Chapter 18

Prolong The Lifetime of Wireless Sensor Networks Through Mobility: A General Optimization Framework

Jun Luo and Liu Xiang

Abstract Though mobility is rarely considered in traditional *wireless sensor networks* (WSNs), actively exploiting mobility to improve the performance of WSNs has been increasingly recognized as an important aspect of designing WSNs. This chapter focuses on exploiting mobility to improve the network lifetime of a WSN. We present a general optimization framework that is able to capture several aspects of maximizing network lifetime (MNL) involving mobile entities. Based on this framework, we conduct an in-depth analysis on each of these aspects and also describe algorithms that can be used to solve the resulting optimization problems. We also present certain numerical results where engineering insights can be acquired.

18.1 Mobile Elements in Wireless Sensor Networks: Stir Up the Pond

Traditionally, mainstream research envisioned *wireless sensor networks* (WSNs) as an avatar of **static** multi-hop wireless networks [1]. Although the mobility issues were present even from the early stage of the WSN-related investigations (e.g., [27]), those issues failed to attract a lot of attention until very recently. The reason is twofold. On one hand, it is much more difficult, from both theoretical and practical point of views, to deal with networks with mobile entities. On the other hand, we only recently realized that we could **actively** utilize mobility rather than having to **passively** accept its inevitable presence. In this chapter, we will focus on one of the active applications of mobility to improve an important aspect of network

Luo Jun
School of Computer Engineering, Nanyang Technological University (NTU), Singapore, e-mail: junluo@ntu.edu.sg

Xiang Liu
School of Computer Engineering, Nanyang Technological University (NTU), Singapore, e-mail: xiangliu@pmail.ntu.edu.sg

performance – *lifetime*. However, instead of directly addressing the main topic, we will start with a brief survey of various aspects of WSNs that are concerned with mobility, which should provide the readers with a better technological context in understanding the main topic.

Typically, mobility gets involved in WSNs in two ways: either passively or actively. In the former case, mobility comes as input to certain system design aspects of WSNs, and a certain design has to cope with the negative effects (e.g., unreliable communication channels and high cost of route maintenance) brought by mobility. Typical instances of this case exist where either *sensor nodes*[1] or *sinks*[2] need to move according to the application requirements: for example, a sensor node or a sink may be attached to a tactic unit in a battle field [16, 37], or a sink is someone's PDA that helps him/her to navigate within a sensor field [20]. Another approach (e.g., [17]) exploits the anyway present mobility as an efficient replacement for connectivity and data propagation redundancy. In the latter case, mobility is actively introduced to a network by system designers, aiming at improving certain performance aspects of the original design that consists of only static network components. Here, both theoreticians and practitioners are trying to make the best use of mobility while still coping with its side-effects. Typical performance aspects that may benefit from the introduction of mobile entities are load balancing/lifetime maximization [7, 21, 22, 24, 30, 32, 35, 36], buffer overflow prevention [15, 33], coverage enhancement [34], and high fidelity data collection [3].

Although many performance aspects of WSNs may benefit from mobility, the lifetime issue seems to have attracted the majority of attention and contributions. Therefore, we focus on the issue of prolonging network lifetime using mobility in this chapter. As shown in Figure 18.1, the traffic load within a WSN is highly unbalanced among nodes that have different distances from the sink. Whereas no routing strategy may alleviate such an imbalance, actively moving certain network entities may further balance the load and hence improve the lifetime. Basically,

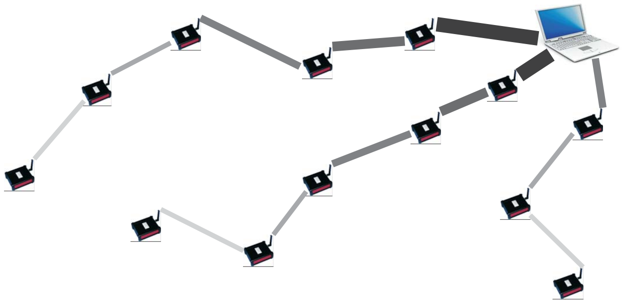


Fig. 18.1 The unbalanced traffic load in a WSN due to the converging traffic pattern that accumulates traffic towards the "last hop" nodes.

[1] In this paper, the words *sensor node* and *node* are used interchangeably.

[2] These are the entities that collect data from WSNs; sometimes they are also termed *base stations*.

two approaches, namely *fast mobility* and *slow mobility*, are used to exploit entity (sink or node) mobility to improve network lifetime. They are distinguished by the relationship between the moving speed of an entity and the tolerable delay of the data delivery. In the former case, an entity (typically a sink) can transport data with its mechanical movements if its speed is sufficiently high so that the mechanical data transportation yields a tolerable data delivery delay. In this case, nodes may be totally or partially spared from the traffic forwarding load and can hence save their energy. We term this approach **fast** mobility approach, as the entity should move at a sufficiently high speed. In the latter case, moving an entity, even very infrequently (say once a day or week), may still benefit the network lifetime, thanks to the distribution of the role of bottleneck nodes within the entire network. We denote this approach as **slow** mobility, because the moving speed of a mobile entity is too low to be used for transporting data within a tolerable delay (but it barely affects the delay due to the way it is used).

The general reason that mobility, no matter fast or slow, can improve network lifetime lies in the fact that mobility increases the dimension (thus the degree of freedom) of the problem. This follows the general principle that optimizing an objective in a high-dimension space always leads to a result no worse than what can be achieved in a subspace of reduced dimension. However, solving problems in high-dimension space incurs a higher complexity. In the remainder of this chapter, we will discuss a general optimization framework that can be used to model and formulate such problems, and we will highlight the solution techniques that are used.

