

# Mobility in Wireless Sensor Networks

Stefano Basagni

ECE Dept.

Northeastern University

Boston, MA 02115

E-mail: [basagni@ece.neu.edu](mailto:basagni@ece.neu.edu)

Alessio Carosi and Chiara Petrioli

Dipartimento di Informatica

Università di Roma “La Sapienza”

Roma, Italy

E-mail: [{carosi,petrioli}@di.uniroma1.it](mailto:{carosi,petrioli}@di.uniroma1.it)

## 1 Wireless Sensor Networks: Generalities and Applications

Technological advances as well as the advent of 4G communications and of pervasive and ubiquitous computing have fostered a renewed interest in multi-hop (ad hoc) communications [1]. In particular, the interest is in self-organizing wireless multi-hop networks composed of a possibly very large number of nodes. These nodes can be either static or mobile, and are usually constrained as for the most critical resources, such as power and computation capabilities.

*Wireless sensor networks* (WSNs) are a typical example of this kind of networks [2,3]. In this case, the well-known paradigm of ad hoc networking specializes to consider the following characteristics.

**Mobility.** Whereas mobility is a fundamental aspect of all nodes in an ad hoc networks, mobility in WSNs pertains mostly to a subset of the network elements, and it is more specifically application dependent.

**Volume of nodes.** WSNs are usually comprised of a higher number of nodes (in the thousands) rather than the few hundreds that are typical of ad hoc networks.

**Resource availability.** Even if an ad hoc node is portable, power and computational resources are not usually crucial elements. Laptops and PDAs have rechargeable batteries, plenty of volatile and durable memory, and run the same software available for bigger, static computers. WSN nodes are heavily resource-constrained. Sensor nodes are usually unreplaceable, and become unusable after failure or energy depletion. Available memory is in the order of KBs, and computational capabilities are limited. This imposes custom operative systems, reduced protocol complexity, and highly specialized software.

**Data communications.** Ad hoc data communication is typically peer-to-peer. One node that wishes to communicate with one or more nodes does so using one of the many available protocols. In WSNs the emphasis is on data transport from the sensor to specific data collection nodes (*sinks*). Nodes are informed about *events of interest* to the users. When such an event is detected by the sensor, a corresponding data packet is created and sent to the sinks possibly via a multi-hop route (data routing).

Given these very specific characteristics of WSNs it is crucial to devise WSNs protocols for topology organization, interest dissemination and data routing that are energy conserving, scalable and able to

prolong the overall network longevity, especially in networks with a large number of devices. Differently from what is commonly understood and found in the WSNs literature, such protocols often have to deal with the mobility of some of the network components. This is the case of applications recently been proposed, which include:

- Underwater monitoring, such as submerging a network of sensors in an ocean bed to detect debris from plane crashes for recovery and identification purposes. In this case, beyond (partially) mobile sensors, the network comprises *unmanned* or *autonomous underwater vehicles* (AUV) that are sent roaming through the network [4] for data collection.
- Networks in support of mobile small-scale robot squads that coordinate for performing a common task. While performing their operations, the robots exchange information among each other and/or transmit collected measures to remote centers by using the network [5,6].
- Sensor networks for collecting information about the location of a user/piece of equipment, etc., like tracking objects as well as humans in a hospital, items in a warehouse, etc. [7]
- Enabling autonomous albeit controlled living of elderly people (independent assisted living) through *e-health systems* made up of wearable sensors as well as tele-medicine equipment, home networking (*domotics*) techniques, and sensor-based *smart spaces* [8].

All these applications involve mobility: Sensor nodes can move (e.g., assisted living through wearable sensors, tracking, etc.) as well as the data collectors (robots or AUVs).

In this chapter we motivate and illustrate the use of mobility in WSNs. Furthermore, we demonstrate how mobility is actually advantageous for improving the performance of these networks.

## 1.1 Motivating the use of mobility in WSNs

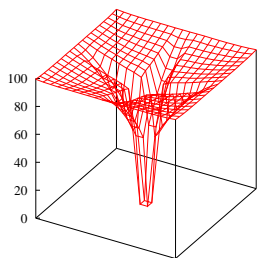


Figure 1: Node residual energy

Most of the research in WSNs concerns networks whose nodes do not move and cannot be replaced. Nodes sense events of interest, and some energy efficient routing protocol is used for delivering the sensed data to static sinks. In this scenario it has been observed that the nodes that more than all the others have their energy drained from data communication are those closer to the sinks. These nodes relay data for all the other nodes in the network as well as packets from the sinks to the sensors. As a consequence, nodes that are closer to the sinks soon “die” from energy depletion resulting in the disconnection of the sinks from the rest of the network. We term the problem of energy drainage at the sink’s neighbors the “sink’s neighbors problem.” The consequences of this problem are illustrated in Figure 1, which shows the average nodal residual energy when the

nodes closest to the sink die. The figure refers to networks with 400 homogeneous nodes, each with a short transmission range (25m), placed on a  $20 \times 20$  grid. The static sink is (optimally) located at the center of the deployment area. Packets are periodically sent to the sink by using a shortest path-like multi-hop routing protocol [9,10]. The picture shows the quite uneven node average residual energy (percentage of the nodal initial energy) at the time the four sensor nodes that relay packets to the sink die, leaving the

sink unable to receive any more data from the network. The remarkably high variance among the residual energies is due to the different distance of each node from the sink and, in general, to the different number of sensor-to-sink routes to which a node belongs, which implies different number of packets to relay. Nodes along the “cross” centered at the sink tend to be the preferred data relays. The closer these nodes are to the sink, the higher the number of packets they receive and transmit, and consequently the higher their energy consumption. These are the nodes with the lowest residual energy in the figure. In particular, when the nodes around the sink die the energy at the nodes along the cross arms averages at 71.07% of their initial energy, while 42.75% of the network nodes have more than 95% of their initial energy available! This incapacity of balancing node energy consumption results in short *network lifetime* (the time until packets are no longer deliverable to the sink) and inefficient use of available resources. The sink is soon disconnected from the network, while a large number of the deployed nodes are still fully operational!

This chapter explores solutions for mitigating the sink’s neighbors problem illustrated above, by exploiting the mobility of some of the network components, thus improving network performance.

Given the typical mobile nature of general ad hoc networks, the impact of mobility on their performance has been extensively explored [11–19]. The primary objective of these works is that of exploiting the mobility of some of the nodes for message delivery in disconnected ad hoc networks, for improving network throughput as well as studying mobility-assisted routing in general.

This chapter concerns approaches designed specifically for WSNs, and in particular those solutions where the mobility of some network elements brings considerable improvement in key performance metrics that are typical of sensor networking. In the rest of the chapter we describe research on mobility in WSNs proposed recently. In particular, we will survey works where a) the *sensor nodes can move*, so that batches of “fresher” nodes are kept close to the sink for providing uninterrupted data forwarding; b) *relays* are sent throughout the network to collect data and bringing them to the sink, and finally we will describe works where the *sink* itself moves to collect sensed data. The last two approaches appear to be the more promising for energy efficiency and longer network lifetime, since sink and relays are usually considered resource-rich. Therefore, energy consumption and network lifetime are not impacted by the energy needed to move them. In the case the sensor nodes move, a great deal of the nodes’ energy is spent on the movement itself, thus having a detrimental impact on the lifetime of the nodes. The final part of the chapter concerns a quite thorough discussion on how to compare different approaches to mobility in WSNs, focusing on sink and on relay mobility. In particular, we show how, by using simulations, one can effectively assess pros and cons of different solutions, and gain useful insights for defining and testing new ones.

## 2 Sensor Nodes Mobility

*Mobility of the sensor nodes* has been exploited for improving, or enabling altogether, sensing and communication coverage [20–22]. The idea presented in [20] is to have the sensors move into positions that minimize the energy cost of reporting streams of data to the sink, which is statically placed. The protocols proposed by Wang et al. in [21] aim at moving mobile sensors from densely deployed areas to areas with coverage holes, where for some reasons, a limited number of sensors has been deployed. The three protocols are proven by simulations to be effective in terms of coverage, deployment time and moving distance. Minimization of the energy consumption of moving nodes has been subsequently addressed by the authors in [22] by letting the node “move logically,” i.e., only after they can decide whether their moving maximizes the coverage or not. The idea of using mobile sensors has also been explored within the robotic community, where mobile robots are (also) equipped with sensors. Typical examples are the

works by Howard et al. [23,24]. In [23] an algorithm for the deployment of the members of a robotic team into an unknown environment is given. Aim of this algorithm is the maximization of the coverage area, while maintaining line-of-sight contact among the robots. In [24] the same authors draw from the theory of potential fields to distribute the mobile sensors throughout a given area. The fields are constructed in such a way that each node is repelled by obstacles and other nodes, thereby forcing the node to spread throughout the area.

Finally, distributed algorithms for the mobility of sensor nodes have been investigated in [25]. In this work mobility algorithms are proposed that move the nodes to positions that reduce the transmission power needed to send the data to the (static) sink. The positions for the moving sensors are determined via distributed simulated annealing, as opposed to a greedy strategy that could lead to a sub-optimal placement. By using distributed simulated annealing, a node, based only on information on its current neighbors accepts a “bad move” with a positive probability. This move could be locally non-optimal, but could benefit the network globally in the longer run.

Works that consider mobile sensors and robots are mostly concerned with sensor deployment time and with sensing coverage. The costs associated with sensor movements as well as the cost of transmitting sensed data are often not considered, and network lifetime is rarely a metric of interest.

### 3 Mobile Relays

The work by Chatzigiannakis et al. [26] explores the possibility of using the coordinated motion of a small number of users in the network to achieve efficient communication between any pair of other mobile nodes. A fraction of the network nodes acts as *mobile relays* carrying packets for other nodes: The packet is exchanged when the source node and the agent are neighbors (i.e., in the radio vicinity of each other), and it is then delivered to the intended destination when the agent passes by it.

This basic idea has been introduced to WSNs by Shah et al. in their works on *data MULEs* [15,27]. Mobile nodes in the sensor field, called MULEs, are used as forwarding agents. The idea here is to save energy by having single-hop communication (from a sensor to the MULE that is passing by) instead of the more expensive multi-hop routing (from the sensor to the sink): It is the MULE that will eventually take the sensed data to the sink. The data MULE architecture is effective for energy conservation in delay tolerant networks [28]. Energy is traded off for latency, i.e., the energy needed to communicate a packet to the sink is decreased at the cost of waiting for a MULE to pass nearby (and at the cost of waiting for the MULE to move to the vicinity of a sink). Scheduling problems in sensor-to-sink transmissions within this model have been studied in [29].

The idea of using unmanned vehicles as data collectors has been further investigated in [30]. The sensor nodes send their data to nearby clusterheads via multi-hop routing. The vehicles then pass by the clusterheads to collect the data. Three different solutions are presented in the paper that define different schedules for the collectors to visit the clusterheads. In this way, the collector has to visit only the clusterheads, and not all the nodes, while multi-hop routing is reduced to a smaller number of hops since the data is sent from a sensor node to a clusterhead who is nearby. In this case the sink (base station) is not really mobile, and the collectors return to the base station periodically to deliver the data and for recharging. The architecture of [30] has been expanded in [31] to consider different classes of nodes, where the collectors roam (controllably) among the clustered sensor nodes some of which can be (uncontrollably) mobile. Aim of the paper is to determine schedules for the collectors to visit the nodes so that transmission energy, data latency and buffer requirements at the nodes are minimized.

