WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2008; **8**:385–403
Published online 17 December 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.584

Should I send now or send later? A decision-theoretic approach to transmission scheduling in sensor networks with mobile sinks

Ladislau Bölöni and Damla Turgut*,†
School of Electrical Engineering and Computer Science, University of Central Florida, FL, U.S.A.

Summary

Mobile sinks can significantly extend the lifetime of a sensor network by eliminating the need for expensive hop-by-hop routing. However, a sensor node might not always have a mobile sink in transmission range, or the mobile sink might be so far that the data transmission would be very expensive. In the latter case, the sensor node needs to make a decision whether it should send the data now, or take the risk to wait for a more favorable occasion. Making the right decisions in this *transmission scheduling problem* has significant impact on the performance and lifetime of the node. In this paper, we investigate the fundamentals of the transmission scheduling problem for sensor networks with mobile sinks. We first develop a dynamic programming-based optimal algorithm for the case when the mobility of the sinks is known in advance. Then, we describe two decision theoretic algorithms which use only probabilistic models learned from the history of interaction with the mobile sinks, and do not require knowledge about their future mobility patterns. The first algorithm uses Markov Decision Processes with states without history information, while the second algorithm encodes some elements of the history into the state. Through a series of experiments, we show that the decision theoretic approaches significantly outperform naive heuristics, and can have a performance close to that of the optimal approach, without requiring an advance knowledge of the mobility.

KEY WORDS: sensor networks; mobile sink; transmission scheduling

1. Introduction

Traditional sensor networks are composed of a set of low power sensor nodes which collect information and forward them through 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. In this architecture, the node needs to transmit both its own observations and forward the transmissions of the other nodes.

An alternative approach, more economical in terms of consumed power would be for the data to be collected by a set of mobile sinks, which are periodically visiting the vicinity of each node. The sensor nodes are collecting and buffering their observations,

*Correspondence to: Damla Turgut, School of Electrical Engineering and Computer Science, University of Central Florida, FL, U.S.A.
†E-mail: turgut@eecs.ucf.edu

and occasionally transmitting them to the closest actuator node. This approach leads to a better energy economy in the sensor nodes, because it eliminates the costly forwarding of packets done by nodes without renewable energy resources.

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 a large energy consumption. This creates a new problem for the sensor node: should it send the data now, or wait for a more favorable moment, when a sink will be closer, thus the data can be sent with a lower power consumption? Given that the necessary transmission power increases very quickly with the distance (in certain cases, for nodes close to the ground, it can be as much as the 4th 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.

In this paper, we study the transmission scheduling problem for sensor networks with mobile sinks. 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 is presented in Section 3. In Section 4, we present the Oracle Optimal 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. Section 5 describes a decision theoretic approach to the transmission scheduling problem. The approach is based on the building of probabilistic models of the environment and positions the problem as a Markov Decision Process (MDP). We propose two versions of the MDP encoding: one without explicit encoding of the history of mobile sink and another one with a simplified encoding of the history. In Section 6, we present the results of an experimental study. As expected, the Oracle Optimal algorithm outperforms the other algorithms. We show that the decision theoretic algorithms show a performance close to the optimal and significantly outperform simple heuristics. We conclude in Section 7.

2. The Transmission Scheduling Problem

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

Sensor networks with mobile sinks 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, thus the transmission scheduling problem needs to be solved by the node. 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 position for receiving data.

In the following, we describe our assumptions about the deployment scenario. Naturally, the algorithms proposed in this paper need to be appropriately modified if deployed under different assumptions.

- 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.
- The data transmission always happens between the sensor node and the closest mobile sink.
- The sink does not move during the transmission.
- The nodes have a finite buffer of constant size and collect observations with a constant bit rate.
- There is no deadline associated with the transmission of the observations. That is, the node can buffer information for an arbitrary amount of time without penalty.

Naturally, the relaxation of some of these assumptions leads to more complex problems.

DOI: 10.1002/wcm

One of the most important assumptions is that we consider all the transmissions to be from the source node to the mobile sinks. If we assume that the nodes can choose between a single-hop transmission to the sink or a multi-hop transmission, a series of new, complex choices appear. The node needs to decide whether to transmit to the sink, to the neighboring nodes (initiating a hop-by-hop transmission), or to wait. At the same time, the node needs to make decisions whether and when to transmit, buffer or drop incoming hop-by-hop messages. While the cost of the node-to-sink direct transmission is limited to the node, the energy cost of the hop-by-hop transmission is spread across the nodes of the path. While an interesting challenge, the detailed consideration of such systems are outside the scope of this paper. We will show that an important special case, when the nodes resort to hop-by-hop transmission only to avoid loosing data, can be handled with our model.

Let us now consider the objectives of the nodes. In the big picture, the nodes are striving to transmit all the observations (i.e., to minimize the number of observations lost) while simultaneously minimizing the energy consumption. The scheduling strategy will try to minimize an objective function which represents a balance of these two factors. Neither of these two objectives alone would yield the desired behavior. Considering only the energy minimization criterion would create a sensor which does not transmit any observation. Considering only the goal to minimize lost data would create a system which will transmit at every available opportunity.

Thus, a suitable objective function would consider both components, for instance, in the form of a weighted sum, which is calculated cummulatively over the considered time interval. We call this objective function the Cummulative Policy Penalty (CPP). The 'cummulative' aspect of the definition is important; for instance a sensor can make a bad decision (e.g., not to transmit at a favorable moment) without immediately occurring a penalty.

The transmission energy is fully determined by the physical factors. We use the following model for the energy dissipation used for communication [1]:

$$p_{tx} = (\alpha_{11} + \alpha_2 d^n)b \qquad (1)$$

where p_{tx} is the power dissipated when the node is transmitting to the mobile sink, d is the distance to the sink, n is the path loss index, and b is the number of bits transmitted. α_{11} and α_2 are positive constants. The path loss index varies between 2..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 will 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^4 (for $n = 4$).

In most cases, the data loss penalty component can be determined by the user based on the requirements of the application. A special case are systems which consider hop-by-hop transmission only in the last resort. These systems would transmit through a hop-by-hop model only the information which in other cases would be lost through buffer overflow. One way to model this by setting the buffer overflow penalty to the average cost of the hop-by-hop transmission.

3. Related Work

The traditional view of wireless sensor networks was based on the assumption of fixed sinks and multihop 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 methods to optimize 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 they concentrate on.

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 which do not have direct access to the sink are using hop-by-hop routing either toward the mobile sink or toward sensor nodes which are periodically visited by a sink.

The Mobile Ubiquitous LAN Extension (MULE) [2] architecture has 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

DOI: 10.1002/wcm

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. The theoretical results were verified with a custom discrete event simulator.

Scalable Energy-efficient Asynchronous Dissemination (SEAD) [3] is a distributed self-organizing protocol that reduces the energy consumption by the construction of a dissemination tree (*d-tree*) and dissemination of the data to the mobile sinks. SEAD extends the data placement heuristic [4], but differs from it by the use of mobile sinks and relaxing the high network density assumption. SEAD is evaluated using ns-2 and its performance is compared against the Directed Diffusion [5], Two-Tier Data Dissemination (TTDD) [6], and ADMR [7] protocols in terms of energy consumption per node and average end-to-end delay. The simulation results show that SEAD outperforms these approaches in terms of building and maintaining dissemination trees to mobile sinks.

Hybrid Learning-Enforced Time Domain Routing (HLETDR) [8] aims to deliver sensor data toward a mobile sink over multiple-hops. 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 toward the *moles*, sensor nodes located within the sink's path. The objective is to forward the data in a path toward a mole which is located within the proximity of the sink. Once the data arrives at the mole, the mole evaluates the 'goodness' value of the path based on the probabilistic local information of the current location of the sink and reinforces the route toward the sink. This reinforcement proliferates to the source and sets up gradients.

3.2. Mobility Models of the Sinks

The mobility of the sink can be categorized into 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 Reference [9]. In their experiments, 256 homogeneous sensor nodes are arranged in a square grid, with a single mobile sink moving in the area. 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 queueing system in Reference [10] 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.

Reference [11] examines how the various sink mobility patterns affect the network lifetime. The goal is to adaptively control the sink mobility to reduce energy consumption, in turn maximizing the lifetime of the network. The paper assumes an event-driven scenario with multi-hop communication between the sink and the sensors. The sink roams inside the network as a result of the current events which are based on the 'intruder movement' event model.

The SEnsor Networks with Mobile Agents (SENMA) [12] architecture was proposed for power constrained large scale dense sensor networks. SENMA consists of two types of nodes: (i) *sensor nodes* which are resource constrained, lightweight, and low cost, and (ii) resource-rich *mobile agents* (the equivalents of mobile sinks). SENMA relies on one hop transmission between the sensor nodes and mobile agents. For communication, the system uses a slotted time division duplexing (TDD) 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 TTDD [6] protocol 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 toward the sink.