Under the slow mobility regime, the mobility may take a discrete form: the movement trace consists of several anchor points between which the mobile entities move and at which they pause. Consequently, data packets have to be carried from their origins to the sinks through multi-hop routing. In Sections 18.2 and 18.3, we discuss the approach that makes use of the slow sink mobility to balance the traffic load within a WSN and hence to improve the network lifetime, and we introduce a general optimization framework to model, formulate, and solve the problem. The approach considered in Section 18.2 aims at obtaining or approximating the optimal movement traces of multiple mobile sinks, but the anchor points that constitute the traces can only be chosen from a predefined set of locations. The extension reported in Section 18.3 takes one step further by relaxing the location constraint: should the algorithm in Section 18.2 obtain an optimal solution, an optimal unconstrained trace could be obtained exactly or be approximated to an arbitrarily small granularity, at a cost of solving many instances of the problem addressed in Section 18.2.

In Section 18.4, we present an approach that extends the general framework discussed in Section 18.2 to a distinct direction. This approach, though still categorized as a slow mobility approach, chooses to move certain powerful (in terms of energy reserve) nodes rather than sinks. The underlying rationale is to use these powerful (mobile) nodes to replace certain highly loaded (static) nodes from time to time, which could substantially reduce the energy consumption of those static nodes and hence prolong the network lifetime. Although it has been shown that this *mobile node* approach is in general inferior to the *mobile sink* approaches discussed in Sections 18.2 and 18.3, the proposal is still meaningful because moving sink(s) might not always be feasible. Also, as shown by our numerical results in Section 18.4.3, the performance of this approach can be comparable to that of the mobile sink approach.

Under the fast mobility regime, entities may move fast enough to deliver data with a tolerable delay, WSNs can hence take advantage of mobility capacity [14]. We term this approach *mobile relay* although the mobile entities are the sinks, because the mobile sink, instead of only receiving multi-hop transmissions, may "pick up" data from nodes (through one-hop transmissions) and transport the data with mechanical movements. In Section 18.5, we first extend our framework to show the complexity of finding an optimal tour for the mobile relay, and then we will introduce a possible simplification of the problem along with the approximation algorithms designed for a single mobile relay.

Although the algorithms we discuss are centralized, they serve as benchmarks or guidance for distributed implementations (e.g., [7,24]). The intention of this chapter is to give an in-depth treatment on the issue of using mobility to prolong network lifetime from the theoretical perspective, so we do not claim any thoroughness in surveying the vast literature on the mobility related issues in WSNs; such literature is too vast even for this specific topic.

18.2 Balancing Traffic Load with Mobile Sinks: The Case of Constrained Mobility

As shown in Figure 18.1, it is the converging traffic pattern of WSNs that leads to the unbalanced traffic load within a network. Consequently, simply manipulating the routing protocols would not fully address the lifetime issue. Fortunately, recent proposals (e.g. [11, 21]) suggest that moving the sinks could distribute the role of bottleneck nodes (those close the sinks) over time and thus even out the load, as illustrated in Figure 18.2. To support the feasibility of such an approach, a simple implementation is also reported later [23]. In this section, we are aiming at develop-

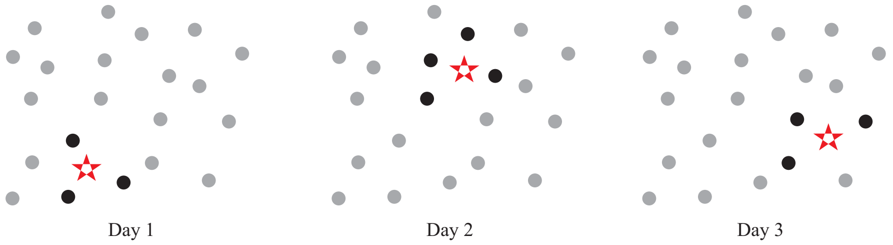


Fig. 18.2 Using a mobile sink to balance the traffic load within a WSN. The star represents the sink, and nodes with a darker color are the bottleneck nodes. Note that the mobility is slow, as the sink may change it location very infrequently.

ing an optimization framework to analyze the problem and also a solution technique to solve the optimization problem. In addition, we are making this framework sufficiently general such that we can extend it in different directions later on.

18.2.1 Network Model and Problem Formulation

For a WSN, we use set $\mathscr{N} : |\mathscr{N}| = n$ to represent all the sensor nodes and set $\mathscr{S} : |\mathscr{S}| = m < n$ for the sinks. The former set is static and it determines the basic topology of the network, while the latter changes its layout occasionally so as to collect data and to balance the traffic load. We allow the sinks to choose their locations only within a finite set $\mathscr{V}$. We denote by *on-graph* mobility the case $\mathscr{V} = \mathscr{N}$, and by *off-graph* mobility the case $\mathscr{V} \supset \mathscr{N}$. There is a cost assignment $\mathbf{c} : \mathscr{V} \times \mathscr{V} \to \mathbb{R}^+$, such that a link (i, j) exists (or $\exists(i, j) \in \mathscr{E}$) if and only if 1) $i \in \mathscr{N}$, 2) $j \in \mathscr{N}$ or j is a sink, and 3) the transmission energy[3] e_i^{T} of node i is no less than $c(i, j)$. All these allow us to model the WSN as a digraph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$. We assume that wireless communication is the dominating energy-consuming factor and hence omit other energy consuming functions such as sensing. We emphasize the crucial behaviors of mobile sinks in our investigation: each sink travels among a set of locations chosen within $\mathscr{V}$ and stays with each one of them to collect data from the whole WSN for a relatively long time, which makes the traveling time negligible. When co-located with a node $i \in \mathscr{N}$, the sink, apart from collecting data from other nodes, inherits all the energy-consuming functions of the co-located node.