A first in-depth discussion on how to incorporate controllable mobile *relays* into the network infrastructure has been presented in [32]. The authors describe an implementation of a sensor network with an autonomous mobile relay (a robot) that visits the (static) sensors, collects their data, and delivers them to the sink. The idea of collecting data in a single-hop fashion (i.e., when the robot approaches a sensor) is similar to that of data MULEs. However, in this case the movements of the robot adapt to data collection performance parameters, which are dictated by the network application priorities. The robot is part of the system, and it is the system that *controls* its mobility. The testbed-based experimental results in this paper concern the evaluation of methods for controlling the speed of the robot for optimizing data collection. The robot traverses networks with different densities following a straight trail and collects the data that are then brought to the sink. In case the robot does not pass sufficiently close to a sensor node, methods are defined for routing the sensed data to nodes that are one hop from the robot route. Further development of this work with multiple controlled mobile elements (here called explicitly data MULEs) has been presented in [33]. This work considers the two cases where nodes are deployed uniformly and randomly in a given geographic area and when, more realistically, they are distributed differently. In the first case, criteria are given for the choice of the number of MULEs and for dealing with nodes that can be served by multiple mules. In case of non-uniform nodal distribution, a load balancing algorithm is introduced for distributing the number of sensor nodes to the various MULEs so that each MULE serves approximately the same number of sensors. MULEs roam through the network in straight lines and gather information about the nodes they can reach. Then the MULEs, that can wirelessly talk to each other directly, elect a leader and send the information they gather to it. Based on this information the leader executes the load balancing algorithm and associates nodes to MULEs. Data collection is finally performed by the MULEs that travel through the network (in straight lines) and explicitly poll the assigned sensor nodes for collecting their data. The problem of scheduling the visit to the sensor nodes of a single relay (called ME, for Mobile Element) so that there is no data loss (due to buffer overflow) is tackled with in [34]. The corresponding Mobile Element Scheduling (MES) problem is proven to be NP-Complete and a centralized analytical model (ILP) and algorithms are given for solving the problem. In particular, given as input the data generation patterns (sensing rates) at the sensors, the corresponding buffer overflow times and a matrix cost for the ME movements, model and algorithms determine the sequence of visit to the nodes so that none of the buffers overflow. General discussions on the advantages and challenges of controlled mobility are conducted in [35, 36]. In the first paper a new approach to exploiting mobility, termed *Morph* is defined, and the benefits to sensor networks sustainability are shown and discussed. In [36] considerations about the use of controllable mobility are more general, and are followed by a detailed description of how mobility induces improvements in WSNs performance.

The problem of controlled data MULE-like mobility has been also recently addressed in [37]. The authors first propose an algorithm for avoiding sensor nodes buffer overflow while minimizing the speed of the mobile relay. They also extend this algorithm to the case where some of the packets have delay constraints (i.e., they are “urgent messages” that have to be delivered to the sink within a given time since their generation). Finally, an investigation of the controlled use of relay nodes for data collection and subsequent report to the sink has also been proposed in [38]. Although the authors recognize that moving the sink directly yields better resource utilization and hence longer lifetime, they argue that for certain applications moving the sink is infeasible. Therefore, having one or more resource-rich mobile relay nodes is remarkably helpful. Given that the sensor nodes know about the current location of the relay node, routing protocols are presented for delivering the data from the sensors to the relay, from the relay to the sink, and finally for determining the route of the relay. Improvements on network lifetimes are fourfold with respect to the case of a static sink.

## 4 Mobile Sinks

A data dissemination protocol, termed SEAD (Scalable Energy-Efficient Asynchronous Dissemination) is defined by Kim et al. in [39] where a tree-like communication structure is built and maintained. Differently from previous data centric solutions à la *Directed Diffusion* [40], that also use tree structure for data and interest gathering and dissemination, according to SEAD the *sink moves* (randomly) to certain sensor nodes in the tree (access points) from which it collects the data. Communication between the sink and the access points can be multi-hop. This happens when the sink moves away from the access points. SEAD is designed for finding a trade off between the data delay to the sink and the energy consumption spent for reconfiguring the tree so that the access points are closer to the current position of the sinks. SEAD is shown to be more effective for conserving energy than other solutions for data dissemination in wireless sensor networks where the sink does not move, such as directed diffusion [40], TTDD [41] and ADMR [42].

The problem of building and maintaining routes to a mobile sink with the precise aim of minimizing the corresponding overhead has been tackled with in [43,44] and [45]. In the first paper, local update techniques are described for detecting disconnections and perform route repair in “sink oriented trees.” In [44], the ERUP protocol is proposed for conducting route re-discovery only in the vicinity of the damaged route. In [45], initial routes are constructed from the nodes to the sink according to any viable WSN routing. If, because of the movement of the sink, the routes are no longer valid, forwarder nodes are designated to extend the current routes.

Common to all these works is that the mobility of the sink is unpredictable and *uncontrollable*. For example, in [39] and [46] sinks move according to the random waypoint model.

The use of mobile sinks with *predictable* mobility has been more recently presented in [47–49] and [50]. In these works the sinks (airplanes) fly over the sensor field and gather the sensed data periodically. While the movement of the sink is fully controllable, it is external to the network infrastructure, i.e., the trajectories are not determined by network components and activity. The main contribution of these papers concerns the energy-efficient transmission to the passing sink [47,48,51]. In [49] the authors consider heterogeneous sensor networks made up of two types of nodes. Type 0 nodes do the basic sensing, perform short range communications and are partitioned into clusters whose clusterhead is a type 1 node. The clusterheads take care of receding data from type 0 nodes (possibly through multi-hop routes), do some sensing, aggregate data and perform long range transmissions to the aircraft. Each passing of the aircraft triggers a new data sensing and data collection cycle. The aim of the paper is to determine the optimum node deployment (i.e., the densities of each type of nodes) and the nodal energy needed to achieve a given network lifetime (numbers of data gathering cycles), while ensuring sensing coverage and radio connectivity with high probability.

Inherent patterns of the sink movement are exploited in [52] for the design of robust and energy-efficient routing. This work assumes that there is a certain degree of predictability in the sink movement, such as the routine route of a ranger patrolling a forest. Sensor nodes learn about the sink whereabouts at given times via statistics techniques as well as methods from distributed reinforcement learning.

A model for sink movement is proposed in [50], where “observers” (i.e., the sinks) move along the same route repeatedly. The sensed data are collected while the observer traverses the network. When passing by sensor nodes, the observer wakes them up and receives their data (if any). The authors describe a prototype system developed at Rice University where the observers are carried by campus shuttles, and the sensors are spread out throughout university premises. In particular, the authors determine the transmission range needed to collect data from a predefined percentage of the sensor nodes, given the observer speed, the time required to transmit a piece of information, and the traffic pattern. The correlation among the various

system parameters is investigated analytically.

*Network-controlled sink mobility* for reducing energy consumption and for maximizing the lifetime of a sensor network has been considered initially in [53] and then in [54], [55], [56] and [9,10]. In these works the sink moves among the (static) sensor nodes and, while sojourning at given locations, collects data that are sent to it via multi-hop routes. The first work is mostly concerned with energy minimization. The authors present an ILP model to determine the locations of multiple sinks and the routes from the sensors to the sinks. Time is divided into rounds. At the beginning of each round information on the nodes' residual energy is centrally gathered and the ILP problem is solved to determine new, feasible locations the sinks should travel to for minimizing the maximum energy consumption spent at the nodes during that round. Minimizing the energy consumption results in increased network longevity. No constraints are enforced on the sink movements, and there is no relation between the number of the sinks and their position in subsequent rounds.

The problem of network lifetime maximization through controlled sink mobility is explicitly addressed in [54] for networks with a single sink. Sink locations (in this case the sensor sites) and sink sojourn times at those locations are determined that maximize the network lifetime via a new LP formulation of the problem: Maximizing the network lifetime equals maximizing the sum of sojourn times of the sink at the visited locations. Although the model is completely general, the experiments performed in the paper refer to scenarios where  $n = L^2$  nodes are arranged in a  $L \times L$  grid. The sink has no limitation on the time  $t_k \geq 0$  it can spend at sensor  $k$  and can move from any location to any location in the network. Improvements on network longevity are obtained that are almost five-fold when the sink sojourns at the nodes located at the four corner areas and in the central area of the grid.

By combining the model presented in [54] and the LP formulation for maximum lifetime routing described in [57], Papadimitriou and Georgiadis [55] present another centralized solution for the problem of maximizing network lifetime. By turning a constant of the model in [54] into a variable, the model presented in [55] jointly solves the problem of determining the sink sojourn times at the given sites, and the routing of the packets to the current position of the sink. This (data) routing-dependent solution achieves improvements with respect to the lifetime values of [54] that are twofold.

The problem of lifetime maximization has been formulated as a min-max problem by Luo and Hubaux [56]. By considering together sink mobility and data routing, a load balancing solution is obtained that, while keeping the sink moving along the external perimeter of the network, achieves lifetimes 500% higher than when the sink stays in the center of the network.

These centralized solutions concern how to drive the sink to places in the network where data collection has the least impact on the sensor nodes' residual energy. However, several aspects of sensor networking (such as the possibility of collision and corresponding energy cost) are not considered. Similarly, the cost of building, maintaining and releasing routes to the current position of the sink and its impact on network lifetime and other metrics is never taken into account. A Mixed Integer Linear Programming (MILP) model has been presented in [58,59] and in more details in [9,10] where more realistic data communication costs and constraints of sensor networking and on sink mobility are explicitly considered. Moreover, those papers introduce and evaluate the first distributed and localized heuristic for controlled sink mobility. The MILP model and the distributed solution are described thoroughly in the next section.

## 5 Comparing Different Approaches to Mobility

The most promising use of mobility for enhancing the performance of WSNs has been shown to be obtained when resource-rich network components move. In other words, when either the sink itself, or mobile relays are left free to roam through the network for collecting data.

This part of the chapter is dedicated to describe and discuss how to compare two very different approaches to mobility in WSNs. Our aim is twofold. First of all, we want to gain an understanding of which, among the many proposed solutions, are the most promising ones in terms of usage and performance enhancement. More than specific mobility protocols, we are interested in determining which network architecture is the best for supporting efficiently the application listed above. Then, we want to establish clear metrics and tools for comparing mobility solutions and obtain from this performance comparison new insights for protocol design and effective use of mobility.

By the study of what proposed in the literature, we could observe that when considering mobility of sinks and relays, the major difference in performance is made by whether routing between the sensor nodes and a sink/relay is single-hop or multi-hop. This choice for the routing affects greatly essential metrics such as nodal energy consumption, packet latency, and probability of successfully delivering data to the sink/relay. It has also immediate consequences on the design, and cost, of the wireless sensor node: When a multi-hop data routing protocol needs to be implemented, a node has to provide resources (storage, computation, etc.) for enabling the forwarding of packets to the data collector, i.e., a protocol stack has to be available. When routing is single-hop, instead, a simpler nodal architecture suffices for storing the packets and for transmitting them to a relay when it passes by.

In the following we describe and investigate two network architectures that clearly show the impact of choosing multi-hop vs. single-hop routing when sinks or relays roam through the network. We consider single-hop routing according to the data MULE approach. We will both consider the original model [27], which exploit the *uncontrolled* mobility of multiple relays, and more recent works where it is shown that the relay mobility can be controlled by the network current conditions [33]. We then show the pros and cons of multi-hop routing by describing an analytical model and distributed heuristics for sink mobility. As in the previous case, we consider and compare solutions where the mobility of the sink is controlled by the network itself with those where sink mobility does not depend on what is going on in the network [9, 10].

### 5.1 The data MULE approach to mobility in WSNs

The idea of a *Mobile Ubiquitous LAN Extension* (MULE) [15,27] stems from that of deploying mobile agents for carrying information [60], especially among nodes in possibly disconnected networks. The concept is quite simple: MULEs are resource-rich nodes that roam freely throughout the network. When as a result of its motion a MULE gets to be in the radio proximity of a wireless sensor node, it receives all the packets generated by that sensor so far (if any). Upon getting close to one of the network *access points* (i.e., the sinks), the MULE transfers to it all the collected packets. In this way, whether there is a route from a sensor to one of the sinks or not, i.e., independently of network connectivity, a packet is eventually delivered to the sink. The key concept here is that sensor-relay routing as well as relay-sink routing is single-hop.

The data MULE architecture is made up of three layers. The highest layer comprises one or more access points, i.e., the sinks. These are the resource-rich, static components to whom the MULEs deliver the data collected throughout the network. The middle layers is the set of the MULEs. These are nodes characterized by mobility, large storage as well as renewable power and by the ability of communicating with the sensor nodes as well as with the sinks wirelessly. Most importantly, MULEs movements cannot



be predicted in advance, and are considered completely random (uncontrolled mobility). Examples of MULE nodes are animals roaming in an given area (as in an environmental monitoring application [11]), or vehicles traveling in a city or university campus. Finally, the bottom tier is comprised of a possibly large number or resource-constrained static sensor nodes that can communicate with a MULE that is passing by.