DOI: 10.1002/wcm

The Mobile Enabled Wireless Sensor Networks (mWSN) architecture [13] uses multi-hop forwarding to form a cluster around the expected position of the mobile sink. mWSN is similar to Data MULEs in that it is a three-tier architecture, with the top tier composed of a base station, also called the final fusion point, the middle tier including mobile sinks such as mobile phones, laptops, and so on, while the bottom tier contains the static sensor nodes. 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. Theoretical results show that by learning the mobility pattern of the mobile sinks, the multi-hop clustering scheme can forward packets to the estimated positions of the sink in more timely and energy-efficient manner.

Kansal *et al*. [14] proposed the use of controlled and coordinated motion of network elements to alleviate resource limitations and improve system performance by adapting to the deployment demands. The authors developed an Autonomous Intelligent Mobile Micro-Server (AIMMS) prototype which travels across the network to route data from the deeply embedded nodes. Nodes have to relay data only for the nodes which do not fall into the transmission range of the micro-server.

In Reference [15], 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. The initial locations of the base stations are selected by the modified Minimum Cost Forwarding (MCF) routing protocol [16].

Reference [17] investigates various combinations of networks with mobile sinks and/or mobile relays. The paper describes a performance study comparing different routing algorithms in three cases when (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. The comparison of mobile relay and mobile sink revealed that for a sensor network with a radius of R hops, $O(R)$ mobile relays are required to be equivalent in performance with the mobile sink scenario.

A combination of base station mobility and multi-hop routing strategy are proposed in Reference [18] 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 per cent.

The MobiRoute architecture [19], an extension of MintRoute [20], 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 Reference [21], the authors design ANSWER, 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 which depends on the current task. The AFN can perform a controlled mobility which 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. The paper also proposes a specific communication infrastructure which puts emphasis on a dynamic coordinate system, based on coronas and wedges which also serves as a clustering architecture, dynamic partitioning of the graph through coloring and a security architecture.

DOI: 10.1002/wcm

3.4. Transmission Scheduling

This is the process of determining when to transmit the buffered data.

Song *et al.* [22] propose several algorithms for transmitting from sensor nodes to a sink which moves on a linear path. The optimal multiple nodes transmission scheduling algorithm (MTSA-MSSN) requires the sink to estimate its own current velocity and direction of the mobility from GPS. The estimated state, $\hat{E}(i)$, is modeled as a Markov chain in time domain. The paper also proposes two suboptimal algorithms MSPS-MSSN and MSUS-MSSN. A series of simulations are performed to study the tradeoff between the number of successfully transmitted packets and energy consumption. It was found that the two suboptimal algorithms show very close performance to the optimal MTSA-MSSN algorithm.

A distributed opportunistic information retrieval algorithm that uses channel state information (CSI) is proposed in Reference [23]. This protocol encodes channel state into the backoff strategy of the carrier sensing, which improves robustness against propagation delay. The information from the sensors is gathered by the mobile access point. The sensors are sending their data at the moment when they are activated by a beacon signal from the mobile access point. This opportunistic transmission strategy is extension to SENMA [12] and CSI-based carrier sensing with negligible propagation delay, the earlier works of the authors. To minimize the effects of performance degradation due to propagation delay, the backoff function is constructed using asymptotic extreme order statistics. The simulation results indicate that the CSI-based carrier sensing performance decays gracefully with the propagation delay. It is also shown that the performance of the opportunistic strategy depends on the number of activated sensors.

4. The Oracle Optimal Algorithm for Complete Knowledge Transmission Scheduling

In this section, we develop an algorithm which finds the optimal transmission schedule under the assumption that the mobility pattern of the sinks is known. The definition of optimality in this case is that the algorithm finds a schedule which minimizes the cummulative policy penalty for the specified interval. The objective of this algorithm is to serve as a baseline for the more realistic algorithms. We call this algorithm Oracle Optimal to indicate the fact that it needs advance knowledge of the future movement of the mobile sinks.

As one of our assumptions, we have stated that the transmission always happens 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 $D = (d_{t_{\text{start}}} \ldots d_{t_{\text{stop}}})$, 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 $A = \{t_{\text{start}} < a_1, a_2, \ldots a_k = t_{\text{stop}}\}$, $a_i < a_{i+1}$ and $d_{a_i} \leq d_{tr}$ $\forall i$ where d_{tr} is the transmission range of the sensor node.

We define the cummulative policy penalty as a function $CPP([t_1, t_2], A) \in \mathbb{R}$. CPP can have various expressions but it is additive over disjoint, consecutive time intervals:

$$\begin{aligned} &CPP([t_1, t_2], \{a_i | a_i \in [t_1, t_2] \wedge a_n = t_2\}) \\ &+ CPP([t_2, t_3], \{b_j | b_j \in (t_2, t_3]\}) \\ &\;= CPP([t_1, t_3], \{a_i\} \cup \{b_j\}\}) \end{aligned} \tag{2}$$

Let us now investigate the number of distinct possible schedules. Let us assume that we have n timepoints, out of which in $m \leq n$ points the distance is smaller that the transmission range. At any of these timepoints, the sensor has the choice to send or not to send, thus the number of valid schedules is 2^m. As m can be as high as n, the naive search for the best schedule is of exponential complexity. We will design a dynamic programming-based algorithm which, for the average case, can significantly reduce the number of choices which needs to be investigated.

Property 1. *If* $A = \{t_{\text{start}} < a_1, a_2, \ldots a_k = t_{\text{stop}}\}$ *is the optimal schedule for the time interval* $[t_{\text{start}}, t_{\text{stop}}]$ *than for all* a_i *the schedules* $A_1 = \{t_{\text{start}}, \ldots a_i\}$ *and* $A_2 = \{a_{i+1}, \ldots, a_k = t_{\text{stop}}\}$ *are optimal schedules for the intervals* $[t_{\text{start}}, a_i]$ *and* $[a_i, t_{\text{stop}}]$*, respectively.*

Proof. Let us assume that there is a different schedule A'_1 for which $P_{\text{total}}(A_1) > P_{\text{total}}(A'_1)$. Then the schedule A' obtained from the concatenation of A'_1 and A_2 will have a total power consumption $P_{\text{total}}(A') = P(A'_1) + P(A_2) < P(A_1) + P(A_2) = P(A)$, which means that A is not an optimal schedule, which is a contradiction. □

The pseudocode of the Oracle Optimal algorithm is presented in Algorithm 1. While still exponential in the worst case, the Oracle Optimal algorithm can significantly cut the computation time by pruning off

DOI: 10.1002/wcm

branches of computation which yield worse solutions than the ones already found. In addition, the algorithm uses an additional heuristic to sort the solutions starting from the most promising ones. The better the heuristics, the more significant pruning can be obtained. In addition to this, the algorithms uses a cache for the partial results. The exact performance analysis of the algorithm is outside the scope of this paper. In practice, the algorithm showed acceptable running times of less than a minute on a desktop computer for datasets with up to 1000 possible transmission timepoints. However, the computational complexity and the memory requirements (for the cache) clearly exceed the possibilities of a sensor node.

Algorithm 1. The Oracle Optimal algorithm

```
Function OracleOptimal(D = {d_t_start, d_t_end}, currentBest)
  If solution already exists in the cache
    Return the solution from the cache
  EndIf
  PTP = possible transmission points in D
  STP = PTP sorted by heuristics
  For all points a_i in STP
    (A_1, CPP_1) = OracleOptimal({d_t_start, a_i}, currentBest)
    If CPP_1 > currentBest
      Continue
    EndIf
    (A_2, CPP_2) = OracleOptimal({a_i, d_t_end}, currentBest)
    If CPP_1 + CPP_2 < currentBest
      A = A_1 ∪ A_2
      currentBest = CPP_1 + CPP_2
    EndIf
  EndFor
  add solution to cache
  Return (A, currentBest)
EndFunction
```

5. A Markov Decision Process-Based Approach for Transmission Scheduling

A MDP models decision making in situations where the outcome depends both on the actions of the agent as well as outside, stochastic factors. The operation of an MDP can be briefly described as follows. The agent can be in any of the states s_i, $i = 1, \ldots, n$. The actions the agent can take are a_j, $j = 1, \ldots, m$, although some of the actions might not be available in all states. If an agent in state s_i takes an action a it will transition to state s_j with probability $p_a(i, j)$. An important component of an MDP is the reward or punishment: this can be associated with a given state $R(s_i)$, with an action in a given state $R(s_i, a)$ or with a certain transition taken as a result of an action $R(s_i, a, s_j)$. The formulations are ultimately equivalent, for matters of convenience we had chosen the rewards to be associated with state-action pairs.

