Three heuristics 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
Email: turgut@eecs.ucf.edu,lboloni@eecs.ucf.edu

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Design, Measurement, Experimentation

Keywords

Sensors, Sensor Networks, Transmission scheduling

ABSTRACT

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 which 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 the choosing the appropriate moment of transmission. In general, the transmission energy increases quickly with the distance, thus it makes sense for the nodes to transmit when one of the mobile sinks is in close proximity.

One of the side effects of the one-hop transmission is that the node is responsible only for its own data and its own energy resources. Thus, it makes good sense to treat the sensor node as an autonomous agent which maximizes the utility of the collected and transmitted data, while minimizing the energy expenditure.

This paper proposes several approaches for such an agent controlled sensor node. As the basis of comparison, we describe a dynamic programming-based approach for the optimal policy in the context of full world knowledge. 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.

1. INTRODUCTION

Traditional sensor networks are composed of a set of low power sensor nodes which 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. 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, 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 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 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 a dynamic programming based optimal algorithm. The latter, however, is not a practical implementation alternative due both of its information requirements (it requires advance knowledge of the movement patterns of the mobile sink) 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 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. 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 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.

Let us now consider the objectives of the nodes. In the big picture the nodes are striving to transmit all the observations (that is, 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 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 [12]:

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

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 α_{11} =45 nJ/bit and α_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 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 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.

Routing towards 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 towards the mobile sink or towards sensor nodes which are periodically visited by a sink.

The MULE (Mobile Ubiquitous LAN Extension) [13] 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 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.

Scalable Energy-efficient Asynchronous Dissemination (SEAD) [6] 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.

Hybrid Learning-Enforced Time Domain Routing (HLETDR) [1] aims to deliver sensor data towards 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 towards the *moles*, sensor nodes located within the sink's path.

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 Z. Wang et. al. [18]. 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 5 times compared with a network with a static sink.

The data collection process is modeled as a queueing system in Chakrabarti et. al. [3] 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 and Vida [16] examine 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) [15] architecture was proposed 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 (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 Two-Tier Data Dissemination (TTDD) [8] 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.

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 [4] 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., [7] 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.

In Gandham et. al. [5], 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. [17] 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 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.

A combination of base station mobility and multi-hop routing strategy are proposed by Luo and Hubaux [9] 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 [10], 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 towards the mobile sink which changes its location based on route traces. A routing protocol forwarding data towards 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. [11], 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. **Transmission scheduling** is the process of determining when to transmit the buffered data.

Song et. al. [14] 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.

A distributed opportunistic information retrieval algorithm that uses channel state information (CSI) is proposed by Zhao and Tong [19]. 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.

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 $D = (d_{t_{start}} \ldots d_{t_{stop}})$, 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 $A = \{t_{start} < a_1, a_2, ...a_k = t_{stop}\}$, $a_i < a_{i+1}$ 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:

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

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

Proof: Let us assume that there is a different schedule A_1' for which $P_{total}(A_1) > P_{total}(A_1')$. Then the schedule A' obtained from the concatenation of A_1' and A_2 will have a total power consumption $P_{total}(A') = P(A_1') + P(A_2) < P(A_1) + P(A_2) = P(A)$, 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 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. THREE HEURISTICS

In the following, we propose three heuristic algorithms for making the transmission decision. As opposed to the oracle optimal algorithm, which is based on a search in the space of possible solutions, these algorithms make their decision based on very simple calculations, which do not explore the solution space and do not plan for the future transmissions.

In the remainder of this section we will use the following notations:

M	the current buffer content
M_{full}	the size of the buffer
r	data collection rate
d_{tr}	transmission rate
d	current distance of the closest mobile sink

5.1 H1: Human-inspired simple heuristics

The first heuristic we are presenting attempts to mimic the human decision process for the transmission scheduling. It 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 which the human subjects were developed 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 persons said that that does not appear to be a good strategy.

The heuristics is based on three parameters:

d_{opt}: optimal distance. This is the distance which 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 which is hit with a relative certainty during the maximum collection interval.