Communication in the MULE architecture is single-hop: A node stores the sensed data until a MULE passes by. Once the MULE arrives, the node transmits all the stored packets to the MULE. When the MULE passes by the sink it transmits to it all the packets it has collected.

Clear advantages of the MULEs approach to data gathering and delivery in wireless sensor networking include the following.

- **Robustness.** No node depends on any single MULE. Any of the roaming MULEs would do. If one of the MULEs fails, a node can still count on another MULE to pick up its packets.
- **Lower complexity.** The one-hop nature of the sensor-to-MULE communication allows very simple nodal protocol stack. Unsophisticated MAC is often enough to guarantee safe transfer of data packets from their source to the passing MULE.
- **Decreased energy consumption.** The typical overhead associated to multi-hop ad hoc routing is not present in the MULE system. This overhead is oftentimes the major culprit of energy consumption in WSNs.

However, the flip side of this quite simple and effective system presents non-negligible drawbacks. The first that comes to mind is the average latency incurred by the packets from the time they are created to when they are delivered to the sink. Given the serendipitous nature of the MULEs movements, there is no guarantee on deterministic delay bounds. It is not possible to surely state when and where a MULE will show up close to a sensor or to the sink. Therefore, a packet can sit in a sensor queue, or in the queue of a MULE for quite a long time, which imposes very high overall sensor-to-sink latency. The uncontrolled MULE mobility has also a detrimental effect on the packet delivery ratio, i.e., the ratio between all packets that are created in the network over those that are successfully delivered to the sink. Sensor nodes, having limited storage, can buffer only a limited amount of packets. If the inter-arrival time of a MULE in the radio vicinity of a sensor is too large, chances are that the sensor is forced to discard all the packets in excess of its buffer size. This forces down the amount and type of applications that could use the MULE architecture for data gathering and dissemination. Only applications that are delay tolerant can endure the possibly very high latencies imposed by the MULE system.

Recent studies have shown that it is possible to mitigate the drawbacks of the MULE architecture by having the relay movement guided by the network itself, i.e., by *controlling* the mobility of the MULEs. One of the first attempts is described by Kansal et al. in [32]. As mentioned earlier, this work has quite an experimental flavor and is concerned with a network where one MULE (a mobile robot) travels along a straight trail up and down the sensor node deployment area to collect data. Variation in the speed of the MULE are determined by the network load: The robot will go slower if more data are to be transmitted by the nearby sensors, faster otherwise. Since the MULE's route is always the same straight line it is highly likely that some sensors are not able to transmit their data to the MULE directly. For this reason the solution proposed in [32] organizes the network operation into two parts. The first, named the *network algorithm*, takes care of how the sensor nodes interact with each other so that packets from those nodes that are not sufficiently close to the passing MULE are delivered to it by nodes that are close. The second

part specifies how the MULE moves, and it is called the *motion control algorithm*. Here is a description of the two parts.

The **Network algorithm** defines the way in which those nodes that cannot directly transmit to the sink find data routes to send their packets via intermediate nodes. The network algorithm is performed in three steps: *Initialization*, *local multi-hop* and *data collection*.

1. **Initialization.** During the first run through the network deployment area, along its only route, the MULE beacons a simple “hello”-like packet. All nodes that receive this packet are neighbors to the passing MULE, set a hop-count variable to 1, and broadcast the hello packet further, updating the hop-count field to its own. A node that receives multiple hello messages from neighboring nodes updates its own hop-count variable to the least values received plus 1, and keep forwarding the message (with the updated hop-count). At the same time, it records the node closer to the MULE path that sent the hello packet with the least values (possible ties are easily broken). In this way each node become aware of a) its hop distance from a node that can directly transmit to the passing MULE, and b) a route to that node. Overall, a forest (set of disjoint trees) is built as a first step of the network algorithm, with the nodes closer to the MULE’s route as the roots.
2. **Local multi-hops.** Data packets are sent at the root nodes from any sensor node in the corresponding tree. Any suitable routing algorithm (e.g., *directed diffusion* [40] or a simpler “convergecasting” as the one defined in [61]) can be used for implementing this step.
3. **Data collection** by the MULE. After its first passage through the network, the MULE keeps moving up and down its route collecting data from the nodes along its path. Nodes polled by the passing MULE transmit their packets along with those of the nodes in their trees.

**Motion control algorithm.** Since the route of the MULE is fixed, motion control reduces to speed regulation. This is done by fixing the round trip time (RTT) of the MULE. Then, three possible strategies are described in [32] that correspond to different strategies of MULE movements.

1. The path is traversed at a certain fixed speed. Let us assume that at this speed the MULE takes  $T < \text{RTT}$  to travel its route (and back). The remaining “spare time”  $\text{RTT}-T$  is divided equally among the nodes along the MULE’s route, and the MULE stops at each node for this time. The MULE might not know about the node location. Therefore, the MULE stops as soon as it hears from the node.
2. The MULE travels the route at the constant speed of  $(\text{length of route})/\text{RTT}$  and does not stop at the nodes.
3. The speed of the MULE adapts to the current situation of (the queue of) the nodes. This can be realized by having the MULE traveling at the speed  $2*(\text{length of route})/\text{RTT}$ . The extra amount of time is divided among those nodes from which the MULE has collected a limited amount of data (determined by a given threshold) in the previous round. (The set of nodes the MULE stops at changes at each round.)

Further details about the described algorithms as well as its implementation on a real-hardware testbed can be found in [32].

Here we describe the natural evolution of this *single*-MULE solution to controlled mobility in the “mostly single-hop” routing realm: *Multiple* controlled mobile elements (data MULEs) for data collection [33]. The motivations are clear: The single MULE approach does not scale well. If the number of nodes in the network increases, there are more nodes from which data must be collected in the same amount of time. Therefore, packets might be dropped (buffer overflow) because the MULE cannot get to visit certain nodes on time. Deploying multiple MULEs solves this scalability problem. If the nodes are scattered randomly and uniformly throughout the deployment area, the obvious solution consists in dividing the area into same-size parts and having a MULE in each of these parts. Given the deployment distribution, each MULE would have roughly the same amount of nodes to serve. In this case, two are the main problems to deal with: Deciding the number of data MULEs and deciding what to do with nodes that are shared by two MULEs.

- **Number of data MULEs.** The number of data MULEs needed to collect data while avoiding packet loss is computed based on the RTT of a MULE (defined as the time it takes to the MULE for traversing the assigned area along a straight line) and on the *buffer\_fill\_time*, which is the time needed to a node to fill its buffer. The RTT is defined as follows:

$$(l/s) + n \times st + (l/s),$$

where  $l$  is the side of the deployment area (a square),  $s$  is the MULE speed,  $n$  is the number of nodes in the network and  $st$  is the service time, i.e., the time needed to transfer packets from a node’s buffer to the servicing data MULE. It is assumed that the MULE stops at each node while traversing the networks for  $st$  time. This happens only one way. Once the MULE has reached the end of the line, it goes back without collecting data. In case there are nodes that are not in the transmission range of any of the MULEs, multi-hop routing is used to send the packet to a node (root of the data distribution tree, as in the single-MULE case described above) close to the MULE route. In order not to loose packets because of buffer overflow the number of data MULEs required will be:

$$\lceil \frac{\text{RTT}}{\text{buffer\_fill\_time}} \rceil.$$

(One MULE suffices when  $\text{RTT} \leq \text{buffer\_fill\_time}$ .)

- **Sharing nodes.** Whenever a node can be serviced by two MULEs, most likely because it is equidistant (in hops) from the MULEs routes, ties are broken randomly or by simple techniques. For instance, the node might decide to send one packet to one MULE and the next one to the other MULE, or it may just flip a coin every time it has a packet to send, and decide which MULE is going to get the packet accordingly.

Deployment of multiple MULEs is also described in case nodes are not scattered randomly and uniformly. This can happen, for instance, when nodes are manually placed in specific places, or when the terrain induces a non-uniform nodal distribution. The MULEs move along straight lines that are not necessarily equally spaced throughout the area. Assuming that each node is at most one hop from a MULE’s route (i.e., that every sensor is directly served by at least a MULE), nodes are divided into two classes: *Non-Shareable (NS) nodes* and *Shareable (SH) nodes*. The first set comprises those nodes that can be served only by one of the MULEs. The set SH instead contains nodes that are in the transmission range of multiple MULEs (at least two). For NS nodes there is no other option than to be served by the only

MULE that passes close to them. Shareable nodes should instead being assigned to only one MULE. The challenge here is to define a way to assign nodes in SH to the MULEs so that the number of nodes served by each MULE is roughly the same. The algorithm for data gathering in this case is made up of five phases: *Initialization*, *leader election*, *load balancing*, *node assignment* and *data collection*. The following is a brief description of the algorithm (details can be found in [33]).

- **Initialization.** The MULEs go on their first trip beaconing around their presence. Nodes that receive this beacon reply back with their ID. By the end of this phase the MULEs are back at their starting points and know of all the nodes they can serve.
- **Leader election.** All the MULEs are assumed to be able to communicate with each other (i.e., at least when at their initial position, the topology of the “MULE network” is a clique). One of the MULEs is elected as the leader of the bunch (for instance, the MULE with the smaller ID). At this point, every MULE sends to the leader the information gathered in the first phase.
- **Load balancing.** The leader divides the network nodes into shareable and non-shareable. The shareable nodes are further classified according to the MULEs that can serve them. This is the most delicate part of the algorithm, all performed locally by the leader MULE. It is divided into two main parts: The first part takes care of calculating the average number of nodes that could be assigned to each MULE. The second takes care of the actual load balancing [33], determining the nodes that could be assigned to each MULE.
- **Node assignment.** As a result of the previous phase some of the shareable nodes can be served by two MULEs. In the node assignment phase nodes that are shared by two MULEs are ordered according to their unique ID. At this point the first part of the set is assigned to the MULE of the two with the smallest ID, and the remaining nodes are assigned to the other MULE. After the assignment has been calculated the leader MULE communicates to all the other MULEs the nodes they have been given to serve.
- In the **data collection** phase the MULEs start their journey along their routes, looking for data. At this time the nodes do not know the MULE they are assigned too, and hence when they hear the MULE beacon they respond with their packet. They will receive an acknowledgment only from the MULE that is supposed to collect their packets. A node records this information and in the future responds only to its assigned MULE.

Simulation results are shown in [33] that demonstrate how effective this algorithm is in distributing nodes to MULEs.

## 5.2 Multi-hop routing and controlled sink mobility

We consider here the case of a mobile sink traveling through the network and receiving data at its current location via multi-hop routing. We start by describing in details a solution where the sink moves according to the current network condition (controlled mobility), with the aim of balancing energy consumption among the network nodes and thus prolong network lifetime.

This solution, presented in [9, 10], illustrates two fundamental research steps: Mathematical modeling (for the centralized determination of an optimal solution) and the definition of distributed heuristics. A

mathematical model is defined that taking as input realistic parameters such as the cost of route set up, maintenance and release, the cost of transmitting and relying packets to the sink and limitations in sink movement, determines where the sink should go and how much it should stay there so that network lifetime is maximized. The solution, clearly centralized, is based on a Mixed Integer Linear Programming (MILP) modeling. Despite this kind of models are computationally hard to solve, a mathematical formulation is defined that without losing in expressiveness is solvable for quite non-trivial and realistically large instances of the problem. The optimum sink route determined by solving the MILP model is used for comparison with the route determined by a distributed heuristic which is more suited for use in WSNs. The distributed heuristic, termed Greedy Maximum Residual Energy (GMRE), determines sink movements and sojourn times by greedily sending the sink from one point to the one surrounded by the nodes with the highest residual energy.