For agents with a finite time horizon, the goal of the agent is to maximize the sum of the rewards collected over its time horizon. For agents with an infinite time horizon, we define a discount factor $\gamma \in [0, 1]$, and let the agent maximize the *discounted reward* $\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t)$. The discount factor shows that the agent prefers γ times less a reward at time $t + 1$ compared with the preference for the same reward at moment t. γ is usually chosen as a value close to 1.

Solving an MDP is equivalent to finding a policy $P : S \to A$, that is, a rule which tells the agent that in state s_i it should take action $a = P(s_i)$. Following the policy will maximize the agents' discounted reward. There are several ways of solving MDPs, the most popular ones being *value iteration* which seeks to establish the value of each state and *policy iteration* which seeks to find the policy directly, without establishing the state values as well. Various combinations of these approaches are also frequently used.

The Markovian nature of the problem is reflected by the fact that the action of the agent depends only on the current state—it does not depend on the history. What this means in practice is that the MDP state needs to encode not only the state of the agent but the state of the environment as well, together with whatever historical information is deemed necessary. Finding an efficient representation of all the necessary information in the form of a finite (and preferably small) set of states is critical to the success of the MDP approach. The advantage is that once we determined this representation and have acquired the associated probabilities, the MDP approach will calculate the optimal decision policy (in the limit of the expressivity of the state representation and the accuracy of the transition probabilities).

The transmission scheduling problem can be conveniently and (with some representational effort) compactly represented in the terms of an MDP. The general outline of such a representation is as follows. The state of the sensor is determined by the level of the data buffer as well as the distance of the closest mobile sink. The actions of a sensor are whether to send (SEND) or not (DO-NOT-SEND). The reward associated with a state-action pair is the current component of the CPP, that is, a combination of the cost of sending the content of the buffer and the penalty associated with loosing data.

The transition probabilities between states reflect the probabilistic evolution of the distance to the closest

DOI: 10.1002/wcm

mobile sink. We assume that the data transfer always succeeds if the mobile sink is within transmission range, thus this component of the state is fully determined by the action of the agent and the previous state. If the data buffer content was c bytes at state s_i, after a SEND action it will be 0 bytes, while after a DO-NOT-SEND action it will be $c + r$ bytes where r is the data rate of the sensor. The MDP model can also elegantly handle various complicating assumptions. For instance, the fact that a transmission might not always succeed can be simply represented by adding two outcomes to the given action—one when it succeeds and one when it does not.

Starting from this general model, there are several alternatives for the building of the MDP model. We find that relatively subtle decisions can significantly affect the performance of the model. The main differentiator among these models is the representation of the state. The other parameters of the MDP, such as the transition probabilities are uniquely determined, once the representation is decided (which does not mean that they are necessarily easy to compute). In the following, we describe in detail the construction of two alternative models; in the first model, we build the most compact state representation possible, while in the second model we include information about the distance history at the cost of the increase of the state space.

5.1. A Historyless MDP Model of the Transmission Scheduling Problem

In this model, we encode the state as a tuple (b, d) where $b \in \{1, \ldots, m\}$ is a quantization of the buffer level, while $d \in \{1, \ldots, n\}$ is a quantization of the distance of the nodes.

Let us now discuss the details of the quantization process. The goal of the quantization of the distance is to reduce the continuous valued distance measurement to a (preferably small) number of discrete values. The quantization is determined by a quantization schedule $\mathrm{QD} = \{d_0, d_1, \ldots d_{n-1}, d_n\}$ with $d_0 = 0$ and $d_n = \infty$. We say that a distance D is in quantum i according to the schedule QD if $d_{i-1} \leq d < d_i$.

The choice of the quantization schedule can influence the quality of the decision making. The goal is to retain as much useful information as possible, while at the same time reducing the number of quantums. While developing a theory for the optimal quantization of distance is beyond the scope of this paper, we can apply our knowledge of the application domain to develop an appropriate quantization schedule.

First, the sensor knows about the distance of the mobile sink through the transmissions or beacon signals of the sink. One way this can happen is through the sink broadcasting its own position. As the mobile sink is usually a large device such as an unmanned ground or air vehicle, we can make the assumption that it has a GPS or other means of self-localization. Another approach is the node measuring the signal strength of the sink and using this information to infer distance. For both cases, the accuracy of distance measurement is limiting the number of quantums worth considering. For instance, if the distance can be measured with an error of $-50/+100$ per cent (which is reasonable for a measurement based on signal strength) then an appropriate choice of the quantums would be 10, 20, 40, 80, 160 m. If, on the other hand, the measurement is based on a mobile sink identifying its own position with an accuracy of 10 m, for instance, through GPS, then the right choice of the quantums would be 10, 20, 30, 40, ... , and so on.

In all cases, the last quantum should be not larger than the transmission range of the mobile sink $\mathrm{MS_{tr}}$. Note that if the sink has a high accuracy localization method, we can still end up with a large number of quantums.

The second consideration is the transmission range of the sensor node $\mathrm{S_{tr}}$, which is normally significantly lower than the one of the mobile sink. As the sensor cannot transmit beyond its transmission range, necessarily, the action in all states beyond that range will be DO-NOT-SEND. Therefore, there is no benefit in partitioning the distances between $\mathrm{S_{tr}}$ and $\mathrm{MS_{tr}}$ into multiple quantums, as they will always map to the same action.

What remains to be determined is the way in which the distances between 0 and $\mathrm{S_{tr}}$ are partitioned in quantums. The simplest choice is to divide the distance into k quantums of equal size. This would yield a quantization into $k + 2$ possible values with a schedule of:

$$\mathrm{QD} = \left\{0, \frac{1}{k}\mathrm{S_{tr}}, \ldots, \frac{i}{k}\mathrm{S_{tr}}, \ldots, \mathrm{S_{tr}}, \mathrm{MS_{tr}}\right\} \quad (3)$$

The problem remains whether the quantization retains the most useful information necessary for the decision process. Notice that the decision to send or not send depends on the energy necessary to send the data. If we assume that the node can use the minimum energy necessary to send the data then this energy can be calculated either from the laws of propagation [1] or experimentally for the specific type of environment

DOI: 10.1002/wcm

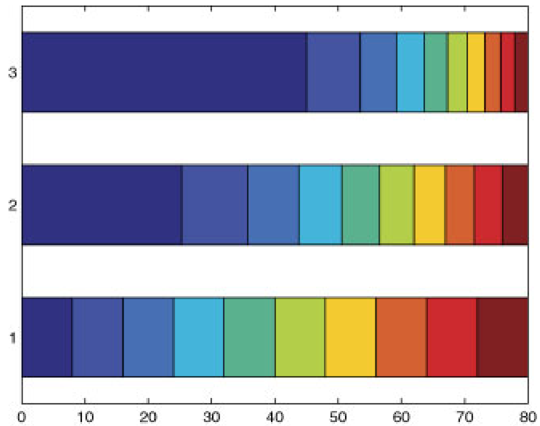

Fig. 1. Comparison of distance quantization schedules for the range of 0 to 80 m: (1) iso-distance schedule, (2) iso-power schedule for path loss index 2, (3) iso-power schedule for path loss index 4.

[24]. The most important component is the path loss factor, which can range between 2 and 4.

Intuitively, the distance quantization schedule retains the maximum information not when it partitions the distance equally, but when it partitions the corresponding transmission power equally, the latter being the basis of the transmission decision. If the transmission power has an expression of the type shown in Equation (1), with a path loss index n and we want to divide the schedule evenly for the consumed power, we need to choose a quantization schedule:

$$\mathrm{QD} = \left\{0, \sqrt[n]{\frac{1}{k}}\mathrm{S}_{\mathrm{tr}}, \ldots, \sqrt[n]{\frac{i}{k}}\mathrm{S}_{\mathrm{tr}}, \ldots, \mathrm{S}_{\mathrm{tr}}, \mathrm{MS}_{\mathrm{tr}}\right\} \quad (4)$$

We will call this an iso-power quantization schedule. Figure 1 compares the iso-distance schedule and two iso-power schedules for an example where $\mathrm{S}_{\mathrm{tr}} = 50$, $\mathrm{MS}_{\mathrm{tr}} = 75$ and the path loss index is 2 and 4, respectively.

Let us now consider the problem of quantization for the buffer level. The buffer is naturally discrete, so the goal in this case is not to discretize a continuous value, but rather to reduce the number of quantums. As a sensor can have several kilobytes of buffer space, having a separate quantum for each possible value would yield thousands of states (which then need to be multiplied with the number of distance quantums to obtain the states of the MDP).

For the buffer level quantums, again, we need to consider the nature of the function we plan to quantize. If our goal is to optimize the transmission power, we need to consider the full expression of the cost of the sending of the data. This might include the cost for the packet overhead. For instance, the overhead of an Ethernet packet is 26 bytes, to which additional overhead is added by the higher layers of the protocol stack (if they are present). In addition, if the transmission cannot be done in one packet, the transmission power will increase stepwise, as at every occasion when an additional packet is necessary a new overhead will be added.