Initialized with an energy reserve E_i for each node $i \in \mathscr{N}$, the network is said to be "dead" once some node runs out of battery. In other words, the network lifetime T is defined as the time when the first node dies [6]. Taking into account the fact that sinks change their locations from time to time, we define an *epoch* as a time duration within which no sink changes its position. Thus T can be represented by the summation of time duration of each epoch t_k. In formulating the *maximizing network lifetime* (MNL) problem, we consider the constraints related to $\mathscr{V}$ and $\mathscr{S}$ separately: each node $i \in \mathscr{V}$ is associated with a flow conservation and an energy conservation, while each sink $s \in \mathscr{S}$ is constrained by its location choices. The two sets of constrains are coupled by an indicator matrix $[\delta_{is}^k]$ where $\delta_{is}^k = 1$ if sink s is co-located with $i \in \mathscr{V}$ during the kth epoch and $\delta_{is}^k = 0$ otherwise. We formally present the mixed-integer nonlinear programming of MNL as below, and the detailed notations can be found in Table 18.1.

[3] The physical features of the radio of node i are usually specified by a tuple (P_i, R, μ); here P_i is the transmission power, R is the data rate, μ is the threshold (specified by the required *bit error rate* (BER) of a given modulation scheme that produces the rate R) such that a link (i, j) may operate on rate R iff $P_i \cdot \eta_{i,j} \geq \mu$, where $\eta_{i,j}$ represents the fading, shadowing, and path loss effects between nodes i and j. Our model can be considered as a more generalized form of the aforementioned model, as $e_i^{\mathrm{T}} = \frac{P_i}{R} \geq \frac{\mu}{\eta_{i,j}} = c(i, j)$ is indeed the criterion to indicate the existence of link (i, j). Note that, under our model, e_i^{T} may have a unit of, for example, Joules/Bit.

Table 18.1 Notations used through out this section.

	Notation of The Network
$\mathscr{N}$	The set of sensor nodes in the WSN
n	$= \lvert\mathscr{N}\rvert$, the number of sensor nodes in the WSN
$\mathscr{S}$	The set of sinks in the WSN
m	$= \lvert\mathscr{S}\rvert$, the number of sinks in the WSN
$\mathscr{V}$	$\supset \mathscr{N}$, the potential locations of the mobile sinks
$\mathscr{E}$	The set of all feasible wireless links
$\mathbf{c}$	Cost assignment that defines feasible links
T	Network lifetime
$\hat{T}$	Maximum network lifetime
t_k	The duration of the kth epoch
	Notation of Sensor Nodes
E_i	Initial energy reserve of node i
e_i^{T}	Energy consumption for node i to transmit a unit of data
e^{R}	Energy consumption for any node to receive a unit of data
λ_i	Information generation rate of node i
r_{ij}^k	Data rate from node i to node j during the kth epoch
r_i^k	Data rate draining out of the WSN from node i during the kth epoch
q_{ij}^k	Quantity of data from node i to node j during the kth epoch
q_{is}^k	Quantity of data from node i to sink s during the kth epoch
P_{is}^k	The set of paths from node i to sink s during the kth epoch
P_i^k	The set of paths going through node i during the kth epoch
	Notation of Sinks
$\mathscr{L}_k$	$\subset \mathscr{V}$, the set of sink locations during the kth epoch
δ_{is}^k	Indicator for the location of sink s during the kth epoch
sl_k	$= [\delta_{is}^k]_{i\in\mathscr{V},s\in\mathscr{S}}$: The sink layout during the kth epoch

$$\text{maximize} \quad T = \sum_k t_k \tag{18.1}$$

$$\text{subject to} \quad \sum_{(i,j),(j,i)\in\mathscr{E}} \left(r_{ij}^k - r_{ji}^k\right) + r_i^k \delta_{is}^k \;\geq\; \lambda_i \quad \forall i,k \tag{18.2}$$

$$\sum_{k:\delta_{is}^k \neq 1} \left[\sum_{(i,j),(j,i)\in\mathscr{E}} \left(r_{ij}^k e_i^{\mathrm{T}} + r_{ji}^k e^{\mathrm{R}}\right) \right] t_k \;\leq\; E_i \quad \forall i \tag{18.3}$$

$$\sum_{s\in\mathscr{S}} \delta_{is}^k \;\leq\; 1 \quad \forall i,k \tag{18.4}$$

$$\sum_{i\in\mathscr{N}} \sum_{s\in\mathscr{S}} \delta_{is}^k \;=\; m \quad \forall k \tag{18.5}$$

$$t_k, r_{ij}^k, r_i^k \;\geq\; 0 \quad \forall i,j,s,k \tag{18.6}$$

$$\delta_{is}^k \in \{0,1\} \quad \forall i,s,k \tag{18.7}$$

By this formulation, we implicitly assume that the data rate between any two nodes i and j, r_{ij}^k, is feasible under the corresponding link capacity; otherwise, we can always make it feasible by adjusting the data generating rate vector $\Lambda = [\lambda_i]$. Note that the data can be drained out from node or location i only if a sink s happens to be there, i.e., $\delta_{is}^k = 1$; otherwise that draining rate equals zero. We also assume that all nodes use an identical receiving power e^{R}, whereas the transmitting power e_i^{T} is set by a node i according to, for example, certain topology control mechanisms [18, 19]. Therefore, transmission and reception together contribute to the energy consumption of a node, and the energy consumed by other activities (e.g., data sensing) is considered as negligible. Finally, as a sink inherits the functions of a co-located node, we do not count the energy consumption of that node during the epoch when there is a sink co-located with it. We denote this phenomenon *substitution effect*, and we will discuss it in detail later on.