Here is the typical scenario. A large number  $|N|$  of resource constrained, static nodes with sensing and wireless communication capabilities are scattered in a given geographic area. Data packets are generated periodically at the sensor nodes: Node  $i \in N$  transmits at a given data rate  $r_i$  packets that are delivered to the sink for processing.

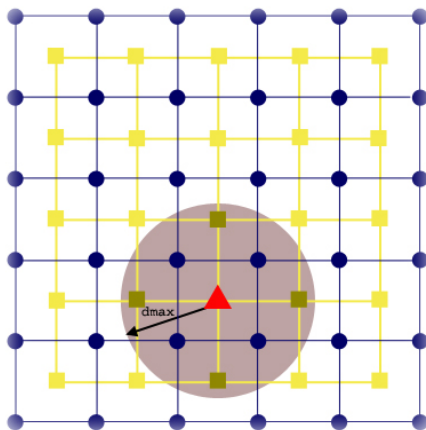


Figure 2: Sensor nodes, sink sites and sink movements

While the nodes are static, the sink can be mobile. More specifically, a set  $S = \{1, \dots, q\}$  of  $q$  sink's sites is considered which are the points within the geographic area the sink can visit. For instance, Figure 2 shows a typical scenario where 36 nodes (represented by circles) are placed on a  $6 \times 6$  grid and 25 sink's sites (squares) are organized according to a  $5 \times 5$  grid. A link between two nodes indicates that those two nodes are neighbors (i.e., they can hear each other's transmissions). A lighter-colored link between two sites indicates that the sink can move from one site to the other and vice-versa. Because of the sink's neighbors problem the sink moves throughout the network in an attempt to balance the energy consumption among the nodes. Every time the sink reaches a new site, it floods a packet  $f$  to all the network nodes making them aware of its current site. A node that receives  $f$  starts sending/relaying its packets toward the new site of the sink. Every routing scheme that works with the topological information provided by  $f$ , such as geographic or shortest paths-based routing, is a viable routing for data delivery to the sink. The independence from the particular routing protocol yields a twofold advantage. First of all, it guarantees the longest possible network lifetime given the specific routing. Furthermore, it allows the network users to design or choose the routing algorithm that best meets the WSN application requirements in terms of

a host of different metrics of interest (not just the lifetime). Every time the sink leaves a site, it again floods a packet to all nodes to communicate that it is no longer reachable at that site. Upon receiving this second packet, a node stops forwarding data (remaining packets are buffered), and waits to receive a new packet  $f$  from the sink, carrying its whereabouts.

There is virtually no bound on how far the sink can travel between two sites. However, while the sink is traveling, the sensors do not transmit. Therefore, if new data are sensed, these are buffered. This implies the possibility of high delays for data packets. In order to contain this delay, a new parameter  $d_{MAX}$  is introduced which represents an upper bound on the distance that the sink can travel from a site to the following one. Thus, the pair  $(S, d_{MAX})$  uniquely defines a graph of sink's sites where there is a link between two sites if and only if their (Euclidean) distance is  $\leq d_{MAX}$ . Figure 2 shows the four sites (darker squares) the sink (the triangle) can reach from its current position. The lighter lines between the sites of Figure 2 indicate that the sink can only move horizontally or vertically in the  $5 \times 5$  grid.

In case of high sink mobility and low data traffic the energy cost for route construction and release can be significant. Therefore, this cost should be explicitly taken into account. In order to evaluate the impact of different (higher or lower) sink mobility rates, the parameter  $t_{min}$  is introduced to represent a mandatory minimum time the sink has to sojourn at a site.

Here is the problem that is solved via mathematical modeling and by designing distributed protocols: *Determine the starting site and the route for the mobile sink over the graph  $(S, d_{MAX})$ , together with the sojourn times  $t_k \geq t_{min}$  of the sink at each visited site  $k \in S$  so that network lifetime is maximized.*

### 5.2.1 MILP formulation

Here are the sets, the parameters and the variables used for formalizing the problem.

- Let  $S$  be the set of sink's sites, i.e., the locations at which the sink may sojourn:  $S = \{1, \dots, q\}$ .
- $N$  is the set of the network nodes:  $N = \{1, \dots, n\}$ .
- $e_0$ : Initial energy (Joules) of each node.
- $f_{ik}$ : Energy consumption (Joules) at node  $i \in N$  for setting up/releasing routes when the sink moves to site  $k \in S$ .
- $c_{ik}$ : Power consumption (Watts) for receiving and transmitting packets at node  $i \in N$  when the sink sojourns at site  $k \in S$ .
- $t_{min}$ : Mandatory minimum time (secs) for which the sink is required to stay at site  $k \in S$ .
- $d_{jk}$ : Euclidean or shortest path distance (meters) between any two sink sites  $j, k \in S$ .
- $d_{MAX}$ : Maximum distance (meters) the sink is allowed to travel each time it moves.
- $A$ : The set of directed edges joining sink sites whose distance is less than or equal to  $d_{MAX}$ , i.e.,  $A = \{(j, k) \in S \times S : j \neq k, d_{jk} \leq d_{MAX}\}$ .
- $O$ : The set of directed edges  $(0, k)$ ,  $k \in S$ , joining a fictitious site 0 (origin) with the sites in  $S$ .
- $D$ : The set of directed edges  $(k, q + 1)$ ,  $k \in S$ , joining the sites in  $S$  with a fictitious site  $q + 1$  (final destination).

- $X$ : The union of  $A$ ,  $O$  and  $D$ .

The following variables are also considered:

- $t_k$ : Sojourning time (secs) of the sink at site  $k \in S$ .
- $y_k$ : Binary variable taking the value 1 if the sink sojourns at site  $k \in S$  ( $t_k > 0$ ); 0 otherwise ( $t_k = 0$ ).
- $x_{jk}$ : Binary variable indicating the status of  $(j, k) \in X$ .  $x_{jk} = 1$  if and only if arc  $(j, k)$  is on the sink movement route;  $x_{jk} = 0$  otherwise.
- $u_k$ : Auxiliary variable used to enforce a unique sink route.

The MILP formulation is defined by the following objective function and constraints.

$$\text{Max } \sum_{k \in S} t_k \quad (1)$$

$$\text{subject to: } \sum_{k \in S} c_{ik} t_k + \sum_{k \in S} f_{ik} y_k \leq e_0 \quad (i \in N) \quad (2)$$

$$t_{\min} y_k \leq t_k \leq M y_k \quad (k \in S) \quad (3)$$

$$y_k \in \{0, 1\} \quad (k \in S) \quad (4)$$

$$\sum_{k \in S} x_{0k} = 1 \quad (5)$$

$$\sum_{k \in S} x_{k,q+1} = 1 \quad (6)$$

$$\sum_{\substack{j \in S \cup \{0\} \\ (j,k) \in O \cup A}} x_{jk} = \sum_{\substack{j \in S \cup \{q+1\} \\ (k,j) \in A \cup D}} x_{kj} \quad (k \in S) \quad (7)$$

$$\sum_{\substack{j \in S \cup \{0\} \\ (j,k) \in O \cup A}} x_{jk} = y_k \quad (k \in S) \quad (8)$$

$$u_j - u_k + q x_{jk} \leq q - 1 \quad ((j, k) \in A) \quad (9)$$

$$u_k \geq 0 \quad (k \in S) \quad (10)$$

$$x_{jk} \in \{0, 1\} \quad ((j, k) \in X) \quad (11)$$

The objective function (1) maximizes the sink's total time at sojourning sites,  $\sum_k t_k$ , which is the effective network lifetime. Constraint (2) states that the combined energy spent at node  $i$  for data delivery ( $\sum_{k \in S} c_{ik} t_k$ ) and data route construction and release ( $\sum_{k \in S} f_{ik} y_k$ ) during  $\sum_k t_k$  (the time before the death of the first node) should not exceed the node's initial energy  $e_0$ . The right part of double inequality (3) forces  $y_k$  to take the value 1 if the sink sojourns at site  $k$  ( $t_k > 0$ ), thus linking the binary variable  $y_k$  (constraint (4)) with the continuous variable  $t_k$ .  $M$  is a significantly large number. The left part of double inequality (3) restricts the sojourn time  $t_k$  to be at least equal to the mandatory minimum sojourn time  $t_{\min}$  if the sink sojourns at site  $k$  ( $y_k = 1$ ) and at the same time forces  $y_k$  to take the value 0 if the sink does not sojourn at site  $k$  ( $t_k = 0$ ). The first sojourning site in the sink's movement route is allowed to be any site in  $S$ . To implement this, a fictitious fixed initial site 0 (origin) is introduced. At the beginning of the sensor network's lifetime, the sink moves in zero time (and cost) from the origin to some site  $\alpha \in S$ , determined by the model. This is that particular site such that  $x_{0\alpha} = 1$  (equation (5)), namely, it is the

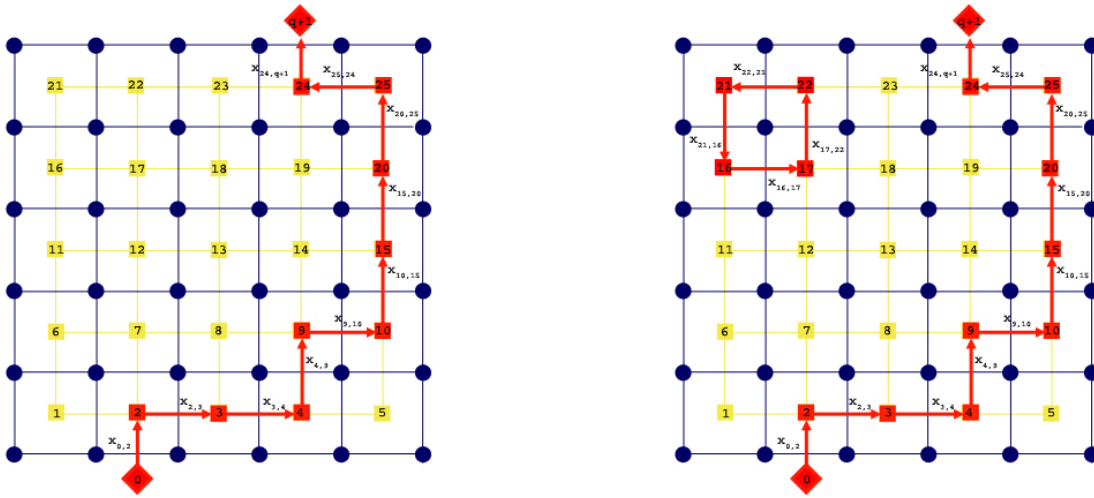
optimum starting point of the sink journey. Then, the sink sojourns at that first site and at subsequent other sites in  $S$  to be determined by the model. Finally, from the last sojourning site  $\omega$  the sink moves to a second fictitious site “q+1” (destination), again in zero time (equation (6)). The site  $\omega$  completes the sink route started at site  $\alpha$ . This is the last site at which the sink sojourns, and marks the end of the sensor network lifetime. The arcs  $(j, k) \in X$  on the sink route are associated with binary variables  $x_{jk}$  equal to 1. The variable  $x_{jk}$  is equal to 0 for all the  $(j, k) \in X$  that do not belong to the route. Equivalently, one can think of a unit of flow moving from the origin to the destination. Constraint (5) induces a unit of flow from the origin to some node  $\alpha \in S$ , while constraint (6) causes the destination to absorb a unit of flow coming from some node  $k \in S$ . Constraint (7) forces flow conservation at all sites  $k \in S$ , thus ensuring the generation of a route. Constraint (8) ensures that the sites  $k \in S$  on the generated route are sites at which the sink sojourns ( $k|y_k = 1$ ). To elaborate, if  $y_k$  in constraint (8) equals 1, then the sink sojourns at site  $k$ , and therefore there must be one and only one arc on the sink’s movement route reaching site  $k$ . On the other hand, if  $y_k$  equals 0, then there will not be any incoming arc to that site. Finally, constraint (6) induces a unit of flow from the last node in the sink route ( $\omega$ ) to the fictitious final node  $q + 1$ .