Despite these complications, the transmission power will increase roughly linearly with the amount of data to be transmitted, so a uniform quantization of the buffer level will be appropriate. We assume that a step of the quantization represents an increase equivalent to the sensor data rate for one time step.

With these quantization decisions, we build an MDP with $m \times n$ states, where m is the number of buffer content quantums and n is the distance quantums. A state is represented as $\{b, d\}$. Based on domain knowledge, the MDP needs to satisfy the following requirements:

Requirement 1. $\forall d, \forall d', \forall b \in \{0 \ldots m-3\}, \forall b' > b+1$, $P_{DO\text{-}NOT\text{-}SEND}(\{b, d\}, \{b', d'\}) = 0$.

That is, the sensor cannot move directly into a state where the buffer content has increased with more than the sensor data rate.

Requirement 2. $\forall d, \forall d', \forall b, \forall b' \leq b$, $P_{DO\text{-}NOT\text{-}SEND}(\{b,d\},\{b',d'\}) = 0$.

That is, the buffer content cannot decrease if the sensor data is not transmitted.

Requirement 3. $\forall d, \forall d', \forall b, \forall b' > 0$, $P_{SEND}(\{b, d\}, \{b', d'\}) = 0$.

This requirement encodes our assumption that the transmission of the data is always successful.

Requirement 4. $\forall d, \forall d', \forall b < n-1$, $P_{SEND}(\{b, d\}, \{0, d'\}) = P_{DO\text{-}NOT\text{-}SEND}(\{b, d\}, \{b+1, d'\}) = P_{DO\text{-}NOT\text{-}SEND}(\{n-1, d\}, \{n-1, d'\})$.

This requirement expresses the independence of the distance of the mobile sink from the actions and the buffer content level.

An MDP which respects all the requirements above for the case with 2 distance and 3 buffer level quantums is shown in Figure 2. Due to the structure of the state encoding, it is convenient to visually arrange the MDP into an $n \times m$ rectangle, where the buffer content levels are arranged in the columns, while the distance quantums correspond to the rows. Thus, the state of

DOI: 10.1002/wcm

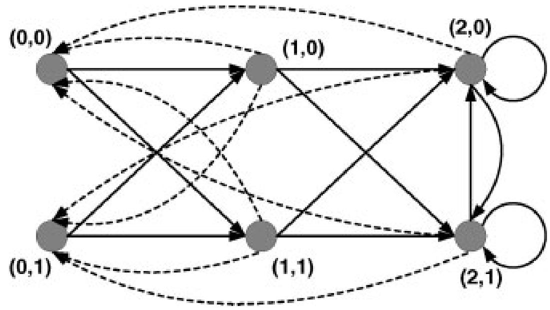

Fig. 2. An example Markov Decision Process, for the historyless encoding. The continuous line arrows indicate the DO-NOT-SEND actions, while the dotted line arrows indicate SEND actions.

the system will progress from left to right for the DO-NOT-SEND actions, and jump back to the first column for the SEND actions.

Finally, we need to decide on the rewards attached to various state-action pairs. Again, we can introduce some requirements based on domain knowledge.

Requirement 5. $\forall d, \forall b < n-1$
$R(\{b, d\}, DO\text{-}NOT\text{-}SEND) = 0$

That is, not sending in a state where the buffer is not full, carries no immediate reward or penalty.

Requirement 6. $\forall d$
$R(\{n-1, d\}, DO\text{-}NOT\text{-}SEND) = R_{Dataloss} < 0$

That is, if the sensor does not transmit when its buffer is full, it will loose an amount of data equal to its data acquisition rate, and it will occur a policy dependent penalty R_{Dataloss}.

Requirement 7. $\forall b, \forall d$
$R(\{b, d\}, SEND) = R_{Energy}(b, d) < 0$

That is, when sending the amount of data encoded by the quantum b to the distance specified by the quantum d, the sensor will occur a penalty (cost) of $R_{\text{Energy}}(b, d)$. Naturally, this cost involves factors such as overhead, transmission data path loss, and so on. However, this penalty is strictly determined by the hardware; it does not involve policy decision.

5.1.1. *Acquiring the transition probabilities*

Notice that the MDP defined in the previous section has $n \times m$ states, and therefore, $(n \times m)^2$ transition probabilities, which is a very large number even for a moderate number of quantums. For instance, if $n = 10$ and $m = 10$ we would need to compute 10 000 individual probabilities. However, requirements 1, 2, 3 set most of those probabilities to zero. Furthermore, requirement 4 guarantees that the number of *unique* transition probabilities is even lower:

$$\begin{aligned} &P_{\text{SEND}}(\{b, d\}, \{0, d'\}) \\ &\quad = P_{\text{DO-NOT-SEND}}(\{b, d\}, \{b+1, d'\}) \\ &\quad = P_{\text{DO-NOT-SEND}}(\{n-1, d\}, \{n-1, d'\}) \\ &\quad = P(d'_{t+1}|d_t) \end{aligned} \tag{5}$$

Thus, the only probabilities we need to measure are the ones of the form $P(d'_{t+1}|d_t)$, which is the conditional probability that the distance is d' provided that at the previous time point it was d. There are m^2 distinct probabilities of this form, that is, in the case of our running example, 100. Further simplifications are possible. As the mobile sinks are performing a continuous movement, the resulting transition matrix will have most of its values zero except on the diagonal and the values immediately above and below the diagonal. This basically means that the distance of the mobile sink does not jump over quantums—if it does, it means that either the sampling rate is to low, or the number of quantums is not sufficient. With this simplification, the number of unique probabilities to be calculated will be $3m$, that is 30 unique values for our running example.

These values can be easily acquired from historical information about the movement of the mobile sink. Due to the relatively small number of transition probabilities which need to be acquired, relatively short histories can be used.

5.2. MDP With History Information

MDPs are 'historyless,' that is, the decision to take an action is based only on the current state. The state, therefore, needs to encompass all the historical information necessary for decision making. The choice of the right amount of information is critical in maintaining the state space at a manageable size, and ultimately, can determine the quality of the decision making.

Let us now investigate potential ways of using historical information in the states of the MDP. In the historyless version of the previous section, we have defined the state as dependent of the buffer level and the distance to the mobile sink.

Note that the history of the buffer level is uninformative. If the buffer level in a given state is b,

DOI: 10.1002/wcm

the level of the previous state was $b - r$, where r is the data accumulation rate. The only case where multiple history alternatives are possible is when the buffer level is zero—this can happen if the previous action was a SEND. The state itself does not contain information about how much data was in the buffer before the SEND action—but anyhow, the future decision is not affected by this information. In conclusion, we can ignore the history of the buffer level in the state representation.

Things are different with the distance of the mobile sink. Let us assume that the current distance is 40 m. In the historyless current model, the decision to send or not send is made based on this information alone. However, if the sensor node knows the history of distances it might be able to make a more informed decision. For instance if the distances were 60, 50, 40, the node might conclude that a mobile sink is approaching, it will come closer with high probability and therefore it is worth waiting with the sending. However, if the values were 20, 30, 40 the node will conclude that the mobile sink is moving away and it should either send now or be prepared to wait for a longer time interval, until another mobile sink comes closer or the current mobile sink turns around.

In the first approximation, we might consider directly adding the history (for instance for the last several time periods) into the state. The state encoding would be $\{b, \{d_t, d_{t-1}, \ldots d_{t-k+1}\}\}$ where k is the number of time intervals the history is considering. The problem with this encoding is the explosion of the number of states, which is $m \times n^k$. For our running example, if we choose $n = 10$, $m = 10$, and $k = 4$, we have an MDP with 100 000 states and 10^{10} transition probabilities. Note that various considerations of our previous model reduced the number of probabilities we need to acquire to 30. Although some of the same simplifying assumptions are applicable here as well, 10^{10} is such a huge number, that even after all the simplifying assumptions, we still have a state space so large that acquiring the transition probabilities from observations will be unpractical.

Therefore, we need to consider a different, more compact encoding. We will therefore encode the history h in four discrete cases: STATIONARY, FAST-SLOPE-DECREASE, SLOW-SLOPE-DECREASE, and INCREASE. These classifications are determined by considering the last three readings of the distance. With this encoding, and considering the number of distance quantums $n = 10$, there are 40 different distance states. This yields a theoretical number of 1600 unique transition probabilities. However, some of the transitions are not possible, while others are exceedingly rare. For instance, the transition from (5, INCREASE) to (4, INCREASE) is not possible, because if the distance is reduced, the history cannot indicate an increase of the distance. On the other hand, the transition (0, INCREASE) to (10, INCREASE) is technically possible, but it requires a mobile sink moving unrealistically fast. Overall, the number of transitions which need to be considered are about 200–300, which number, although higher than in the case of the historyless encoding, is still manageable and can be acquired from historical information of a reasonable length.