M_L: too low to transmit. Represents the level at which the amount of collected data is too low to justify its transmission, for instance because the transmission over

M_H: buffer emergency level. This represents an amount of collected data which puts the system in danger of a buffer overflow. 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 2. Obviously, the approach needs to be calibrated by setting the values for the d_{opt}, M_L and M_H values. These values depend on many factors. For instance, if the transmission overhead is 0, $M_L = 0$. If in the system there is always an actuator in the transmission range of the sensor, then $M_H = M_{full}$ (the buffer capacity).

Finally, there is the problem of the relative payoff of a close range transmission versus the penalty for data loss. Although, intuitively, the data loss is a more serious problem, we need to rate that the exhaustion of the battery of a sensor node with no recharge capabilities can be even more serious. This input is also encoded in the M_H parameter.

Algorithm 2 H1: Human inspired simple heuristics

```
Function transmitDecision_H1(d, M, M_L, M_H, d_opt)
    If M > M_L
        If d < d_opt
            transmit
        Else If M < M_H
            transmit
        Else
            wait
        EndIf
    Else
        wait
    EndIf
EndFunction
```

5.2 H2: Stochastic transmission

The stochastic transmission heuristics transmits randomly with a probability distribution which is affected by two factors. The first factor is the level of the buffer. In general, if the buffer is empty, this contributes a 0 value, while if the buffer is full, it contributes a 1 value to the probability.

$$p_b = \frac{M}{M_{full}} \tag{3}$$

The second factor is the distance of the mobile sink. We assume that there is a minimum distance d_{min}, where any distance smaller that that it can not be translated into an energy advantage - this is practically the transmission range of the node at the lowest transmission energy setting. 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 which achieves these values is:

$$p_d = \frac{d_{tr} - \max(d, d_{min})}{d_{tr} - d_{min}} \tag{4}$$

Finally, we introduce a parameter w which represents the balance between the two contributions.

$$p = w \cdot \frac{M}{M_{full}} + (1 - w) \cdot \frac{d_{tr} - \max(d, d_{min})}{d_{tr} - d_{min}} \tag{5}$$

5.3 H3: Constant risk

The reason for a node not to transmit is because it hopes 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 does a decision carry and then take decisions based on a constant risk factor.

Algorithm 3 H2: Stochastic transmission

```
Function transmitDecision_H2(d, M, d_min, w)
    p = w · M/M_full + (1 − w) · (d_tr − max(d, d_min))/(d_tr − d_min)
    generate u ∈ [0,1] from a uniform random distribution
    If u ≤ p
        transmit
    Else
        wait
    EndIf
EndFunction
```

The goal is to prevent the algorithm from being too bold in one occasion and too cautious in others.

To implement this algorithm, the heuristics maintains a two-dimensional array 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 will 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 measurement 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 is probability is lower than p_{risk} the node will send, otherwise it will wait.

Normally, p_{risk} should be put to a very high number, such as 0.98. Note that an OP value of 1 does not mean that there is no risk involved, only that in the history of the node, those types of situations finished "well", which might not be true in the future.

Algorithm 4 H3: Constant risk

```
Function transmitDecision_H3(d, M, p_risk)
    t_q = quantization of remaining time
    d_q = quantization of distance to the sink
    If OP[t_q][d_q] < p_risk
        transmit
    Else
        wait
    EndIf
EndFunction
```

6. EXPERIMENTAL STUDY

6.1 Experimental setup

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. We have assumed that the speed of the mobile sink was 1 m/sec 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 meters, with 4…20 mobile sinks. The transmission range of the node was considered to be between 10-80 meters, a realistic range for a sensor node. Finally, we assumed a 32kB buffer and a data rate of 0.2kB/sec. The parameters of the simulation environment are summarized in Table 1.

We have implemented this scenario in the YAES simulator framework [2]. In our experiments we compare four different sensor implementations:

Table 1: The parameters of the simulation experiments

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

Oracle Optimal (OrOpt): This implementation has advance knowledge of the movements of the mobile sinks and calculates an optimal schedule which minimizes the given cummulative policy penalty (CPP). The implementation follows the description in Section 4. The calculation of the optimal schedule took approximately 10-30 seconds on a 2.8GHz 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.

Human inspired (H1:HI): This is an implementation for the algorithm described in Section 5.1 and the pseudo-code in Algorithm 2. The free parameters were chosen as: $M_L = 0.2$, $M_H = 0.9$ 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 3. The relative weight factor is chosen $w = 0.5$.