Figure 3(a) shows a possible optimum sink route that goes from the initial site  $\alpha = 2$  to the final site  $\omega = 24$ . Constraints (5) and (6) ensure that, independently of  $\alpha$  and  $\omega$ , respectively, there is only one initial site and one final site for the sink route, since the corresponding arcs ( $x_{0\alpha}$  and  $x_{\omega q+1}$ ) must be 1. The combination of constraints (7) and (8) takes care of generating the route between  $\alpha$  and  $\omega$  that passes through all the sites where the sink has to sojourn for maximizing the network lifetime. In particular, the first constraint mandates that there must be one outgoing arc  $x_{kl}$  for every incoming one  $x_{jk}$  (with the natural exception of the two fictitious sites 0 and  $q + 1$ ). For instance, this is the case of arcs  $x_{34}$  and  $x_{49}$  in Figure 3(a), which are both set to 1. The remaining arc  $x_{45}$  is forced to be 0. According to constraint (8) for every site  $k$  where the sink sojourns ( $y_k = 1$ ) there must be a way to get there, i.e., there must be exactly one site  $j$  (which includes the fictitious site 0) from which  $k$  is reachable ( $x_{jk} = 1$ ). At the same time, the sink should not pass through sites where it does not sojourn. For instance, sites with  $k = 2, 3, 4, 9, 10, 15, 20, 25$  and 24 in Figure 3(a) are all and only those for which  $y_k = 1$ , i.e., these are all and only those sites that can be in the sink route. All other sites  $h$  are such that  $y_h = 0$ .

We note that flow conservation constraints (7) and (8) do not prevent the formation of cycles disjoint from the route from the origin to the destination. The (disjoint, non simple) route depicted in Figure 3(b) comprising nodes 2, 3, 4, 9, 10, 15, 20, 25 and 24 and nodes 16, 17, 22 and 21 (cycle) is possible according to this model up to constraint (8), since none of these constraints is violated by having  $y_{16} = y_{17} = y_{21} = y_{22} = 1$  as well as  $x_{16,17} = x_{17,22} = x_{22,21} = x_{21,16} = 1$ . This situation is undesirable, since quite unrealistic. It is practically impossible, for instance, to have the sink moving from site 9 (a site in the connected route from site 2 to site 24) to site 17 (a site in the cycle): Sites 9 and 17 are not directly connected, i.e., their distance is  $\geq d_{MAX}$ . Constraint (9) ensures that no such cycles are formed. (A similar constraint has been used in the integer programming formulation of the Traveling Salesman Problem (TSP) to avoid sub-tours [62].) According to constraint (10) a site  $k$  is associated with a “weight”  $u_k \geq 0$ . Constraint (9) imposes that the sites visited by the sink are traversed in increasing order, i.e., if  $x_{jk} = 1$  then  $u_j < u_k$ . This renders clearly impossible to return to the same node, and hence to form cycles like the one in Figure 3(b).

Some comments are in order. The parameter  $t_{min}$  has been introduced to assess the effect of different (higher or lower) sink mobility rates on network performance. For a given  $t_{min}$  the model will produce the sink route and sojourn time  $t_k \geq t_{min}$  at site  $k$  that maximizes network lifetime. By varying  $t_{min}$  we can explore a number of trade-offs. For instance, at higher  $t_{min}$ s we expect to have lower overhead (e.g., for route construction and release). Shorter  $t_{min}$ s result in a finer tuning of sojourn times at different sites (which maybe useful in achieving a longer network lifetime) at the price of increasing overhead.





(a) A sink route

(b) A disjoint cycle in the sink route

Figure 3: Sink optimum routes produced by constraints (5) to (8)

The model is also flexible in letting the user employ any method for determining the power consumption rate  $c_{ik}$  of each node  $i \in N$  when the sink sojourns at site  $k \in S$ . This cost depends on both node  $i$ 's transmission rate  $r_i$  and on the particular protocol for routing the packet to the sink sojourning at site  $k$ . The costs  $c_{iks}$  could be computed analytically [54], or they can be provided as input to the model from simulations or from real-data traffic traces [9, 10]. In short, the model can be customized to find the optimum lifetime for different routing protocols (by computing the corresponding values of  $f_{ik}$  and  $c_{ik}$ ).

We notice that the model as described does not allow the sink to pass twice for the same site. However, the model can be easily generalized to allow the sink to sojourn at the same site multiple times. A single “physical” site can be represented by  $h$  “logical” sites, where  $h$  is the number of times we want the sink to be able to pass through that site. The logical sites have no arcs between them, and are connected to all the (logical) sites of adjacent (physical) sites. Figure 4 depicts the case of two adjacent physical sites and the corresponding six logical sites ( $h = 3$ ).

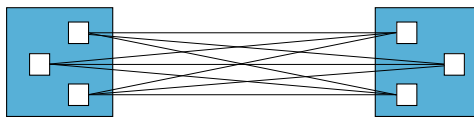


Figure 4: Two adjacent physical sites, six logical sites and their interconnections

With this simple modification optimal lifetime is obtained given that the sink is allowed to visit each site at most  $h$  times.

The MILP formulation improves over previously proposed models in multiple ways. The model is independent of a number of factors such as the specific sensor node deployment and sensor density; the sink site topology; the size and shape of the geographic area of deployment, and the sensor node technical

features (e.g., transmission radius, energy model, etc.). The given formulation includes a number of realistic constraints, such as the non-instantaneous movement of sink between sites potentially far apart from each other. Most importantly, and differently from all previously proposed LP solutions, this formulation explicitly includes the costs for changing sink sites.

### 5.2.2 The Greedy Maximum Residual Energy (GMRE) protocol

The solution presented above determines the movements and the sojourn times of the mobile sink at different sites so that network lifetime is maximized. Movements and times are determined by solving the described analytical model by providing as input a host of information concerning the *whole* networks. In other words, this solution is *centralized*: Information is collected at a “solver” and the resulting output is the best route for the sink.

Collecting information about network condition can be overwhelmingly expensive in terms of energy and time, and most of the times is unfeasible in resource constrained networks like WSNs. Therefore, protocols for controlled mobility have to be designed and deployed that can realistically be deployed in WSNs. The optimality is traded off for feasibility, as often happens. This motivates the definition of the following heuristic.

In the Greedy Maximum Residual Energy (GMRE) protocol [9, 10] the sink periodically moves to a new site. More specifically, every  $t_{min}$ , it decides whether to move or to stay at the current site. If it moves, the sink greedily selects the site within  $d_{MAX}$  from its current position surrounded by nodes that have the most energy left. The idea is that in time, this should most likely result into a balanced energy consumption throughout the network, and hence into a longer network lifetime. In order to decide whether to move or not, the sink gathers information about the residual energy at the nodes around each of the potential future sites (we call this energy value the residual energy at the site), and compare it with the residual energy at the current site. If there are adjacent sites with a residual energy higher than that at the current site, the sink moves to the site with the highest residual energy (selecting randomly among sites with the same residual energy in case of ties). Otherwise the sink stays at the current location. Key to the definition and implementation of GMRE is the communication to the sink of the residual energy at the adjacent sites. This communication proceeds in two phases. First, for each of the adjacent sites, the sink identifies one *sentinel* sensor node that will be in charge of measuring and reporting the residual energy at the site when requested by the sink. The second phase concerns the sink interrogation of the sentinels. This is performed whenever the sink has to decide whether to move or not. To implement the first phase we take advantage of the flooding performed by the sink when it makes the nodes aware of its new location. For this heuristic protocol we assume that a node that is in the “transmission vicinity” of a site (i.e., whose Euclidean distance from a site is less than or equal to the nodes transmission range) is aware of that. This can be obtained by endowing the nodes with a suitable localization mechanism (such as one of those described in [63]). The flooding message contains the coordinates of the current location of the sink. Upon receiving the flooding packet, a node knows if it is in the vicinity of a possible future sink site. In this case, it sends to the sink a (small) packet for its candidacy as sentinel. Upon receiving such packets the sink decides which is the sentinel for a given site. This mechanism also allows the sink to identify those sites that are isolated (no candidacy packet is received from nodes around that site). In this case, the sink will not consider that site as a possible future one. An example of the sentinel identification mechanism is depicted in Figure 5.

The sink current site  $E$  is indicated by a triangle. The squares  $A, B, C, D, F, G, H$  and  $I$  identify the adjacent sink sites. The figure shows what happens upon flooding the route construction message.

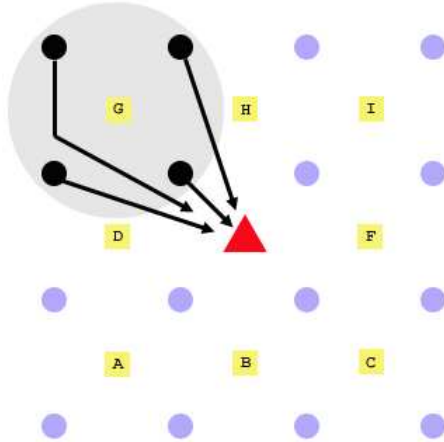


Figure 5: Sentinels at adjacent sites

Potential sentinels for sink site  $G$  are marked as black circles. Their distance from  $G$  is less than or equal the node transmission range. When such potential sentinels receive the flooded message, they answer back to the sink, identifying themselves as candidates. It is the sink that selects the (single) sentinel for a given possible future site.

The second phase starts when the sink has to decide whether to move to a new site or not. At this time, the sink interrogates the selected sentinels (one for each adjacent sink site) about the residual energy at their sites. This is accomplished by sending a (small) packet to the sentinels. When interrogated, the sentinels query their neighboring sensor nodes about their residual energy and communicate back to the sink the minimum of the obtained values (or any suitable function that can express how critical for the network lifetime is to place the sink in that area). Based on this information the sink selects to move to the site in the area with the highest minimum residual energy.

### 5.3 Multi-hop routing and uncontrolled mobility

The mathematical model and the distributed heuristic (GMRE) are clear example of controlled mobility: The sink moves, or attempts at moving, toward network location where nodal energy is available.

For the sake of benchmarking, it is interesting also to compare controlled mobility, and especially the more realistic, distributed GMRE, with a simple protocol for sink mobility that captures the case of uncontrolled, *random mobility* (RM) of the sink à la data MULEs. The difference here is that routing from the sensors to the sink is multi-hop. The heuristic is extremely simple: Every  $t_{min}$  the sink selects randomly and uniformly the new location among all the sites within distance  $d_{MAX}$  from the current. In case a site different from the current is selected, the sink moves to that site. We observe that even in this case, where the sink does not necessarily move to areas with high residual energy, improvements in network lifetime are obtained with respect to the case with an optimally placed static sink.

There are cases where the mobility, although neither random nor controlled by the network, can be exploited to improve network lifetime. In [56] Luo et al. successfully demonstrate this by simply moving the sink around the deployment area. Specifically, by searching the space of all possible sink mobility strategies that are periodic (recurrent sink movements within a constant period), the authors claim that a good mobility strategy is obtained by having the sink traveling at the periphery of the network. While

controlled, the sink movements are not guided by the network situation/activities.

Inspired by the solution by Luo et al. we devise a mobility strategy, termed Perimeter Mobility (PM). According to PM the sink visits cyclically the sites along the perimeter of the (square) deployment area. It stays at each site for  $t_{min}$  and then moves onto the next site in a counter-clockwise fashion.

It makes sense to compare this different cases of controlled and uncontrolled mobility, evaluating which solution is the more performing with respect to selected metrics. This is the natural next step: After having discussed different types of mobility and the corresponding algorithmic solutions (models and protocols) we proceed at selecting scenarios and metrics for comparing them.

## 6 Comparative Evaluation

A fundamental part of the creation and investigation of solutions for networking consists in testing the new solutions (protocols, methods, techniques, algorithms) and comparing them. We describe here the way this is usually done, giving some hints and insights on the process of evaluating the performance of a WSN.

Since experiments on real testbeds are costly and usually scale to a few tens of nodes, it is common to (start to) evaluate a protocol for WSN via simulations. In the attempt of considering the different layers of a WSN architecture, we use the VINT project network simulator ns2 [64]. This widely used software tool allows us to consider some physical layer properties and a MAC protocol CSMA/CS-like which are typical of wireless sensor networking.