In the remainder of this section, we will first analyze the complexity of MNL and derive a duality theory to characterize the optimal solution of the problem. We will then use a simplified version of MNL to motivate a polynomial-time algorithm. Finally, we show that the polynomial-time algorithm can be used to approximate MNL with a provable ratio.

18.2.2 Complexity Analysis of MNL

Merging the explicit sink location constraints (18.4) and (18.5) into the conservation constraints for sensor nodes, we can re-formulate MNL into the Arc-Flow form:

$$\text{maximize} \quad T = \sum_k t_k \tag{18.8}$$

$$\text{subject to} \quad \sum_{(i,j),(j,i)\in\mathscr{E}} \left(q_{ij}^k - q_{ji}^k\right) \geq \lambda_i t_k \quad \forall k, i \notin \mathscr{L}_k \tag{18.9}$$

$$\sum_{k:i\notin\mathscr{L}_k} \left[\sum_{(i,j),(j,i)\in\mathscr{E}} \left(q_{ij}^k e_i^{\mathrm{T}} + q_{ji}^k e^{\mathrm{R}}\right) \right] \leq E_i \quad \forall i \tag{18.10}$$

$$t_k, q_{ij}^k \geq 0 \quad \forall i, j, k \tag{18.11}$$

where $q_{ij}^k = r_{ij}^k t_k$ represents the amount of data going from node i to node j during the kth epoch, and $\mathscr{L}_k \subset \mathscr{V}$ indicates the set of sink locations during that period.

This seemingly simple formulation hides the actual complexity of MNL: There could be a tremendous number of possible $\mathscr{L}_k$, because, to place m sinks on $|\mathscr{V}|$ possible positions, we have $\binom{|\mathscr{V}|}{m}$ choices of $\mathscr{L}_k$, which means the numbers of variables and constraints of MNL problem are both exponential in n. This motivates us to formally evaluate the complexity of MNL in the following. Let us re-formulate the MNL problem into a Path-Flow form:

$$\text{maximize} \quad T = \sum_k t_k \tag{18.12}$$

$$\text{subject to} \qquad \sum_s \sum_{p \in P_{is}^k} f(p) \ \geq \ \lambda_i t_k \quad \forall k, i \notin \mathscr{L}_k \tag{18.13}$$

$$\sum_{k:i \notin \mathscr{L}_k} \left[\sum_{p \in P_i^k} f(p) \left(e_i^{\mathrm{T}} + \mathrm{I}_{p \notin P_{is}^k, \forall s} \cdot e^{\mathrm{R}} \right) \right] \ \leq \ E_i \quad \forall i \tag{18.14}$$

$$t_k, f(p) \ \geq \ 0 \quad \forall k, p \tag{18.15}$$

where I_A is the indicator function of event A, p is a path between a node and a sink, and $f(p)$ is the flow going through that path. Furthermore, we denote by P_{is}^k the path set from node i to sink s and by P_i^k the set of paths going through node i, both in the kth epoch. Note that the data originated at a node may split into several fractions and flow to different sinks via various paths simultaneously, according to the multi-path formulation of the routing strategy. Since the primal formulation of MNL (18.13)–(18.14) is a linear program, the strong duality holds and hence we could instead investigate the dual problem of MNL shown as follows.

$$\text{minimize} \quad G(\mathbf{w}) = \sum_i E_i w(i) \tag{18.16}$$

$$\text{subject to} \qquad \sum_i \lambda_i W(i,k) \ \geq \ 1 \quad \forall k \tag{18.17}$$

$$\sum_{j \in p \in P_{is}^k, j \notin \mathscr{L}_k} w(j) \left(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}} \right) - W(i,k) \ \geq \ 0 \quad \forall i,k,s,p \tag{18.18}$$

$$w(i), W(i,k) \ \geq \ 0 \quad \forall i,k \tag{18.19}$$

where the $w(i)$ is the weight assigned to node i, representing the marginal cost of using an additional unit energy of node i; and $W(i,k)$ is the weight of a commodity, i.e., data flow going from node i to all possible destination sinks during epoch k; it indicates the marginal cost of rejecting a unit demand of the commodity. We can interpret the dual problem as follows: given the optimal solution $\hat{T}$ of the primal, if there existed a solution T' longer than $\hat{T}$, the benefit from the prolonged time period $T' - \hat{T}$ would not cover the cost of maintaining the network for that time period, as either performing the data routing (18.18) or not (18.17) would at least offset the benefit. As a result, the dual formulation implies that such T' should not exist.

As the dual objective is to minimize $G(\mathbf{w})$, implying that $w(i)$ is preferred to be as small as possible. However, we cannot make it as small as we want since it is bounded in (18.18) by $W(i,k)$, which is in turn constrained in (18.17). Thus we conduct the variable elimination by plugging (18.18) into (18.17). For an arbitrary vector $\mathbf{w} = [w(i)]$, the overall cost of rejecting a demand of the commodity from node i to any sink s, according to (18.18), can at most be the minimum transmission cost from i to the destination sinks. Therefore, we set

$$W(i,k) = \sum_{j \in \min\{p | p \in P_{is}^k, s \in \mathscr{S}\}, j \notin \mathscr{L}_k} w(j) \left(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}} \right) \tag{18.20}$$

Using (18.20) to eliminate $W(i,k)$ in (18.18), we get the combined new constraint as

$$\sum_i \lambda_i \left(\sum_{j \in \min\{p | p \in P_{is}^k, s \in \mathscr{S}\}, j \notin \mathscr{L}_k} w(j) \left(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}} \right) \right) \geq 1 \quad \forall k \qquad (18.21)$$

