Alto, CA, USA, April, pp. 129–145.
- [9] Vincze, Z., Vass, D., Vida, R., Vidacs, A. and Telcs, A. (2007) Adaptive sink mobility in event-driven densely deployed wireless sensor networks. *Int. J. Ad Hoc Sens. Wirel. Netw.*, **3**, 255–284.
- [10] Mergen, G., Zhao, Q. and Tong, L. (2006) Sensor networks with mobile access: energy and capacity considerations. *IEEE Trans. Commun.*, **54**, 2033–2044.
- [11] Luo, H., Ye, F., Cheng, J., Lu, S. and Zhang, L. (2005) TTDD: Two-tier data dissemination in large-scale wireless sensor networks. *Wirel. Netw.*, **11**, 161–175.
- [12] Chen, C., Ma, J. and Yu, K. (2006) Designing Energy-Efficient Wireless Sensor Networks with Mobile Sinks. *Proc. Workshop on World-Sensor-Web (WSW-06) at the 4th ACM Conf. Embedded Networked Sensor Systems (SenSys-06)*, Boulder, CO, USA, October–November.
- [13] Kansal, A., Rahimi, M., Estrin, D., Kaiser, W., Pottie, G. and Srivastava, M. (2004) Controlled Mobility for Sustainable Wireless Sensor Networks. *Proc. 1st Annual IEEE Communications Society Conf. Sensor and Ad Hoc Communications and Networks (SECON-04)*, Santa Clara, CA, USA, October, pp. 1–6.
- [14] Gandham, S., Dawande, M., Prakash, R. and Venkatesan, S. (2003) Energy Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations. *Proc. IEEE GLOBECOM*, San Francisco, CA, USA, December, pp. 377–381.
- [15] Wang, W., Srinivasan, V. and Chua, K.-C. (2005) Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks. *Proc. MobiCom*, Cologne, Germany, September, pp. 270–283.
- [16] Luo, J. and Hubaux, J.-P. (2005) Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. *Proc. IEEE INFOCOM*, Miami, FL, USA, March, pp. 1735–1746.
- [17] Luo, J., Panchard, J., Piorowski, M., Grossglauser, M. and Hubaux, J.-P. (2006) MobiRoute: Routing towards a Mobile Sink for Improving Lifetime in Sensor Networks. *Proc. 2nd IEEE/ACM Int. Conf. Distributed Computing in Sensor Systems (DCOSS-06)*, San Francisco, CA, USA, June, pp. 480–497.
- [18] Olariu, S., Eltoweissy, M. and Younis, M. (2007) ANSWER: AutoNomouS netWorked sEnsoR system. *J. Parallel Distrib. Comput.*, **67**, 111–124.
- [19] Song, L. and Hatzinakos, D. (2007) Architecture of wireless sensor networks with mobile sinks: sparsely deployed sensors. *IEEE Trans. Veh. Technol.*, **56**, 1826–1836.
- [20] Harras, K. and Almeroth, K. (2006) Inter-regional Messengers Scheduling in Delay Tolerant Mobile Networks. *Proc. IEEE Int. Symp. World of Wireless, Mobile, and Multimedia Networks (WoWMoM'06)*, Niagara-Falls, Buffalo, NY, USA, June, pp. 93–102.
- [21] Bölöni, L. and Turgut, D. (2005) YAES—a Modular Simulator for Mobile Networks. *Proc. 8th ACM/IEEE Int. Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM-05)*, Montreal, Canada, October, pp. 169–173.