Concerning the protocols for mobility presented here, the first question one is lead to investigate is about the properties of GMRE and of data MULEs mobility in a realistic, typical WSN scenario (static sensor nodes). The comparison should be about properties corresponding to relevant metrics. These include node energy consumption, network lifetime, end-to-end packet latency and probability of successfully delivering the monitored information to the sink.

The next natural step is that of investigating in deeper details the data MULE architecture and the case of mobility with multi-hop routing (GMRE, RM and PM) with the aim of identifying the impact of varying scenarios and protocols parameters on their performance.

Overall, the purpose of the following sections is that of introducing the reader to a deeper understanding of the performance trends when considering different application scenarios.

### 6.1 The choice of a suitable scenario

It is meaningful to choose a scenario that resembles a realistic deployment of a WSN with static nodes. Therefore, parameters like the number or network nodes, the size of the deployment area, the nodal transmission radius and initial energy, etc., should be chosen so that what depicted is substantially similar to a scenario one could find in real life. Here is a suitable example: 400 sensor nodes are deployed on a  $20 \times 20$  regular grid covering a squared area of side  $L = 400\text{m}$ . Sensor nodes have a transmission radius equal to 30m, which imposes a maximum of 8 neighbors per node. Nodes periodically generate packets at a rate of 0.5bps (low traffic scenario, typical of sensor networking). Packet size is equal to 512B. The buffer size of the sensor nodes varies among 10, 50 and 100 packets. Energy consumption follows the TR1000 specifications [65]. An ideal awake/asleep schedule is assumed in which nodes consume energy only when transmitting or receiving packets. The channel is ideal (no errors occurs when transmitting) and it has a data rate equal to 250Kbps. In the case of the data MULE architecture the number of available MULEs is

an important parameter: 1, 2 and 4 data MULEs can be a good starting point. In the case of data MULEs the deployment area is divided into 81 cells. The MULEs move from the center of one cell (source) to the center of another one (destination) according to the random waypoint model [27]. The MULE speed is set to 1m/s. While going from the source to the destination cell, a MULE stops at intermediate cells for collecting data. When in a cell, the MULE is able to communicate with all (and only) the sensor nodes located in that cell. The time spent in each cell is set to 1s. MULEs queues are considered unbounded (1000 packets, well above what needed in the considered scenarios). MULEs report to a sink which is (statically) placed at the center of the deployment area. When in proximity of the sink a MULE stops for the time needed to empty its queue.

Meaningful parameters for GMRE and the other heuristics are as follows. The sink moves among  $8 \times 8$  sink sites (grid deployment) at the speed of 1m/s. A shortest path-like routing is used to deliver data in a multi-hop fashion from the sensor to the current site of the sink. The parameter  $t_{min}$  is set to 50K (best performing value among the many tested). The parameter  $d_{MAX}$  is equal to 190m.

In all the experiments every node generates over 3000 packets during the simulation time. Finally, in order to obtain statistically meaningful results the same experiment (simulation run) should be performed on different networks for an enough number of times. That is why results are obtained by averaging over experimental outcomes from 100 different network topologies.

The following Table 1 gives a bird-eye view of the results of the comparison between GMRE and the data MULE solutions.

Table 1: GMRE and data MULEs, general comparison

	<b>GMRE</b>	<b>Data MULEs</b>
Data latency	Low	High
Energy consumption	Medium	Low
Packet delivery ratio	High	Medium
Computational needs	Medium	Low

In case of low network traffic and of limited deployment area size (such as those we considered) we observed that both GMRE and the data MULEs deliver all generated packets to the sink. In general, successful packet delivery for low/medium network traffic is more problematic for the data MULE solution than for the multi-hop approach (GMRE). As the deployment area grows in size, the inter-arrival time of a MULE at the same cell also grows so that overflow of sensor nodes queue may occur, resulting in a degradation of the packet delivery ratio. GMRE is instead always able to deliver packets to the sink until network congestion occurs.

The advantage of using an approach to mobility à la data MULEs becomes evident when energy consumption and sensor node computational needs are concerned. Given the single-hop nature of data exchange between the sensors and the MULEs, nodes are not required to implement a full protocol stack. Basic physical and MAC layer functions are sufficient for all data communication needs. This translates into lower energy consumption at the sensor nodes as well as lower sensor node complexity (and therefore nodal cost). In the basic scenario described above sensor nodes in a data MULE setting consume one order of magnitude less energy than in the more traditional multi-hop scenario. As expected, this gain in energy conservation is heavily paid in terms of end-to-end data latency. A difference as high as four orders of magnitude is observed between the two considered mobility solutions! This is caused by the extremely long time it takes to a MULE for visiting the same cell twice in a row as well as the time needed to go

back to the sink once a cell is visited. These two factors are of course dependent on the number of MULEs, their speed as well as on the size of the deployment area. Even varying MULEs speed from pedestrian (as shown here) to slow vehicular we would not be able to observe improvements of more than one order of magnitude. Only incurring the cost of introducing a high number of data MULEs would satisfy the latency requirements of many WSN applications.

## 6.2 Experiments for Data MULEs

The kind of experiments that are meaningful to perform in the data MULE architecture mainly concern the average number of packet discarded, and hence lost, because of buffer overflow and the sensor-to-sink packet data latency. It is interesting to notice that in order to interpret correctly the experimental results on these fundamental metrics, it is sometimes necessary to investigate other metrics and patterns, such as the inter-arrival times of the MULEs at the different cells, as well as the average sojourn time of a MULE in a cell.

Here is an example of experimental investigation in the basic scenario depicted above. We observed that the average number of packets discarded by the sensor nodes in the case of one MULE roaming among sensors with buffer size set to 10 (packets) is not high at all, especially for nodes that are toward the center of the deployment area. There are a few nodes, however, that, given the presence of a single MULE and small buffers, have to drop some packets. In this case the culprit is the movement of the MULE itself, given that, as it is well-known, the random waypoint-dictated mobility leads the MULE toward the center of the deployment area most of the times. The permanence at the center implies a MULE longer inter-arrival times at the cells along the area perimeter. This imposes a 4.5% packet loss at the nodes at the four corners.

This behavior is confirmed by an investigation of the inter-arrival times of 1, 2 and 4 MULEs at the same cell. In all the considered scenarios the inter-arrival times values are extremely high (in the thousands of seconds or more), explaining why the data MULE approach results in high packet latency. As the number of MULEs increases, it takes less and less time before a MULE arrives at a given cell, as expected. At the same time, given that the MULEs tend more to visit central cells than peripheral ones, the inter-arrival times are higher at the area borders.

MULEs visit each cell for the enforced 1s. The only cell that experience a longer sojourn time is the one where the sink is located. The higher the traffic and the area size and the lower the number of MULEs, the larger the amount of data stored in a MULE queue, which imposes longer times to unload the data to the sink. Therefore, the time needed for data transfer may exceed 1s.

The study of these last two metrics makes us to understand the results about what it is perhaps the most important metric for assessing whether or not the data MULE architecture is suitable for certain applications: Data latency. Measures on sensor-to-sink latency should take into account the latency incurred by data packets from the sensor to the sink, in a MULE queue and in a (source) node queue. A varying parameter should be the nodal queue size. We observe that, while it has an impact on the number of lost packets, the size of a node queue has little influence on the packet latency. Packets are transferred from the source queue (node or MULE) to the destination queue (MULE or sink) all at once, and the transfer time is quite little. Therefore, the end-to-end delay is given almost entirely by the time from when the packet is generated and a MULE arrives at the cell, plus the time it takes to the MULE to deliver it to the sink. The time a packet sits on its node queue depends on the number of MULEs, as expected. When the number of MULEs doubles, this time halves. We observe that this component is the most significant one. Given that the sink is placed at the center, the inter-arrival time of a MULE at

the sink is on average shorter than that at a (non-central) cell. The time spent by a packet on a MULE queue is basically independent of the number of MULEs. As a consequence, the end-to-end packet latency *almost* halves with doubling the MULEs, since only one of the delay component halves. In any case, and independently of the number of MULEs, the average latency of a packet is in the *thousands* of seconds. As expected, this is quite high.

The trends observed in the basic scenario are also confirmed in a setting where while the nodes are as dense as in the case with  $L = 400m$ , the deployment area is larger ( $L = 572m$ ).

### 6.3 Experiments on GMRE

As outlined in Section 6.1 controlled sink mobility and multi-hop data communication lead to higher energy consumption than in the data MULE case. However, this is the solution for satisfying more stringent latency constraints. What we show here is what is one possible, meaningful choice for evaluating a distributed heuristic like GMRE. In particular, we explore GMRE to explain the impact of different protocol parameters on its performance and for assessing the effectiveness of an energy-aware sink mobility heuristic over other sink mobility schemes. It makes sense to compare GMRE to the following schemes.

- Static sink placement (termed STATIC in the following). In this case the sink is optimally placed at the center of the deployment area.
- The sink mobility as generated by solving the MILP model presented in Section 5.2.1 (OPT).
- RM, the mobility heuristic outlined in Section 5.3, which is similar to the scenario with a randomly moving MULE, but with multi-hop routing.
- The scheme that we term PM (for Perimeter Mobility), according to which the sink moves cyclically along the perimeter of the deployment area (Section 5.3).

Experiments should be performed in different settings. For instance, one should vary the transmission radius (from 25m to 30m), the data routing protocol (shortest path-like vs. geographic) as well as the way nodes are deployed (grid positioning vs. random scattering). The results obtained are consistent with what we describe here (which refers to the basic scenario with transmission ranges of 30m and 25m). The conclusion is that GMRE is a simple but effective solution, achieving a network lifetime close to OPT's, i.e., the best that can be obtained by exploiting sink mobility in a multi-hop WSN. RM and PM have similar performance in terms of network lifetime with PM outperforming RM at higher  $t_{min}$ s. Both schemes improve considerably over STATIC, which advocates for the use of mobility in WSNs. However, the lack of energy awareness does not allow to match the results seen with GMRE (see [10, 59] for further details).

Figure 6(a) and 6(b) show the average network lifetimes achieved by STATIC, RM, PM, GMRE and OPT in networks whose nodes have transmission ranges of 25m and 30m. (The network lifetime has been defined as the time until the first node dies because of energy depletion.)

Improvements with respect to the static case can be as high as 350% when the sink moves according to GMRE in scenarios with 64 sink sites. In this case the GMRE lifetime is never more than 22% (25%) shorter than the OPT lifetime when the transmission range is 25m (30m) and  $t_{min}$  is kept below 250Ks. Improvements on network lifetime are obtained even when the sink moves randomly (RM heuristic), or when the sink moves along the area perimeter (PM heuristic). The latter experiences performances only slightly better than RM when the transmission range is 25m. RM leads to a network lifetime halfway between the STATIC lifetime and that obtained by OPT mobility.

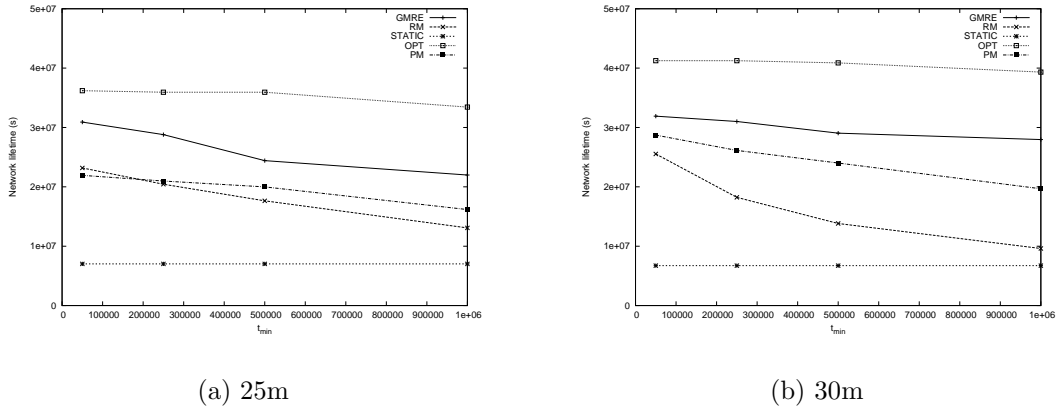


Figure 6: Average network lifetime for transmission ranges of 25m and 30m

In general, for GMRE, RM, and PM we notice that the lower the  $t_{min}$ , the higher the network lifetime. At higher  $t_{min}$ s the sink cannot finely decide the time to spend at each site, being stuck at a site for a longer time. This implies a less balanced energy consumption at the nodes. When  $t_{min}$  is very high it is not even possible for the sink to sojourn at all network sites, since the lifetime is reached before the sink can visit them all.