Actually, an arbitrary vector $\mathbf{w}$ may violate (18.21) and in turn (18.17) and hence be infeasible. But if we can find the most violated constraint and scale up $\mathbf{w}$ accordingly, we are always able to turn it into a feasible solution. More specifically, suppose there is an oracle $\rho(\mathbf{w})$, which is the minimum value of the LHS of (18.21) over k,

$$\rho(\mathbf{w}) = \min_k \left[\sum_i \lambda_i \left(\sum_{j \in \min\{p | p \in P_{is}^k, s \in \mathscr{S}\}, j \notin \mathscr{L}_k} w(j) \left(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}} \right) \right) \right] \qquad (18.22)$$

then testing $\rho(\mathbf{w}) < 1$ or not will suggest the feasibility of $\mathbf{w}$. This is called *separation oracle* in the terminology of linear programming [26]. If $\rho(\mathbf{w}) < 1$, we can scale up $\mathbf{w}$ and $W(i,k)$ by $\rho^{-1}(\mathbf{w})$ to make them feasible under the constraints. As a result, we transform the dual problem of MNL into an equivalent one, which is to find a vector $\mathbf{w}$ to minimize $\frac{G(\mathbf{w})}{\rho(\mathbf{w})}$. Unfortunately, the oracle is not easy to compute; we hereby show it is actually an NP-complete problem for on-graph mobility ($\mathscr{N} = \mathscr{V}$), which implies that it is NP-hard for off-graph mobility ($\mathscr{N} \subset \mathscr{V}$). Let $K = |\mathscr{S}| = m$, $\omega(i) = \lambda_i$, $\ell(j) = w(j) \left(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}} \right)$, and $d(i) = \sum_{j \in \min\{p | p \in P_{is}^k, s \in \mathscr{S}\}, j \notin \mathscr{L}_k} \ell(j)$, then the separation oracle is equivalent to the following decision problem:

INSTANCE: A graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, a weight assignment $\omega(i) : \mathscr{N} \to \mathbb{R}_0^+$, a length assignment $\ell(i) : \mathscr{N} \to \mathbb{R}_0^+$, positive integer $K \leq |\mathscr{N}|$, and positive real number B.

QUESTION: Is there a set $\mathscr{P}$ of K points on $\mathscr{G}$ such that, if $d(i)$ is the length of the shortest path from i to the closest point in $\mathscr{P}$, then $\sum_i \omega(i) \cdot d(i) \leq B$?

This is known as the p-median problem and is NP-complete [12]. As stated in [26] (**Theorem 3.3**), a linear programming problem is NP-hard if the corresponding separation oracle problem is NP-complete, hence we conclude that:

Proposition 1. *The MNL problem is NP-hard.*

18.2.3 Duality Theory and TMNTM

Before developing the algorithm to solve MNL, we will first harvest the benefit coming with the primal-dual interpretation provided in Section 18.2.2: it helps us to build the related duality theory, and it also allows us to address the TMNTM decision problem stated as follows:

> To Move or Not To Move (TMNTM): Is there a sink layout schedule $\{(sl_k, t_k)\}$ (sl_k is a vector of $[\delta_{is}^k]$) such that the lifetime $T = \sum_k t_k$ is longer than what is achieved by any fixed layout sl?

This was never fully addressed in the previous work such as [11,21].

We recapitulate the observation that we make on the dual problem of MNL in the following theorem:

Theorem 1 (Max-lifetime Min-potential ratio Theorem). *Given the lifetime maximization problem formulated in (18.12)–(18.15), the optimal lifetime $\hat{T}$ is such that*

$$\hat{T} = \min_{\mathbf{w}} \left[\frac{G(\mathbf{w})}{\rho(\mathbf{w})} \right]$$

where $G(\mathbf{w}) = \sum_i E_i w(i)$ is a linear combination of the energy reserves of all nodes with coefficients $w(i)$, and

$$\rho(\mathbf{w}) \equiv \min_k \rho_k(\mathbf{w}) = \min_k \left[\sum_i \lambda_i \left(\sum_{j \in \min\{p | p \in P_{is}^k, s \in \mathscr{S}\}, j \notin \mathscr{L}_k} w(j)(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}}) \right) \right]$$

is the minimum "potential" (computed as the sum of the minimum "cost", given $w(i)$, to route λ_i from node i to one of the m centers) achieved among all possible center layouts (or sink layouts) $\{\mathscr{L}_k\}$.

Likewise, for a fixed scheduling (or static sinks), the analogous theorem (first given in [29] and improved in [13]) is cited below:

Theorem 2 (Max-flow Min-distance ratio Theorem). *Given the maximization lifetime problem formulated in (18.12)–(18.15) but with a fixed schedule consisting of only one element (sl, t), the optimal lifetime $\hat{T}_{sl}$ is such that*

$$\hat{T}_{sl} = \min_{\mathbf{w}} \left[\frac{G(\mathbf{w})}{\rho_k(\mathbf{w})} \right]$$

where $G(\mathbf{w})$ and $\rho_k(\mathbf{w})$ are defined in the previous theorem, and the center layout is defined by sl.

One can easily see that **Theorem 1** is a non-trivial extension of **Theorem 2**, which is in turn extended from the **Max-flow Min-cut** theorem of Ford and Fulkerson [10] for a single *s-t* flow. Equipped with these two theorems, we are now ready to answer TMNTM.