As the reward does not depend on the history, the rewards associated with the transitions will be the same to the state-action pairs without history.

6. Experimental Results

We performed a series of experiments with a transmission scheduling scenario involving a field in which a number of mobile nodes are moving and collecting the data from the sensor nodes using a one-hop transmission. The mobility pattern of the mobile sinks was random waypoint [25]. 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 400×200 m, with $4 \ldots 20$ mobile sinks. The transmission range of the node was considered to be between 10–80 m, a realistic range for a sensor node (for instance [24] finds the transmission range of second generation Mica-2 motes to be between 20 and 50 m in an outdoor habitat). 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 I.

Table I. The parameters of the simulation experiments.

General settings	
Movement area	400×200 m
Simulation time	2500 s
Mobile sinks	
Number	$4 \ldots 20$ (10 default)
Velocity	1 m/s
Transmission range	80 m
Sensor nodes	
Buffer size	32 kB
Data rate	0.2 kB/s
Transmission range	$10 \ldots 80$ m (50 default)
Transmission power model	
Path loss index n	4
α_{11}	45 nJ/bit
α_2	0.001 pJ/bit/m^4

DOI: 10.1002/wcm

We have implemented this scenario in the YAES simulator framework [26]. In our experiments, we compare four different sensor implementations:

6.1. 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. The calculation of the optimal schedule took approximately 10–30 s on a 2.8 GHz Pentium 4 computer. Thus, we find that the Oracle Optimal algorithm is not a feasible on-board implementation choice for sensor nodes, even if the movement of the sinks is known. One the other hand, the schedule can be computed off-line (for instance, on the mobile sink) and transferred to the node. The schedule is essentially a list of the time moments when the node should transmit, and can be represented very compactly.

As expected, the OrOpt algorithm always outperforms the other approaches, and as such, serves as a baseline to the level of performance is possible for a given scenario. Note that 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.

6.2. Simple Heuristics (Simple)

This algorithm implements a simple rule-of-thumb heuristics. The agent does not transmit when the buffer is below 90 per cent full. When the buffer is more than 90 per cent full, it will transmit at the first available opportunity. Note that this is not a random algorithm, but a relatively good choice for most possible scenarios.

6.3. Markov Decision Process Without History Information (MDP)

This algorithm implements the MDP as described in Section 5.1. The distance was quantized into 10 quantums using the Equation (4). The buffer level was quantized into 30 equal size, 1 kB quantums.

The MDP was implemented using the jMarkov [27] library. The posterior probabilities were obtained by observing a sequence of 10 000 s with the given number of mobile sinks.

To maintain the cross-validation assumption, we were careful to separate the training data from the test data. For all the recorder experiments, the sensor had seen the given movement sequence the first time.

The MDP was solved using the value iteration algorithm. From the state values, we extracted the policy, which was represented, very compactly, by the list of states where the sensor makes the SEND decision. This choice is justified by the observation that there are a much lower number of SEND states than DO-NOT-SEND states. The learning process is relatively time consuming and probably needs to be executed off-line. However, the execution of a learned policy involves a limited amount of computation and it can be easily performed by the sensor. Essentially, at any moment when it needs to make a decision, the sensor identifies the state, by quantizing the distance to the closest mobile sink and the buffer level and checks whether the state is in the send lists.

For this historyless MDP model, we find that the MDP has 403 reachable states, with 10–30 states being SEND states (depending on the transmission range and the number of mobile sinks).

6.4. Markov Decision Process With History Information (MDP+h)

This algorithm implements the MDP where the state includes historical information as described in Section 5.2. The implementation and training details are similar to the case of MDP without history information. The main difference is that due to the extra state information, we have more than three times more accessible states (1300–1500 depending on the transmission range), and about 30–60 SEND states. This, however, is still within the possibilities of the jMarkov framework, and the ability of the sensor node to implement the policy.

An additional design choice for both MDP models is the discount rate (or its dual, the interest rate). Technically, the transmission scheduling problem does not have a 'natural' discount rate. There is no technical reason for a sensor to prefer loosing data tomorrow *versus* loosing data today. However, the value iteration solver of an MDP needs a discount rate lower than 1 to converge. For this reason, we choose a very slow discount rate of 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, and average measurements retained. We collected the following measurements:

- *Total transmission energy.* The total energy consumed by the sensor node for transmissions over the timespan of the scenario. If all other parameters

DOI: 10.1002/wcm

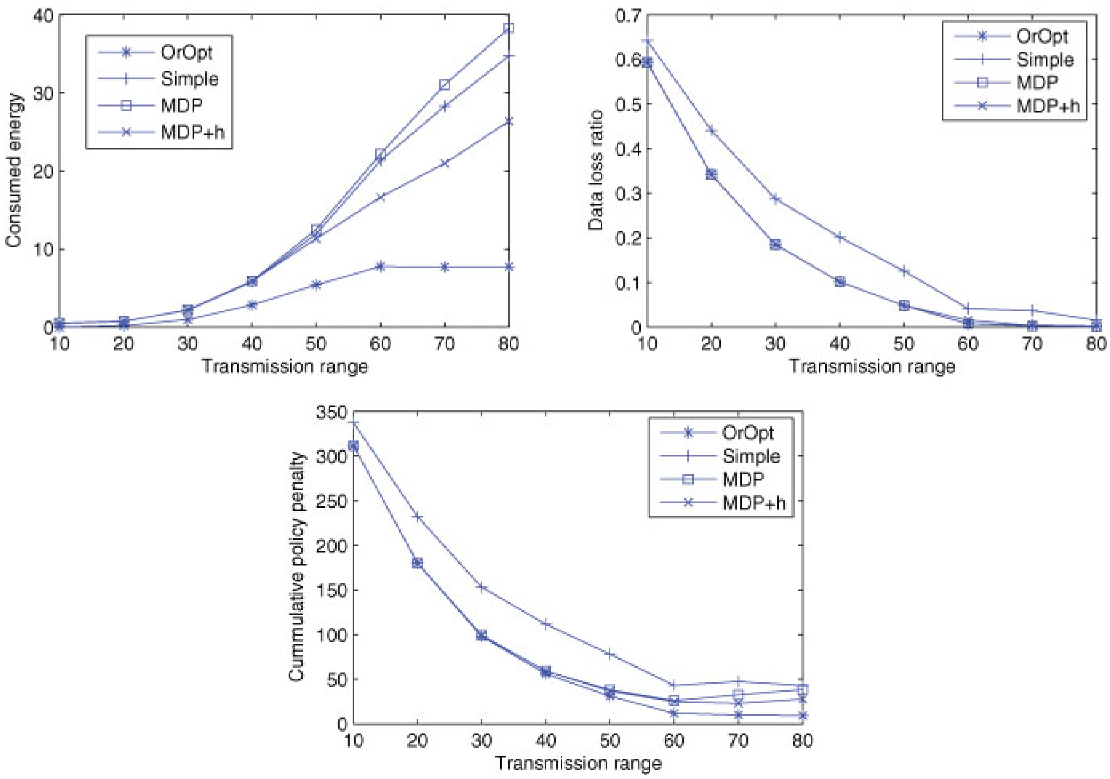

Fig. 3. Scenario 1: measurement results averaged over 10 runs for the four considered transmission scheduling algorithms for various values of the transmission range. The policy requires a balance of minimizing data loss and minimizing energy consumption. The graphs represent the total transmission energy (upper left), data loss ratio (upper right), and cummulative policy penalty (bottom).

are equal, the lower the total transmission energy, the more efficient the algorithm.

- *Data loss ratio.* The ratio of the data loss caused by buffer overflow to the total amount of data captured by the sensor. If all other parameters are equal, the lower the data loss ratio, the more efficient the algorithm.
- *Cummulative policy penalty (CPP).* This is a composite measure calculated as a function of the total energy and the number of lost packets. The exact calculation is dependent on the policy. For the Oracle Optimal and the two MDP-based algorithms, this measure is the optimization criterion. The lower the value of the policy penalty, the more successful are the algorithms in accomplishing their stated goals.

Finally, we repeated our measurements for three different scenarios, differentiated by the expression of the cummulative policy penalty. Different policies can be implemented by setting various values for the data loss penalty. The energy part of the policy penalty is determined by physical constraints so it is not available for modification by the user.

- *Scenario 1: Balanced objectives.* We assume that the data loss penalty is relatively high, but it can be offset by energy savings only in exceptional situations. Thus, the sensors pursue a balanced strategy between maximizing energy use and minimizing data loss.
- *Scenario 2: Data loss reduction priority.* In this case, we set the data loss penalty to a very high value (in our case, 10 000). With this value, the penalty for lost data can never be offset by energy savings. Thus, the agent will have the primary objective to reduce the data loss. Only in the cases of equal amount of data loss will it make transmission decisions in function of energy considerations.
- *Scenario 3: Energy conservation priority.* In this case, we set the data loss penalty to be equal to sending the data with the maximum transmission range. Note that we cannot set the data loss penalty to zero; by doing so, the algorithms would optimize the transmission energy by losing all the collected data.