Even in the case of OPT mobility, lower  $t_{min}$ s correspond to longer lifetimes. In this case, however, the decrease in the network lifetime corresponding to higher  $t_{min}$ s is not as evident as for GMRE and RM. At this point one becomes curious about why this happens. First of all, in the OPT case  $t_{min}$  is simply a lower bound on the sojourn time: The sink has to stay at a site for that time, but does not have to move after it. It can stay an (optimum) amount of time after  $t_{min}$  and then move. In case of GMRE, RM and PM sink mobility, the decision about whether to move or not is due every  $t_{min}$  (after which the sink often moves). This has a twofold consequence. On one side, increasing  $t_{min}$  is more imposing for GMRE than for OPT in terms of fine tuning the sojourn time at a site. Furthermore, being able of deciding optimum sojourn times implies much lower sink mobility (OPT case), which corresponds to lower overhead for route management and hence to lower energy consumption with respect to that incurred by GMRE, RM and PM. Secondly, GMRE, RM and PM do not have a global view of the network topology and do not know the network traffic, i.e., how the nodes energy consumption evolves over time. This induces decreased performance with respect to OPT. The RM and PM heuristics do not enforce any energy-based criterion for sink movement, resulting in the worst performance among all the schemes with mobile sink we investigated. Even if GMRE takes into account nodes residual energy, the decision about whether to move or not and where, is based on the current status of the network and on a local view of the residual energy. According to the best “greedy” tradition, this could lead to a bad move with respect to global network lifetime maximization. The impact of such bad move is clearly higher for high  $t_{min}$ s: The wrong toll is paid for a longer time.

The OPT performance also degrades for higher values of  $t_{min}$ . In this case it converges to values that are typical of when the sink is kept static. For instance, for values of  $t_{min}$  approaching 7.013.801s the MILP model positions the sink at one of the sites in the center of the deployment area and leaves it there (static). However, for what explained above, increasing values of  $t_{min}$  are less critical in case of OPT mobility than in case of GMRE, RM, and PM. OPT network lifetime values start to decrease steeply at very high  $t_{min}$ s (not shown in the figures). Considering that OPT needs global information for deriving optimum sink mobility and sojourn times, and considering also the more “philosophical,” algorithmic differences



between OPT and GMRE, the fact that the heuristic never obtains network lifetimes more than 25% lower than OPT's, for relatively low sink mobility rate (small  $t_{min}$ s), can really be considered an excellent result.

Now, onto the flip side of this mobile story. Increases in network lifetime due to the sink mobility are paid in term of increased data packet latency. The reasons are quite clear. First of all, packets that are generated while the sink is moving and those in transit toward the sink, have to wait until routes to the new position of the sink are established. Furthermore, in order to balance energy depletion, the sink will spend time not only at the center of the deployment area but also along the borders. This imposes longer average routes and hence a higher packet latency than the latency experienced when the sink is statically placed at the center. (The latter is actually the dominant reason for increased latency in low sink mobility scenarios.) In any case, the average packet latency is *always* (i.e., for all the solution investigated) below half a second! This is far beyond measures of latency observed in the data MULE case, which were in the thousands of seconds (independently of the number of MULEs). The improvement is four/five orders of magnitude!

Table 2: Average packet latency (sec)

$t_{min}$	8 × 8 sink sites	
	50K	1M
OPT	.315	.315
GMRE	.32	.29
RM	.25	.26
PM	.30	.30
STATIC	.19	.19

Table 2 reports the average data latency incurred by packets in OPT, GMRE, RM, PM and STATIC for  $t_{min} = 50,000$ s and 1,000,000s. (Intermediate values are consistent with these ones.) Although small, and again: Always below .5s, there are differences between the average data latency experienced by the four schemes that we have investigated and compared. It is interesting to see why. When the sink sojourns on the perimeter or at the corner sites (as in GMRE, OPT and PM) we know that lifetime increases. However, these are also the cases when the average length of the routes to the sink increases, which implies, in turn, a (slightly) higher packet latency. It is thus reasonable to expect that a packet experiences lower latency when the sink is statically placed at the center of the sensor deployment area. The RM heuristic, which tends to move the sink to sites located centrally, is the second best mobility scheme in terms of latency. As  $t_{min}$  increases the sink tends to stay less at central sites, leading to higher average latencies experienced by RM packets. The opposite trend is observed for GMRE. For small  $t_{min}$ s the sink stays at sites on the corners and on the perimeter, which leads to an average packet latency up to 30% higher than those experienced in the RM case. When  $t_{min}$  increases the sink sojourns less at corner sites, which implies a decrease of the average latency.

Although the above discussion allows us to understand the subtle differences between the performance of the investigated schemes, there is very little variance among the measured latencies. All values range between .19s (STATIC) and .32s (GMRE and OPT).

As a result of this sample investigation, which the reader is encouraged to reproduce and vary, we obtain that GMRE shows a good trade-off between network lifetime gains and latency increases, promising to be a suitable solution for those application scenarios that cannot be covered by the data MULE mechanism.

## 7 Some Open Problems and Research Directions

Research on *mobile* WSNs is quite interesting and recent. Mobility in these networks can be tackled with from several points of view. Many are the research directions that could be taken. For instance, as demonstrated in the case of OPT and GMRE, having mathematical models is quite an important benchmark. Although centralized, and also often computationally tough, an optimal solution is quite useful for giving an idea of what can be obtained by designing and deploying a more realistic and necessary distributed heuristic. While progresses are being done in the multi-hop routing/mobile sink realm, this is quite uncharted territory in the case of the data MULE approach.

In the initial approach to controlling the mobility of the MULEs, as described here (Section 5.1), it is pointed out that when not enough MULEs are available, some nodes might not fall close enough to a MULE route. A proposed solution is that of introducing multi-hop routes (organized as trees) to nodes that can directly communicate with a MULE when it passes. This approach could be studied in more details, and improvement can be obtained by investigating, for instance, which is the depth of the tree that best keeps energy consumption low while allowing packets to get to the MULE, i.e., without losing them. One can also think of dividing the data packets into priority classes, so that the MULE trajectory is controlled by the network in such a way that the MULE goes first where high-priority data is to be collected.

## 8 Conclusions

This chapter explores ways for using the mobility of network components in large networks of resource constrained devices, like wireless sensor networks (WSNs). The aim is that of exploiting mobility for improving the performance of these networks in terms of network lifetime, throughput and connectivity without significantly affecting data routing and end-to-end latency. Whereas energy efficient solutions have been proposed for WSNs with statically placed sensors and one or more static data collection points (the *sinks*), our investigation delves into the various ways the sensor nodes, some mobile relays or the sinks can be made mobile for improving network performance. We review the most recent proposals for mobility use and management in WSNs, considering the pros and cons, as well as the costs and trade-offs of using mobile sensors, mobile relays and moving sinks. In the final part of the chapter we give examples on how, once selected typical solutions for dealing with mobility of sinks and relays, these solutions are compared. The aim is to introduce the reader to the selection of suitable scenarios and performance metric, and that of illustrating a general approach to “reading” experimental results. In particular, we illustrate controlled and uncontrolled mobility both in the case of networks with single-hop routing of the packets to a passing mobile relay (data MULE architecture), and in the case of controlled and uncontrolled mobility of a sink roaming through the network and having the data delivered to it in a multi-hop fashion. The two different network architectures, with single-hop vs. multi-hop data routing, are described in details and compared. Our comparison concerns key metrics of interest for WSNs, which include network lifetime and sensor-to-sink data latency. Besides providing evidence of the the advantages of using mobility, the chapter also provides insights on different techniques for protocol design and performance evaluation and comparison.

## 9 Acknowledgments

This work was supported in part by the European FP6 027227 IP Project “E-Sense (Capturing Ambient Intelligence for Mobile Communications through Wireless Sensor Networks).”

## 10 Exercises

1. **Review questions.** a) Why is it less convenient to use resource-constrained mobile sensor nodes rather than resource-rich mobile elements like relays and sinks? b) Give an example that clearly motivates the use of mobility in WSNs. c) What metric is most affected by the use of mobile relays rather than mobile sinks? d) For what kind of application it is clearly more advisable to use the data MULE approach instead of a multi-hop solution with mobile sink? e) List five metrics of interest that should be evaluated when dealing with mobile WSNs. f) Describe pros and cons of developing a mathematical formulation of a problem for WSNs like the one described in Section 5.2.1. g) What are the major differences, in terms of mobility, when comparing GMRE, RM and PM? h) If the MULEs move according to the random waypoint mobility model, where are they more likely to be found? What does this imply in terms of network performance? (I.e., which metric is the most affected in this case?)
2. Consider the MILP model presented in Section 5.2.1. What are the problems that would occur if constraint (9) is removed? Why is this removal a problem?
3. Consider the MILP model presented in Section 5.2.1. Imagine that instead of knowing the sink sites, several different *routes* for the sink are available. Rewrite the model in such a way that the output is the sequence of routes that maximize network lifetime.
4. Consider again the MILP model presented in Section 5.2.1. Rewrite the model so that the sink can pass for the same site at most 5 times.
5. Consider the pattern of energy consumption depicted in the following figure (that refers to the basic scenario described in Section 6.1 with  $8 \times 8$  sink sites). The darker the area, the higher the energy consumption.

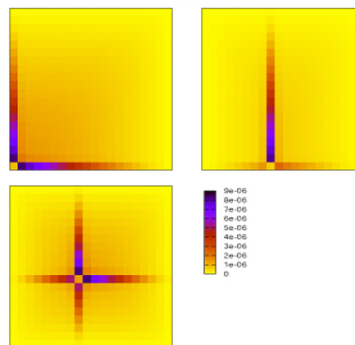


Figure 7: Average nodal energy consumption when the sink sojourns at different sites

Given this pattern, describe where an heuristic will more likely send the sink (i.e., to which sites). Justify your answer.

6. What is the main difference between the data MULE approach and the PM scheme described in Section 5.3?

## References

- [1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds., *Mobile Ad Hoc Networking*. Piscataway, NJ and New York, NY: IEEE Press and John Wiley & Sons, Inc., April 2004.
- [2] I. F. Akyildiz, W. Su, Y. Sanakarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.
- [3] E. H. Callaway, Jr., *Wireless Sensor Networks: Architectures and Protocols*. Boca Raton, FL: Auerbach Publications, August 2003.
- [4] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater acoustic sensor networks: Research challenges,” *Elsevier’s Journal of Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, May 2005.
- [5] G. Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, “Distributed search and rescue with robot and sensor team,” in *Proceedings of the 4th International Conference on Field and Service Robotics, FSR 2003*, Lake Yamanaka, Japan, July 2003, pp. 327–332.
- [6] K. Kotay, R. Peterson, and D. Rus, “Experiments with robots and sensor networks for mapping and navigation,” in *Proceedings of the International Conference on Field and Service Robotics, FSR 2005*, Port Douglas, Australia, July 29–31 2005.
- [7] R. Brooks, P. Ramanathan, and A. Sayeed, “Distributed target classification and tracking in sensor networks,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163–1171, August 2003.
- [8] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. A. Stankovic, “An assisted living oriented information system based on a residential wireless sensor network,” in *Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference*, Arlington, VA, April 2–4 2006, pp. 95–100.
- [9] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and M. Z. Wang, “A new MILP formulation and distributed protocols for wireless sensor networks lifetime maximization,” in *Proceedings of the IEEE International Conference on Communications, ICC 2006*, vol. 8, Istanbul, Turkey, June 11–15 2006, pp. 3517–3524.
- [10] —, “Controlled sink mobility for prolonging wireless sensor networks lifetime,” *ACM/Elsevier Wireless Networks*, 2007, to appear.
- [11] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet,” in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS-X*, San Jose, CA, October 5–9 2002, pp. 96–107.