Proposition 2. *For on-graph mobility, $\hat{T} > \hat{T}_{fs}$, where $\hat{T}_{fs} = \max_{sl} \hat{T}_{sl}$. Literally, the answer to the TMNTM decision problem is positive.*

Proof: Assume that $\hat{T}_{sl} > 0$ is the optimal solution for a certain sl, and $\mathbf{w}^*_{sl}$ is the corresponding weight assignment. By plugging $\mathbf{w}^*_{sl}$ into the dual problem of MNL (18.16)–(18.19), we can always identify a violated constraint with the oracle that computes $\min_k \rho_k(\mathbf{w}^*_{sl})$. For instance, assume that one of the sinks is co-located with i and its most loaded neighbor is j. We know that (18.14) is active for j; otherwise it contradicts the optimality of $\hat{T}_{sl}$. Applying complementary slackness, we have $\rho_i(\mathbf{w}^*_{sl}) = 1$ (by $\hat{T}_{sl} > 0$), $w^*_{sl}(i) = 0$ (by the fact that (18.14) is inactive for i due to the substitution effect defined in Section 18.2.1), and $w^*_{sl}(j) > 0$ (by the fact that (18.14) is active for j and j is the bottleneck of all the paths passing through it). The potential $\rho_j(\mathbf{w}^*_{sl})$ is bound to be less than 1, because, by moving the sink from i to j, we shorten the length of some paths by $w^*_{sl}(j)$ without increasing the length of other paths going through i. Therefore, we identify that $\mathbf{w}^*_{sl}$, as the dual solution, is infeasible. Consequently, according to the principle of *certificate of optimality*, we know that $\hat{T}_{sl}$, as the primal solution for the fixed schedule case, is not optimal for the MNL problem and thus $\hat{T} > \hat{T}_{sl}$. Let $\hat{T}_{fs} = \max_{sl} \hat{T}_{sl}$, we also have $\hat{T} > \hat{T}_{fs}$. Q.E.D. ☒

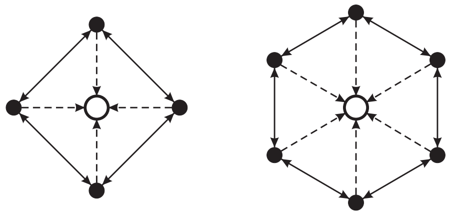
Note that this proof implicitly assumes that the minimum potential $\min_k \rho_k(\mathbf{w}^*_{sl})$ and the maximum lifetime $\hat{T}_{sl}$ for a fixed sink layout is computable. As we show in Section 18.2.2, however, the separation oracle problem is NP-complete. Also, results in [4] suggest that computing $\hat{T}_{sl}$ is NP-hard. Therefore, **Proposition 2** only gives a qualitative comparison rather than a quantitative one.

Interestingly, the proof of **Proposition 2** stresses the importance of a hidden factor behind the evident load-balancing effect, namely the substitution effect. Recall from the model description in Section 18.2.1, we assume that whenever a sink is co-located with a node, it inherits all the functions of that node, i.e., it takes the place of that node in the network and hence saves its energy consumption. While the load balancing effect is the driving force behind a significant lifetime improvement, the substitution effect, as presented in the above proof, makes moving sinks superior to keeping them static if the sinks are constrained to be on-graph. In Section 18.4, we will discuss an extension that fully exerts the substitution effect to improve lifetime.

It is also worth noting that **Proposition 2** holds only for on-graph mobility. In Figure 18.3, we give two examples showing that a static sink layout is already the optimal choice for certain network topologies. Fortunately, we might not have such pathological scenarios in practice. Even if such a case occurs, the optimal (off-graph) sink location might not be available (we refer to [23] for a practical example we have experienced). All the examples we give in Section 18.2.5 confirm that moving the sink, no matter on-graph or off-graph, is always superior to keeping it static.

18.2.4 A Primal-Dual Algorithm to Solve MNL

It is straightforward to see that, if there is only one mobile sink, the MNL problem is solvable in polynomial time, because the separation oracle is a P problem. However, directly solving it is practically ineffective on all but very small scale problems



(a) 4-node ring (b) 6-node ring

Fig. 18.3 Two examples to show that **Proposition 2** might not hold if sink locations can be off-graph. The solid lines represent the original links of the ring networks, and the dash lines represent the new links introduced by an off-graph sink (located at the optimal position). It is straightforward to see that sink mobility does not help in these cases.

(similar to the case of the concurrent flow problem [29]). In addition, common techniques such as the interior point or simplex algorithms cannot be extended to address the MNL problem involving multiple mobile sinks. In Section 18.2.4.1, we will discuss a primal-dual algorithm that solves the MNL problem with a single mobile sink efficiently. Moreover, we will extend the algorithm to approximate the solution of the general MNL in Section 18.2.4.2.

18.2.4.1 MNL with a Single Mobile Sink (MNL–SMS)

Differing from the usual network flow problems that involve multiple *s*-*t* flows, MNL–MMS combines two types of problems, namely *maximum concurrent flow* problem and *maximum multicommodity flow* problem. It is a maximum concurrent flow problem because for each demand λ_i associated with node i, we want to find the maximum multiplier T. Meanwhile it is also a maximum multicommodity flow problem, as for each demand t_k, our objective is to maximize $\sum_k t_k$ without caring the particular value of individual t_k. Therefore, we need to design a new algorithm to address it, and our design is based on the proposal of Garg and Könemann [13].