In the following, we present and discuss the results for these three scenarios.

6.5. Scenario 1: Balanced Objectives

Figure 3 shows the measurements of consumed energy, data loss ratio and CPP for various settings of the transmission range.

As a first observation, in general, the cummulative policy penalty is decreasing with the increase of the transmission range. Having a longer transmission range, the sensor can have more options, which it can use to reduce the amount of data loss. An interesting anomaly can be seen for the two MDP-based sensors: the policy score actually shows a very slight but noticeable increase at the transmission ranges of 60–80 m. This anomaly can be explained with reference to the data loss chart: the MDP-based algorithms have learned a transmission policy which attempts to transmit as soon as the mobile sink gets in the transmission range. As the transmission range increases, this policy, although advantageous from the point of view of data loss, is more expensive from the point of view of energy consumption.

In the comparison among the four algorithms, we find that the score of the OrOpt algorithm is the best, followed in order by MDP+h, MDP and, at some distance, by Simple. For ranges between 10 and 40 m, the OrOpt, MDP+h, and MDP obtain essentially the same CPP score. From Figure 3 top right, we see that the two MDP-based algorithms can essentially match OrOpt for minimizing the data loss; the simple heuristic performs much worse.

The difference between the MDP and MDP+h is visible on the consumed energy graph (Figure 3, top left). Here, MDP+h clearly outperforms MDP. This is due to the higher quality decisions, because the MDP with history information can better predict the future distance of the mobile sink.

Figure 4 shows the evolution of the measurements for varying number of mobile sinks. The overall CPP score shows a decreasing trend with the number of mobile sinks, as the sensor node is visited more often by sinks. Again, we find that the two MDP models match closely the OrOpt for the data loss ratio. This result,

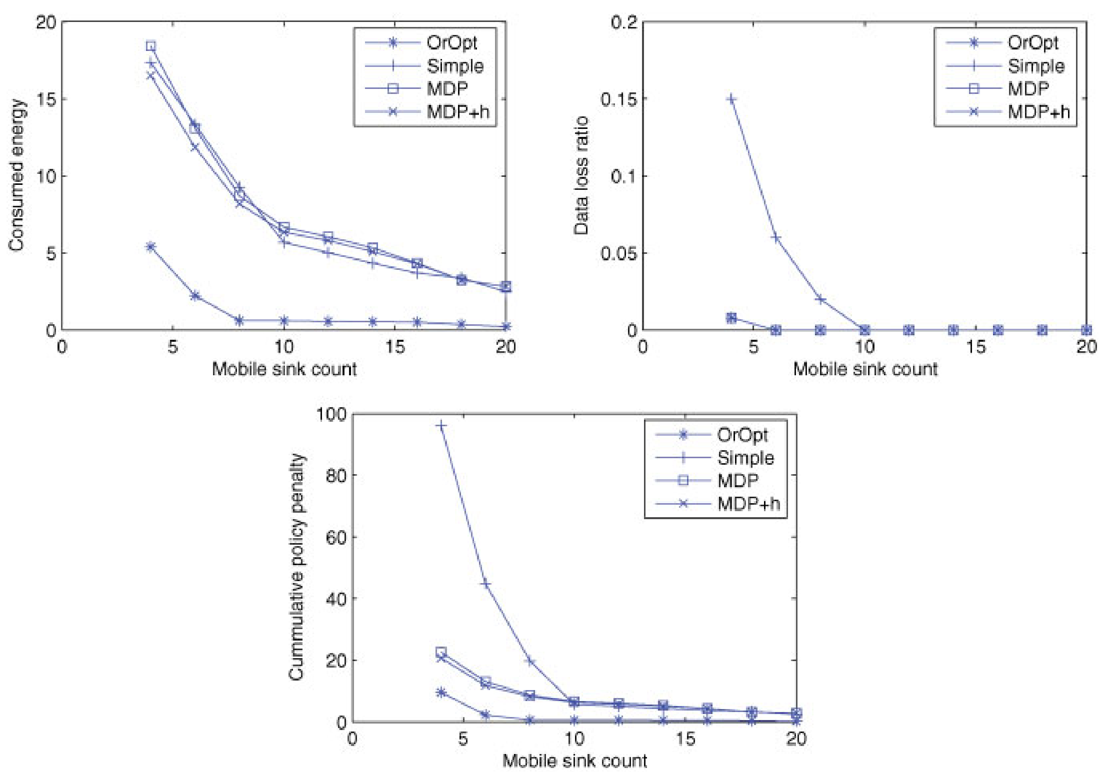

Fig. 4. Scenario 1: measurement results averaged over 10 runs for the four considered transmission scheduling algorithms for various values of the number of mobile sinks. The policy requires a balance of minimizing data loss and minimizing energy consumption. The graphs represent the total transmission energy (upper left), data loss ratio (upper right), and cummulative policy penalty (bottom).

DOI: 10.1002/wcm

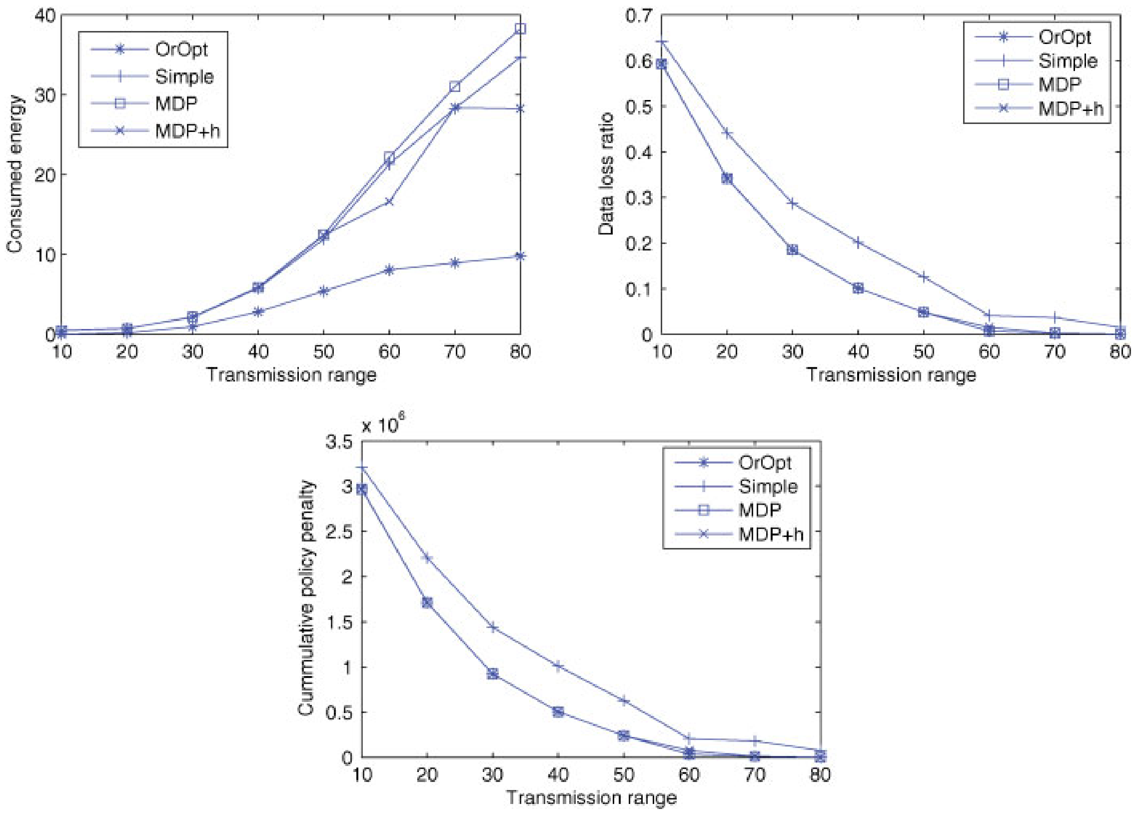

Fig. 5. Scenario 2: measurement results averaged over 10 runs for the four considered transmission scheduling algorithms for various values of the transmission range. The policy requires the sensor to minimize data loss. The graphs represent the total transmission energy (upper left), data loss ratio (upper right), and cummulative policy penalty (bottom).

however, is achieved with various levels of energy expenditure

6.6. Scenario 2: Priority in Minimizing Data Loss

In this scenario, we are using a cost function which assigns a weight of 10 000 to every lost kilobytes of data. With this setting, the OrOpt, MDP, and MDP+h algorithms will make decisions such that they will minimize the data loss. The energy consumption will be considered only in cases when the decision does not affect the probability of data loss.