- [12] Q. Li and D. Rus, “Sending messages to mobile users in disconnected ad-hoc wireless networks,” in *Proceedings of 6th ACM Annual International Conference on Mobile Computing and Networking, MobiCom 2000*, Boston, MA, August 6–11 2000, pp. 44–55.
- [13] M. Grossglauser and D. N. C. Tse, “Mobility increases the capacity of ad-hoc wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, August 2002.
- [14] W. Zhao, M. H. Ammar, and E. W. Zegura, “A message ferrying approach for data delivery in sparse mobile ad hoc networks,” in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2004*, Roppongi Hills, Tokyo, Japan, May 24–26 2004, pp. 187–198.
- [15] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, “Exploiting mobility for energy-efficient data collection in sensor networks,” in *Proceedings of the IEEE Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, WiOpt 2004*, Cambridge, UK, March 24–26 2004.
- [16] W. Zhao, M. H. Ammar, and E. W. Zegura, “Controlling the mobility of multiple data transport ferries in a delay-tolerant network,” in *Proceedings of IEEE Infocom 2005*, vol. 2, Miami, FL, March 13–17 2005, pp. 1407–1418.
- [17] H. Jun, W. Zhao, M. H. Ammar, H. W. Zegura, and C. Lee, “Trading latency for energy in wireless ad hoc networks using message ferrying,” in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications, PerComm 2005*, Kawai, HA, March 8–12 2005, pp. 220–225.
- [18] M. Mukarram Bin Tariq, M. H. Ammar, and E. W. Zegura, “Message ferry route design for sparse ad hoc networks with mobile nodes,” in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2006*, Firenze, Italy, May 22–25 2006, pp. 37–48.
- [19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Performance analysis of mobility-assisted routing,” in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2006*, Firenze, Italy, May 22–25 2006, pp. 49–60.
- [20] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang, “Towards mobility as a network control primitive,” in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2004*, Roppongi Hills, Tokyo, Japan, May 24–26 2004, pp. 163–174.
- [21] G. Wang, G. Cao, and T. F. La Porta, “Movement-assisted sensor deployment,” in *Proceedings of IEEE INFOCOM 2004*, vol. 4, Hong Kong, March 7–11 2004, pp. 2469–2479.
- [22] —, “Proxy-based sensor deployment for mobile sensor networks,” in *Proceedings of the First IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2004*, Fort Lauderdale, FL, October 25–27 2004, pp. 493–502.
- [23] A. Howard, M. J. Matarić, and G. S. Sukhatme, “An incremental deployment algorithm for mobile sensor networks,” *Kluwer Autonomous Robots, Special Issue on Intelligent Embedded Systems (G.S. Sukhatme, editor)*, vol. 13, no. 2, pp. 113–126, September 2002.

- [24] —, “Mobile sensor networks deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotic Systems, DARS 2002*, Fukuoka, Japan, June 25–27 2002.
- [25] R. Rao and G. Kesidis, “Purposeful mobility for relaying and surveillance in mobile ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 225–232, July–September 2004.
- [26] I. Chatzigiannakis and S. Nikolettseas, “An adaptive compulsory protocol for basic communications in highly changing ad-hoc mobile networks,” in *Proceedings of the International Parallel and Distributed Processing Symposium, IPDPS 2002*, Fort Lauderdale, FL, April 15–19 2002, pp. 193–202.
- [27] R. C. Shah, S. Roy, S. Jain, and W. Brunette, “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 2003*, Anchorage, AK, May 11 2003, pp. 30–41.
- [28] T. Small and Z. J. Haas, “Resource and performance tradeoffs in delay-tolerant wireless networks,” in *Proceedings of the ACM SIGCOMM 2005 Workshop on Delay-Tolerant Networking and Related Networks, WDTN 2005*, Philadelphia, PA, August 22–26 2005.
- [29] L. Song and D. Hatzinakos, “Dense wireless sensor networks with mobile sinks,” in *Proceedings of the IEEE International Conference on Acoustic Speech, and Signal Processing, ICASSP 2005*, vol. 3, Philadelphia, PA, March 18–23 2005, pp. 677–680.
- [30] Y. Tirta, Z. Li, Y.-H. Lu, and S. Bagchi, “Efficient collection of sensor data in remote fields using mobile collectors,” in *Proceedings of the 13th International Conference on Computer Communications and Networks, ICCCN 2004*, Chicago, IL, October 11–13 2004, pp. 515–519.
- [31] Y. Tirta, B. Lau, N. Malhotra, S. Bagchi, Z. Li, and Y.-H. Lu, *IEEE Monograph on Sensor Network Operations*, S. Phoha and T. F. La Porta, editors. Piscataway, NJ: IEEE Press and John Wiley and Sons, Inc., December 2 2005, ch. Controlled Mobility for Efficient Data Gathering in Sensor Networks with Passively Mobile Nodes.
- [32] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin, “Intelligent fluid infrastructure for embedded networks,” in *Proceedings of the 2nd ACM/SIGMOBILE International Conference on Mobile Systems, Applications, and Services, MobySys 2004*, Boston, MA, June 6–9 2004, pp. 111–124.
- [33] D. Jea, A. A. Somasundara, and M. B. Srivastava, “Multiple controlled mobile elements (data mules) for data collection in sensor networks,” in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2005*, ser. LNCS, V. K. Prasanna, S. Iyengar, P. G. Spirakis, and M. Welsh, Eds., vol. 3560, Marina del Rey, CA, June 30–July 1 2005, pp. 244–257.
- [34] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, “Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines,” in *Proceedings of the 25th IEEE International Real-Time Systems Symposium, RTSS 2004*, Lisbon, Portugal, December 5–8 2004, pp. 296–305.

- [35] A. Kansal, M. Rahimi, D. Estrin, W. J. Kaiser, G. J. Pottie, and M. B. Srivastava, “Controlled mobility for sustainable wireless sensor networks,” in *Proceedings of the First IEEE Annual Conference on Sensor and Ad Hoc Communications and Networks, SECON 2004*, Santa Clara, CA, October 4–7 2004, pp. 1–6.
- [36] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, “Controllably mobile infrastructure for low energy embedded networks,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, August 2006.
- [37] E. Ekici, Y. Gu, and D. Bozdog, “Mobility-based communications in wireless sensor networks,” *IEEE Communications Magazine*, vol. 44, no. 7, pp. 56–62, July 2006.
- [38] W. Wang, V. Srinivasan, and K.-C. Chua, “Using mobile relays to prolong the lifetime of wireless sensor networks,” in *Proceedings of the 11th Annual ACM/SIGMOBILE International Conference on Mobile Computing and Networking, MobiCom 2005*, Cologne, Germany, August 28–September 2 2005, pp. 270–283.
- [39] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon, “Minimum energy asynchronous dissemination to mobile sinks in wireless sensor networks,” in *Proceedings of the First International Conference on Embedded Networked Sensor Systems, SenSys 2003*, Los Angeles, CA, November, 5–7 2003, pp. 193–204.
- [40] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, February 2003.
- [41] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, “A two-tier data dissemination model for large scale wireless sensor networks,” in *Proceedings of the 8th ACM Annual International Conference on Mobile Computing and Networking, MobiCom 2002*, Atlanta, GA, September 23–28 2002, pp. 148–159.
- [42] J. G. Jetcheva and D. B. Johnson, “Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks,” in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, MobiHoc 2001*, Long Beach, CA, October 4–5 2001, pp. 33–44.
- [43] K. Hwang, J. In, Y. Yun, and D. Eom, “Dynamic sink oriented tree algorithm for efficient target tracking of multiple mobile sink users in wide sensor fields,” in *Proceedings of the 60th IEEE Vehicular Technology Conference, VTC 2004-Fall*, vol. 7, Los Angeles, CA, September 26–29 2004, pp. 4607–4610.
- [44] X. Hu, Y. Liu, M. J. Lee, and T. N. Saadawi, “A novel route update design for wireless sensor networks,” *ACM Mobile Computing and Communications Review*, vol. 8, no. 1, pp. 18–26, January 2004.
- [45] K. Akkaya and M. Younis, “Energy-aware to mobile gateway in wireless sensor networks,” in *Proceedings of the IEEE Globecom 2004 Workshops*, Dallas, TX, November 29–December 3 2004, pp. 16–21.

- [46] H. L. Xuan and S. Lee, "A coordination-based data dissemination protocol for wireless sensor networks," in *Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, ISSNIP 2004*, Melbourne, Australia, December 14–17 2004, pp. 13–18.
- [47] L. Tong, Q. Zhao, and S. Adireddy, "Sensor networks with mobile agents," in *Proceedings of the IEEE Military Communication Conference, MILCOM 2003*, vol. 1, Boston, MA, October 13–16 2003, pp. 705–710.
- [48] P. Venkitasubramaniam, S. Adireddy, and L. Tong, "Sensor networks with mobile agents: Optimal random access and coding," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1058–1068, August 2004.
- [49] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum cost heterogeneous sensor network with a lifetime constraint," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 4–15, January/February 2005.
- [50] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "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 2003*, ser. LNCS 2634, F. Zhao and L. Guibas, Eds., Palo Alto, CA, April 22–23 2003, pp. 129–145.
- [51] P. Venkitasubramaniam, Q. Zhao, and L. Tong, "Sensor networks with multiple mobile access points," in *Proceedings of the 38th Annual Conference on Information Sciences and Systems, CISS 2004*, Princeton, NJ, March 17–19 2002.
- [52] P. Baruah, R. Uргаonkar, and B. Krishnamachari, "Learning-enforced time domain routing to mobile sinks in wireless sensor fields," in *Proceeding of the 29th Annual IEEE International Conference on Local Computer Networks, LCN 2004*, Tampa, FL, November 16–18 2004, pp. 525–532.
- [53] S. R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in *Proceedings of IEEE Globecom 2003*, vol. 1, San Francisco, CA, December 1–5 2003, pp. 377–381.
- [54] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *Proceedings of the 38th Hawaii International Conference on System Sciences*, Big Island, Hawaii, January 3–6 2005.
- [55] I. Papadimitriou and L. Georgiadis, "Maximum lifetime routing to mobile sink in wireless sensor networks," in *Proceedings of the 2005 International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2005*, Split, Croatia, September 15–17 2005.
- [56] J. Luo and J.-P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings of IEEE Infocom 2005*, vol. 3, Miami, FL, March 13–17 2005, pp. 1735–1746.
- [57] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Networking*, vol. 12, no. 4, pp. 609–619, August 2004.



- [58] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, “Controlling sink mobility in wireless sensor networks: A new model and protocols,” in *Poster at ACM/SIGMOBILE MobiCom 2005*, Cologne, Germany, August 28–September 2 2005.
- [59] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and M. Z. Wang, “Protocols and model for sink mobility in wireless sensor networks,” *ACM Mobile Computing and Communication Review, MC<sup>2</sup>R*, vol. 10, no. 4, pp. 28–30, October 2006.
- [60] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Duke University, Durham, NC, Tech. Rep. CS-200006, April 2000.
- [61] S. Basagni, M. Elia, and R. Ghosh, “ViBES: Virtual backbone for energy saving in wireless sensor networks,” in *Proceedings of the IEEE Military Communication Conference, MILCOM 2004*, Monterey, CA, October 31–November 3 2004.
- [62] C. E. Miller, A. W. Tucker, and R. A. Zemlin, “Integer programming formulation of traveling salesman problems,” *Journal of the ACM*, vol. 7, no. 4, pp. 311–325, October 1960.
- [63] A. Savvides and M. B. Srivastava, “Location discovery,” in *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. Piscataway, NJ and New York, NY: IEEE Press and John Wiley and Sons, Inc., April 2004, ch. 8, pp. 231–254.
- [64] The VINT Project, *The ns Manual*. <http://www.isi.edu/nsnam/ns/>, 2002.
- [65] “ASH transceiver designer’s guide,” [www.rfm.com](http://www.rfm.com), May 19 2004.