For the case of single mobile sink, $\mathscr{S}$ includes only one sink referred to as s. Clearly, s can choose its location from those indicated by $\mathscr{V}$, implying that $\mathscr{L}_k$ has $|\mathscr{V}|$ possibilities. Therefore, the dimension of $[t_k]$ is at most $|\mathscr{V}|$, and thus we further simplify the problem by assuming that s is coincident with location $k \in \mathscr{V}$ during the kth epoch. Due to the symmetry of the MNL–SMS problem, the order of the sink locations does not affect the optimal solution. In other words, this assumption leads to the solution applicable to a general case without location ordering. We omit the formulation of the MNL–SMS problem, as it is indeed the same as that of the general MNL case (18.12)–(18.19) under the condition that $\mathscr{L}_k = \{k\}$. Similarly, let $W(i,k) = \sum_{j\in\min\{p|p\in P_{ik}^k\},j\neq k} w(j)\left(e_j^{\mathrm{T}} + \mathbf{I}_{j\neq i}\cdot e^{\mathrm{R}}\right)$, we get the separation oracle for MNL–SMS as follows.

$$\rho(\mathbf{w}) = \min_k \rho_k(\mathbf{w}) = \min_k \left[\sum_i \lambda_i \left(\sum_{j \in \min\{p|p \in P_{ik}^k\}, j \neq k} w(j) \left(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}} \right) \right) \right] \quad (18.23)$$

Here the oracle computes $\min_k \rho_k(\mathbf{w})$ by first using the Floyd-Warshall algorithm [9] to compute all-pairs shortest path with a time complexity $\Theta(n^3)$. Paths ended at a common node are then clustered into groups, and the algorithm searches for the best "median" k that achieves $\min_k \rho_k(\mathbf{w})$. As the complexity of clustering and searching is negligible compared to the Floyd-Warshall algorithm, this oracle has a complexity of $\Theta(n^3)$. Now, we are ready to derive the algorithm for solving the MNL–SMS problem. The pseudo-code is provided, where superscript $^+$ indicates the updated value, $\delta = (1+\varepsilon)[(1+\varepsilon)n]^{-1/\varepsilon}$, and ε is the required error bound.

Algorithm 14 MNL_ALGO

```
Input: $\mathscr{N}, \mathscr{E}, \Lambda = [\lambda_i], \mathbf{E} = [E_i], \mathbf{e} = [e_i^{\mathrm{T}}], e^{\mathrm{R}}$, and initial weight assignment $\mathbf{w} = [\delta/E_i], \forall i \in \mathscr{N}$
repeat
    Identify the most violated element by the oracle: $k^+ = \arg\min_k \rho_k(\mathbf{w})$;
    Increase the kth epoch by 1 unit: $t_k^+ = t_k + 1$;
    Follow the shortest path $p_{ik}^+$ (suggested by the oracle) from each node $i$ to the sink, route $\lambda_i$ units of commodity along that paths, and update the flow through node $j \in p_{ik}^+$:
    $f^+(j) = f(j) + \lambda_i$;
    Update the weight of node $i$: $w^+(i) = w(i)\left(1 + \varepsilon f^+(i)\left(e_i^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}}\right)/E_i\right)$;
    Calculate the dual objective:
    $G(\mathbf{w}^+) = \sum_i E_i w^+(i) = G(\mathbf{w}) + \varepsilon \sum_i w(i) f^+(i)\left(e_i^{\mathrm{R}} + e^{\mathrm{T}}\right) = G(\mathbf{w}) + \varepsilon \cdot \rho(\mathbf{w})$;
until $G(\mathbf{w}^+) \geq 1$
return maximum network lifetime: $\tilde{T} = \log_{1+\varepsilon}^{-1} \frac{1+\varepsilon}{\delta} \sum_k t_k^+$
```

The algorithm proceeds in iterations. In each iteration, the oracle identifies k^+ that gives the minimum "potential", and it also suggests the paths from each node to that sink. Then the weight assignments $\mathbf{w}$, time schedule t_k, flow assignments $[f(i)]$, and dual objective $G(\mathbf{w})$ are all updated accordingly; they will serve as the inputs to the next round of iteration. The algorithm runs until the dual objective exceeds the threshold 1. We show the correctness and the time complexity of this algorithm by the following proposition.

Proposition 3. *Given $\sum_i \lambda_i \le E_i/(e_i^{\mathrm{T}} + \mathrm{I}_{j\neq i} \cdot e^{\mathrm{R}}), \forall i$,*[4] ***MNL_ALGO*** *computes a $(1-\varepsilon)^{-2}$-approximation to the MNL–SMS problem in time $\Theta(n \log n) \cdot T_{\mathrm{oracle}}$, where T_{oracle} is the time complexity for the oracle to compute $\min_k \rho_k(\mathbf{w})$.*

Proof: Let the dual optimal value be β. According to the 6th step, we have at the end of each iteration

$$G(\mathbf{w}^+) \le G(\mathbf{w})(1+\varepsilon/\beta) \le G(\mathbf{w})e^{\varepsilon/\beta}$$

where $\frac{G(\mathbf{w})}{\rho(\mathbf{w})} \ge \beta$ accounts for the first inequality. Suppose that $G(\mathbf{w}^+) \ge 1$ at the end of the tth iteration and given initially $G(\mathbf{w}) = n\delta$; we have

$$1 \le G(\mathbf{w}^+) \le n\delta e^{t\varepsilon/\beta} \Rightarrow \frac{\beta}{t} \le \frac{\varepsilon}{\ln(n\delta)^{-1}}$$

The ratio between the dual and primal solutions is given by:

$$\gamma = \frac{\beta}{\tilde{T}} = \frac{\beta}{t}\log_{1+\varepsilon}\frac{1+\varepsilon}{\delta} \le \frac{\varepsilon \log_{1+\varepsilon}\frac{1+\varepsilon}{\delta}}{\ln(n\delta)^{-1}} = \frac{\varepsilon}{\ln(1+\varepsilon)} \cdot \frac{\ln\frac{1+\varepsilon}{\delta}}{\ln(n\delta)^{-1}}$$

As $\delta = (1+\varepsilon)[(1+\varepsilon)n]^{-1/\varepsilon}$, we have:

$$\gamma \le \frac{\varepsilon}{(1-\varepsilon)\ln(1+\varepsilon)} \le \frac{\varepsilon}{(1-\varepsilon)(\varepsilon-\varepsilon^2/2)} < (1-\varepsilon)^{-2}$$

As the maximal lifetime $\hat{T} = \beta$ due to strong duality, we have $\tilde{T} = t \cdot \log_{1+\varepsilon}^{-1}\frac{1+\varepsilon}{\delta} > (1-\varepsilon)^2\hat{T} \ge (1-2\varepsilon)\hat{T}$. Q.E.D. ☒
We only sketch the proof here but omit the detailed proof for feasibility and time complexity. Interested readers are referred to [22] for details.