Constant risk (H3:CR): This is an implementation for the algorithm described in Section 5.3 and the pseudo-code in Algorithm 4. 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, 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 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 lower the value of the policy penalty, the more successful are the algorithms in accomplishing their stated goals.

6.2 Results

Figure 1 shows the measurements of consumed energy, data loss ratio and cummulative policy penalty (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. However, we can also note a small anomaly, the moderate increase of the CPP for the cases where the transmission range is longer for the three heuristics. What this shows is that the heuristics in their current form are overly cautious - they prefer to decrease the data loss ratio at the expense of power consumption.

As a comparison between the four algorithms, we find that the OrOpt algorithm, as expected outperforms the other algorithms in the CPP measurement. The human inspired heuristic H1 performs the worst, especially for the middle of the range, while H2 and H3 have roughly comparable results, with the stochastic algorithm H2 being slightly better at most transmission ranges.

The graphs for the consumed energy and the data loss ratio show that the different algorithms obtained their score in a very different way. Overall, the consumed energy graph increases with the transmission range, as the node is taking advantage of the longer transmission range to reduce the data loss, at the same time using more energy. The consumed energy graph of the OrOpt algorithm is peculiar, in becoming suddenly horizontal at the transmission range of 60m. This is due to the fact that at this transmission range the algorithm can already achieve close to 0% data loss, thus the node will refrain from transmitting from longer distances. The heuristics, however, do not have information about the future movement of the nodes, thus, they perform long distance transmission relatively often.

As expected, the data loss ratio graphs are decreasing with the increase in the transmission range. Up to the transmission range of 60m, even the optimal algorithm looses some amount of data. The ranking between the individual heuristics is different in parts of the graph. In the 0-60m range the OrOpt algorithm has the lowest data loss, followed by the stochastic, constant risk, and, at some distance, the human inspired algorithm. At a transmission range of over 60m, the stochastic algorithm H2 matches the data loss of the optimal algorithm (albeit, as we had seen, at the cost of a much higher energy consumption), while the human inspired H1 and the constant risk H2 algorithms perform very close to each other.

Figure 2 shows the evolution of measurements for a transmission range of 50m and a varying number of mobile sinks. First looking at the cummulative policy penalty CPP, we find that for all the algorithms the CPP decreases with the increase of the number of mobile sinks. This is expected, because with more mobile sinks, the probability of one being close is increasing, reducing the data loss and creating opportunities for saving energy with small distance, low power transmissions.

The degree at which different algorithms are able to take advantage of this, however, varies. The three main groups are the OrOpt algorithm, clearly the winner at all counts of mobile sinks, followed by the stochastic and the constant risk, and the human inspired algorithm, at some distance. However, for numbers of mobile sinks larger than 10, there is virtually no difference between the three heuristics.

The remaining two graphs, for consumed energy and data loss ratio explain the difference. While the consumed energy is roughly equivalent for the three heuristics, the human inspired heuristics has a much higher data loss ratio for $n < 10$ mobile sinks, due to a more optimistic waiting policy. To put things in perspective, the data loss ratio is relatively small for all algorithms; even for H1 where it is 16% at $n = 3$ and 6% at $n = 6$.