The experimental results using these strategy are shown in Figures 5 and 6. The graphs show similar trends with the scenario with the balanced priority. The differences can be summarized in the following points:

- The cummulative policy penalty of the MDP and MDP+h algorithms are much closer to the OrOpt algorithm for both variable transmission range and the variable number of mobile sinks experiments.
- The consumed energy is virtually identical for MDP and MDP+h. We should compare this with the balanced strategy, where the MDP+h was significantly better. The reason for this phenomena is that in the data loss minimization scenario, the MDP's have little motivation to optimize their decisions for reducing the consumed energy.

6.7. Scenario 3: Priority for Minimizing Energy Consumption

In this scenario, the data loss penalty is set up such that the penalty of loosing a kilobyte of data is the same as transmitting it at the distance of 80 m. Note that with this setup the optimization algorithms will still attempt to send as much data as possible, but they have more leverage, by the ability to occasionally trade some lost data for lower values of the energy consumption.

The results of the experiments with this strategy are shown in Figures 7 and 8. The overall trends are similar to the previous two scenarios. The main observations are as follows:

DOI: 10.1002/wcm

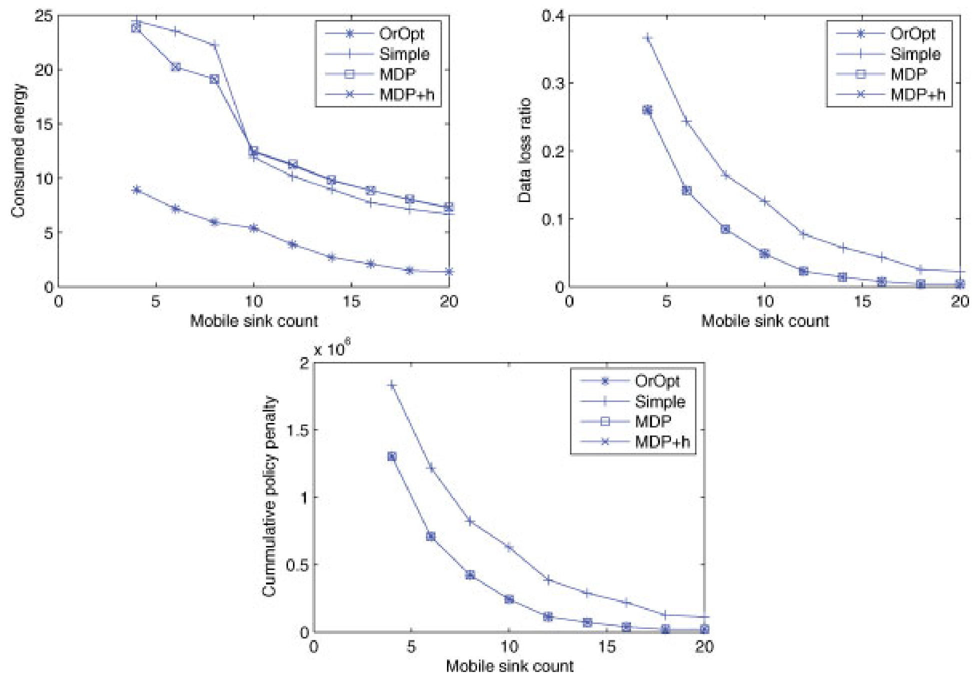

Fig. 6. Scenario 2: measurement results averaged over 10 runs for the four considered transmission scheduling algorithms for various values of the number of mobile sinks. The policy requires the sensor to minimize data loss. The graphs represent the total transmission energy (upper left), data loss ratio (upper right), and cummulative policy penalty (bottom).

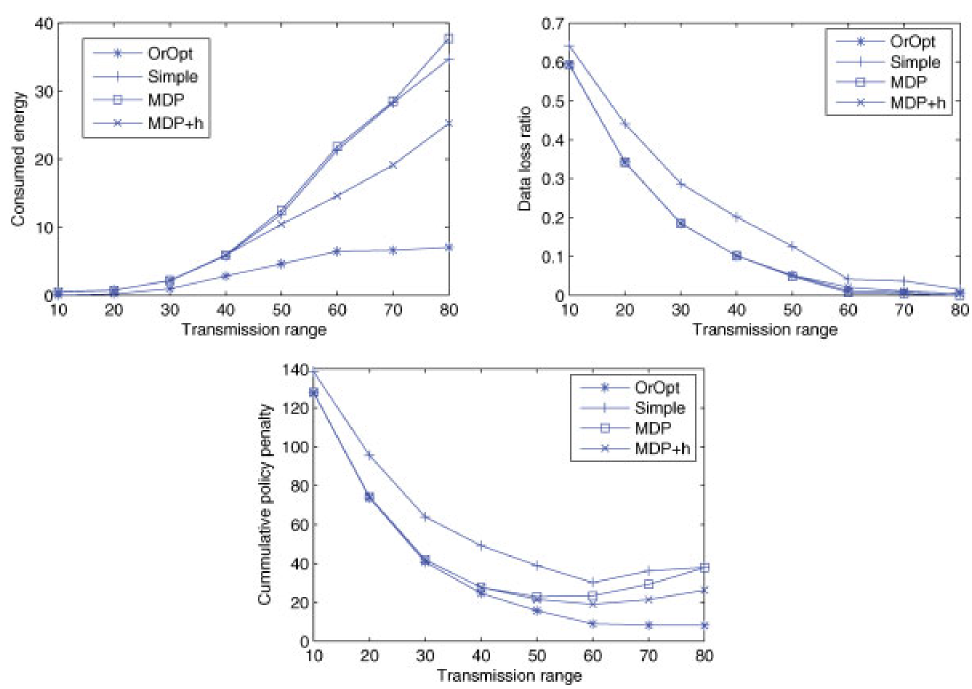

Fig. 7. Scenario 3: measurement results averaged over 10 runs for the four considered transmission scheduling algorithms for various values of the transmission range. The policy requires the sensor to minimize total transmission energy, and considers the penalty of data loss as data transmitted at the maximum transmission distance. The graphs represent the total transmission energy (upper left), data loss ratio (upper right), and cummulative policy penalty (bottom).

DOI: 10.1002/wcm

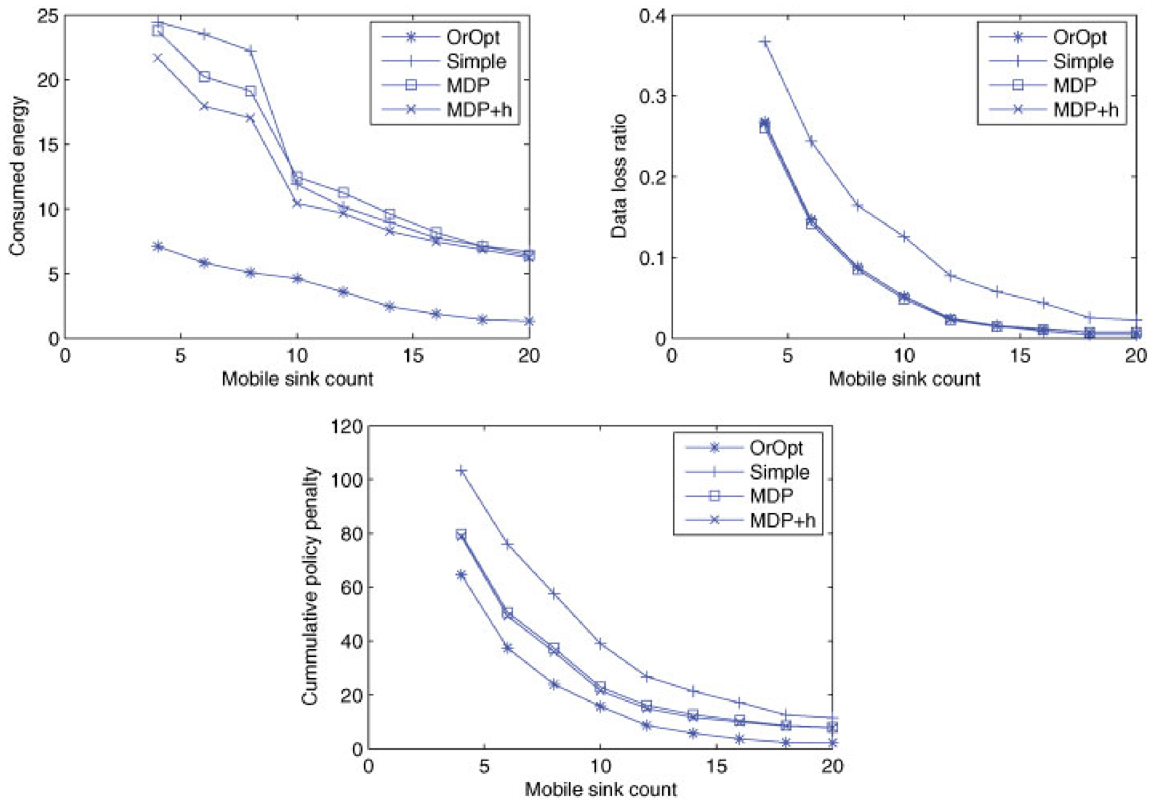

Fig. 8. Scenario 3: measurement results averaged over 10 runs for the four considered transmission scheduling algorithms for various values of the number of mobile sinks. The policy requires the sensor to minimize total transmission energy, and considers the penalty of data loss as data transmitted at the maximum transmission distance. The graphs represent the total transmission energy (upper left), data loss ratio (upper right), and cummulative policy penalty (bottom).