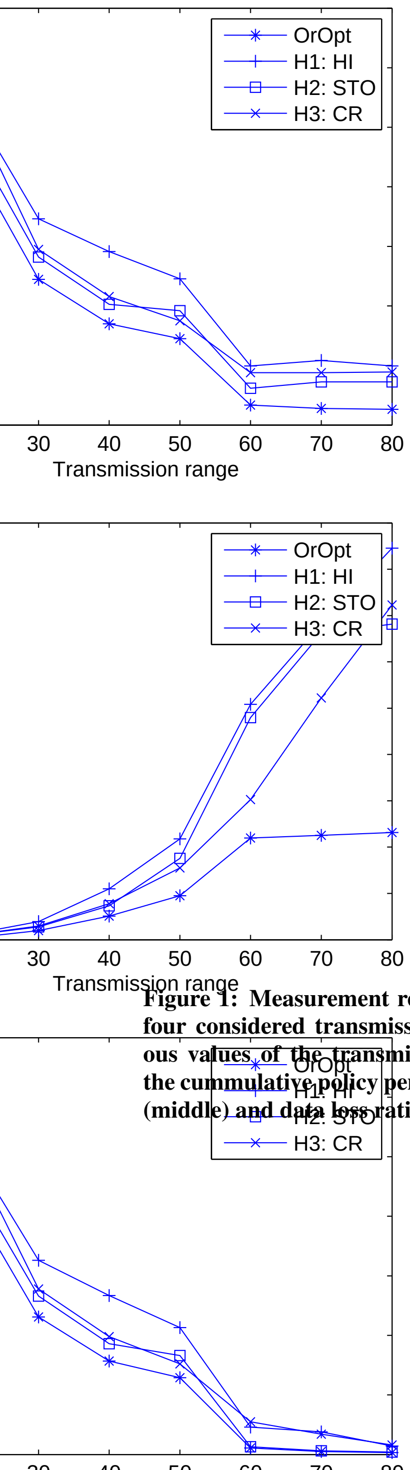

Figure 1: Measurement results averaged over 10 runs for the four considered transmission scheduling algorithms for various values of the transmission range. The graphs represent the cummulative policy penalty (top), total transmission energy (middle) and data loss ratio (bottom)

7. CONCLUSIONS

In this paper, we investigated the problem of transmission scheduling in sensor networks where the nodes communicate their collected data with one-hop transmissions to the mobile sinks. This allows us to take an agent approach to the behavior of the node, where each node tries to maximize its utility by minimizing energy consumption and data loss. We presented an optimal algorithm which requires advance knowledge of the mobility patterns of the mobile sinks. Then, we introduced three heuristics which rely only on local and historical information.

The first conclusion we can draw from our experimental study is that the choices made during transmission scheduling makes a significant difference in the energy consumption and data loss of sensors. The differences can reach values larger than 300% for challenging environments (low transmission range, and/or low number of mobile nodes). In scenarios with a favorable environment the differences are much lower.

Overall, the three heuristics we designed performed in a satisfactory way, but naturally, they had a lower performance than the optimal algorithm. Rather consistently, the human inspired algorithm H1 performed worse than the others. Note, however, that this might be different for other values of the parameters. H1 has a very large parameter space with three parameters, which makes it likely that by careful tuning its performance can be improved for a particular environment setting. What makes the H2 and H3 heuristics more desirable is the fact that they obtain good results without extensive tuning.

Regarding the suitability of the algorithm for deployment in a practical setting, all algorithms have a simple implementation which can readily be implemented in sensor nodes. The H3 heuristic, however, uses a probability table which, depending on the quantization level used might have an impact on the limited memory resources of the nodes.

8. REFERENCES

[1] P. Baruah, R. Urgaonkar, and B. Krishnamachari. Learning-enforced time domain routing to mobile sinks in wireless sensor fields. In *Proc. of 29th Annual IEEE International Conference on Local Computer Networks (LCN-2004)*, pages 525–532, November 2004.

[2] L. Bölöni and D. Turgut. YAES - a modular simulator for mobile networks. In *Proc. of the 8-th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM-05)*, pages 169–173, October 2005.

[3] A. Chakrabarti, A. Sabharwal, and B. Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In *Proc. of the Second International Workshop on Information Processing in Sensor Networks (IPSN-03)*, pages 129–145, April 2003.

[4] C. Chen, J. Ma, and K. Yu. Designing energy-efficient wireless sensor networks with mobile sinks. In *Proc. of the Workshop on World-Sensor-Web (WSW-2006) at the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys-06)*, October 2006.