- The MDP+h model is consistently better than MDP from the point of view of the cummulative policy penalty. The difference is increasing with the transmission range, reaching values of close to 40 per cent.
- MDP and MDP+h still performs very well from the point of view of data loss ratio (only marginally worse than the OrOpt algorithm).

6.8. Summary of Findings

Overall, we find that the choice of the transmission scheduling algorithms makes a significant difference in the performance.

Both MDP and MDP+h can get very close to the data loss ratio obtained by the optimal algorithm. This is a somewhat surprising, but positive conclusion, as we expected the advantage of the optimal algorithm to be much higher, as it has advance knowledge of the movement of the sinks. It turns out that the difference is not significant, both MDP and MDP+h have learned policies which allows them to achieve near-optimal data loss values.

The cost of the successful data transmission, however, is a different matter. It was found that the oracle optimal algorithm can achieve a slightly lower data loss rate with up to 70 per cent less power than the other algorithms. Also, MDP+h can outperform MDP with up to 40 per cent in certain scenarios.

These are significant differences and fully justify the effort put into improving the transmission scheduling algorithms. From the ones considered in our study, the best algorithm for practical deployment was found to be MDP+h. However, if the mobility pattern of the mobile sinks is known in advance, the optimal algorithm offers sufficient improvement to justify its calculation on an off-line, high performance computer and the transmission of the pre-computed schedule to the sensor nodes.

7. Conclusions

In this paper, we investigated the problem of transmission scheduling in sensor networks with mobile sinks. We presented an optimal algorithm which

DOI: 10.1002/wcm

requires advance knowledge of the mobility patterns of the mobile sinks. We also presented two variants of decision theoretic algorithms based on MDP, one with and one without history information encoded in the state. Through an experimental study, we compared the proposed algorithms against a simple heuristics. We found that, as expected, the optimal algorithm performed best, but in many scenarios the MDP-based algorithms showed a performance close to the optimal from the point of view of minimizing data loss. The MDP approach with encoded history information performed better from the point of view of consumed transmission power and appears to be the algorithm with the best balance between implementation and deployment difficulty and performance.

References

1. Rappaport T. *Wireless Communications: Principles & Practice*. Prentice-Hall: Upper Saddle River, NJ, 1996.
2. Shah R, Roy S, Jain S, Brunette W. Data mules: modeling a three-tier architecture for sparse sensor networks. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA-03)*, May 2003; 30–41.
3. Kim HS, Abdelzaher T, Kwon W. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys-03)*, 2003; 193–204.
4. Bhattacharya S, Kim H, Prabh S, Abdelzaher T. Energy conserving data placement and asynchronous multicast in wireless sensor networks. In *Proceedings of ACM Mobile Systems, Applications, and Services (MobiSys-2003)*, May 2003; 173–186.
5. Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom-2000)*, August 2000; 56–67.
6. Luo H, Ye F, Cheng J, Lu S, Zhang L. TTDD: two-tier data dissemination in large-scale wireless sensor networks. *Wireless Networks* 2005; **11**(1): 161–175.
7. Jetcheva J, Johnson D. Adaptive demand-driven multicast routing in multi-hop ad hoc networks. In *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc-2001)*, October 2001; 33–44.
8. Baruah P, Urgaonkar R, Krishnamachari B. Learning-enforced time domain routing to mobile sinks in wireless sensor fields. In *Proceedings of 29th Annual IEEE International Conference on Local Computer Networks (LCN-2004)*, November 2004; 525–532.
9. Wang Z, Basagni S, Melachrinoudis E, Petrioli C. Exploiting sink mobility for maximizing sensor networks lifetime. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS-05)*, January 2005.
10. Chakrabarti A, Sabharwal A, Aazhang B. Using predictable observer mobility for power efficient design of sensor networks. In *Proceedings of the Second International Workshop on Information Processing in Sensor Networks (IPSN-03)*, April 2003; 129–145.
11. Vincze Z, Vida R. Multi-hop wireless sensor networks with mobile sink. In *Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT-2005)*, 2005; 302–303.
12. Tong L, Zhao Q, Adireddy S. Sensor networks with mobile agents. In *Proceedings of the IEEE Military Communication Conference (MILCOM-2003)*, October 2003.
13. Chen C, Ma J, Yu K. Designing energy-efficient wireless sensor networks with mobile sinks. In *Proceedings of the Workshop on World-Sensor-Web (WSW-2006) at the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys-06)*, October 2006.
14. Kansal A, Rahimi M, Estrin D, Kaiser W, Pottie G, Srivastava M. Controlled mobility for sustainable wireless sensor networks. In *Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON-04)*, October 2004; 1–6.
15. Gandham S, Dawande M, Prakash R, Venkatesan S. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Proceedings of the IEEE GLOBECOM-03*, Vol. 1, December 2003; 377–381.
16. Ye F, Chen A, Lu S, Zhang L. A scalable solution to minimum cost forwarding in large sensor networks. In *Proceedings of International Conference on Computer Communications and Networks (ICCCN-01)*, 2001; 304–309.
17. Wang W, Srinivasan V, Chua K-C. Using mobile relays to prolong the lifetime of wireless sensor networks. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom-2005)*, September 2005; 270–283.
18. Luo J, Hubaux J-P. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, Vol. 3, March 2005; 1735–1746.
19. Luo J, Panchard J, Piorkowski M, Grossglauser M, Hubaux J-P. MobiRoute: routing towards a mobile sink for improving lifetime in sensor networks. In *Proceedings of 2nd IEEE/ACM Intl Conference on Distributed Computing in Sensor Systems (DCOSS-2006)*, June 2006; 480–497.
20. Woo A, Tong T, Culler D. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys-03)*, November 2003; 14–27.
21. Olariu S, Eltoweissy M, Younis M. ANSWER: AutoNomouS netWorked sEnsoR system. *Journal of Parallel and Distributed Computing* 2007; **67**(1): 111–124.
22. Song L, Hatzinakos D. Dense wireless sensor networks with mobile sinks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-05)*, Vol. 3, March 2005; 677–680.
23. Zhao Q, Tong L. Distributed opportunistic transmission for wireless sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-04)*, Vol. 3, May 2004; 833–836.
24. Cerpa A, Busek N, Estrin D. Scale: a tool for simple connectivity assessment in lossy environments. Technical Report 0021, Center for Embedded Network Sensing (CENS)—UCLA, September 2003.
25. Johnson DB, Maltz DA. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, Imielinski T, Korth H (eds). Academic Publishers: Kluwer, 1996; 153–182.
26. Bölöni L, Turgut D. YAES—a modular simulator for mobile networks. In *Proceedings of the 8-th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM-05)*, October 2005; 169–173.
27. Riaño G, Góez J. jMarkov: an object-oriented framework for modeling and analyzing Markov chains and QBDs. In *Proceedings of the International Workshop on Tools for solving Structured Markov Chains (SMCTools-06)*, October 2006.

DOI: 10.1002/wcm

Authors' Biographies

Ladislau Bölöni is an Assistant Professor with the School of Electrical Engineering and Computer Science at University of Central Florida. He received a Ph.D. from the Computer Sciences Department of Purdue University in May 2000. He received a Master of Science degree from the Computer Sciences Department of Purdue University in 1999 and Diploma Engineer degree in Computer Engineering with Honors from the Technical University of Cluj-Napoca, Romania in 1993. He received a fellowship from the Computer and Automation Research Institute of the Hungarian Academy of Sciences for the 1994–1995 academic year. He is a senior member of IEEE, member of the ACM, AAAI, and the Upsilon Pi Epsilon honorary society. His research interests include autonomous agents, grid computing, and wireless networking.

Damla Turgut is an Assistant Professor with the School of Electrical Engineering and Computer Science at University of Central Florida. She received her B.S., M.S., and Ph.D. degrees from the Computer Science and Engineering Department of University of Texas at Arlington in 1994, 1996, and 2002, respectively. She has been included in the WHO's WHO among students in American Universities and Colleges in 2002. She has been awarded outstanding research award and has been the recipient of the Texas Telecommunication Engineering Consortium (TxTEC) fellowship. She is a member of IEEE, member of the ACM, and the Upsilon Pi Epsilon honorary society. Her research interests include wireless networking and mobile computing, distributed systems, and embodied agents.

DOI: 10.1002/wcm