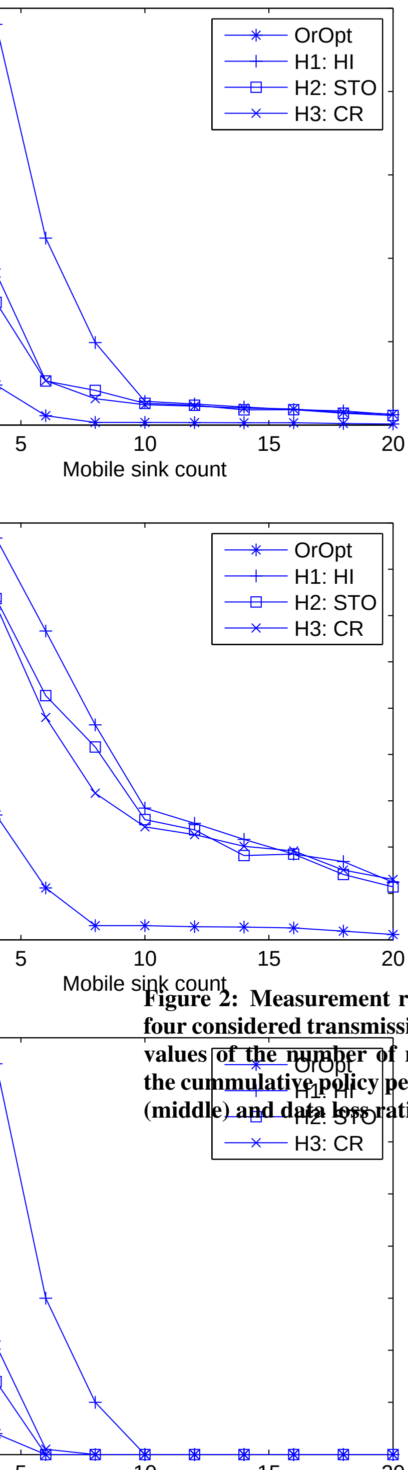

Figure 2: Measurement results averaged over 10 runs for the four considered transmission scheduling algorithms for various values of the number of mobile sinks. The graphs represent the cummulative policy penalty (top), total transmission energy (middle) and data loss ratio (bottom)

[5] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Proc. of the IEEE GLOBECOM-03*, volume 1, pages 377– 381, December 2003.

[6] H.S.Kim, T. Abdelzaher, and W. Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *Proc. of the 1st International Conference on Embedded Networked Sensor Systems (SenSys-03)*, pages 193–204, 2003.

[7] A. Kansal, M. Rahimi, D. Estrin, W. Kaiser, G. Pottie, and M. Srivastava. Controlled mobility for sustainable wireless sensor networks. In *Proc. of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON-04)*, pages 1–6, October 2004.

[8] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang. TTDD: Two-tier data dissemination in large-scale wireless sensor networks. *Wireless Networks*, 11(1):161–175, January 2005.

[9] J. Luo and J.-P. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *Proc. of IEEE INFOCOM*, volume 3, pages 1735–1746, March 2005.

[10] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J.-P. Hubaux. MobiRoute: Routing towards a mobile sink for improving lifetime in sensor networks. In *Proc. of 2nd IEEE/ACM Intl Conference on Distributed Computing in Sensor Systems (DCOSS-2006)*, pages 480–497, June 2006.

[11] S. Olariu, M. Eltoweissy, and M. Younis. ANSWER: AutoNomouS netWorked sEnsoR system. *Journal of Parallel and Distributed Computing*, 67(1):111–124, 2007.

[12] T. Rappaport. *Wireless Communications: Principles & Practice*. Prentice-Hall, 1996.

[13] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA-03)*, pages 30–41, May 2003.

[14] L. Song and D. Hatzinakos. Dense wireless sensor networks with mobile sinks. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-05)*, volume 3, pages 677–680, March 2005.

[15] L. Tong, Q. Zhao, and S. Adireddy. Sensor networks with mobile agents. In *Proc. of the IEEE Military Communication Conference (MILCOM-2003)*, October 2003.

[16] Z. Vincze and R. Vida. Multi-hop wireless sensor networks with mobile sink. In *Proc. of the ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT-2005)*, pages 302–303, 2005.

[17] W. Wang, V. Srinivasan, and K.-C. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *Proc. of the International Conference on Mobile Computing and Networking (MobiCom-2005)*, pages 270–283, September 2005.

[18] Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. In *Proc. of the 38th Annual Hawaii International Conference on System Sciences (HICSS-05)*, January 2005.

[19] Q. Zhao and L. Tong. Distributed opportunistic transmission for wireless sensor networks. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-04)*, volume 3, pages 833–836, May 2004.