18.2.4.2 Approximation Algorithm for General MNL

As we pointed out in Section 18.2.2, if we had an oracle to solve the p-median problem, then we would be able to solve the general MNL problem under the on-graph mobility scenario, because the dual LP (18.17)–(18.18) serves as a polynomial-time reduction from the separation oracle (a p-median problem) to the general MNL problem. For the off-graph mobility, we could extend the graph $\mathscr{G}$ by assigning zero weight to every vertex in $\mathscr{V} \backslash \mathscr{N}$ and connecting it to every other vertex through a directed edge. This extension allows us to run the p-median solver on $\mathscr{G}$ without being interfered by the vertices representing those potential off-graph sink locations.

[4] This assumption is reasonable because each sensor node should be equipped with an energy source that is at least enough for the node to forward data for all nodes in one time unit. Otherwise if a node $i : E_i/(e_i^{\mathrm{T}} + \mathrm{I}_{j\neq i} \cdot e^{\mathrm{R}}) < \sum_i \lambda_i$ is deployed close to a static sink (assuming a randomly deployed WSN), the network lifetime can be even less than one time unit. In addition, it can be proved that an approximation ratio of $(1-\varepsilon)^{-3}$ is still achievable without this assumption.

However, unless P = NP, no efficient p-median solver would exist. Although there exist approximation algorithms for the p-median problem, there is no guarantee that the dual LP may accommodate an approximate oracle. Fortunately, the algorithm we proposed in Section 18.2.4.1, MNL_ALGO, does accommodate an approximate oracle, with a slight change in the 2nd step as $\mathscr{L}_{k^+} : k^+ = \arg\min_k \rho_k(\mathbf{w})$. Therefore, combining any PTAS for p-median with MNL_ALGO will yield a PTAS for MNL, as shown by the following proposition.

Proposition 4. *If the p-median oracle can be approximated within a ratio of $\alpha > 1$ (i.e., the oracle has an α-approximation), then **MNL_ALGO** along with this oracle provides an $\alpha \cdot (1-\varepsilon)^{-2}$-approximation to the general MNL problem.*

Proof: The main difference between MNL and MNL–SMS is that, instead of having an oracle that returns the exact $\rho(\mathbf{w})$, we only have an α-approximation of the oracle. It means that the oracle always returns $\tilde{\rho}(\mathbf{w}) \leq \alpha\rho(\mathbf{w})$ with $\alpha > 1$. Since we have $\frac{G(\mathbf{w})}{\rho(\mathbf{w})} > \beta$ for MNL–SMS, we now have $\frac{G(\mathbf{w})}{\tilde{\rho}(\mathbf{w})} > \tilde{\beta}$, where $\tilde{\beta} = \frac{\beta}{\alpha}$. Therefore, we basically follow the line of proving **Proposition 3** but replacing β by $\tilde{\beta}$, and we will finally have $\tilde{T} > \frac{(1-\varepsilon)^2}{\alpha}\hat{T}$. Q.E.D. ☒

In fact, Arya et al. [2] gave a $(3+\omega)$-approximation algorithm for the p-median problem. Therefore, we have an algorithm to approximate the general MNL problem with a factor of $(3+\omega)(1-\varepsilon)^{-2}$.

18.2.5 Numerical Results

In this section, we show the quantitative improvement on lifetime by using a mobile sink for WSNs. We always assign a homogeneous λ, e^{T}, e^{R} and E to all nodes in order to facilitate the interpretation of the results. Without loss of generality, we assume $\lambda = 1$, $e^{\mathrm{T}} = e^{\mathrm{R}} = 0.5$, and $E = |\mathscr{N}| = n$. We set $\varepsilon = 0.01$. Here we only investigate two metrics, namely lifetime and pause time distribution, and refer to [23] for the evaluation of other metrics. We only focus on a single mobile sink as we surely have optimal solution for this case. Note that the pause time distribution given by an approximate solution may differ a lot from the optimal solution. The numerical results are obtained for networks with regular and arbitrary topologies. We consider both on-graph and off-graph sink mobilities and compare them for all networks. All these problems are solved using the primal-dual algorithm presented in Section 18.2.4.1.

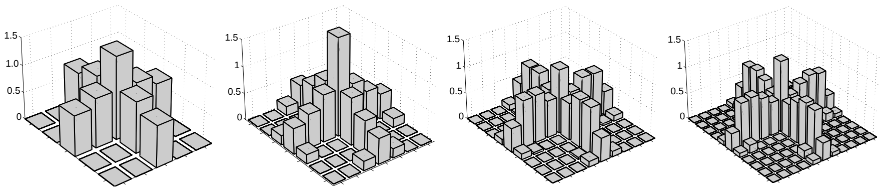
18.2.5.1 Grid Network

For grid networks on $\sqrt{n} \times \sqrt{n}$ lattices, the maximum achievable lifetime by a static sink is $n/(\lceil(n-5)/4\rceil+1)$, because the lifetime is maximized if the forwarding load is balanced among the 4 neighbors of the sink. This lifetime can be obtained by putting the sink at the network center (if $\sqrt{n}$ is odd) or at any of the four nodes close to the center (if $\sqrt{n}$ is even). While this lifetime is converging to 4 when $n \to \infty$, the lifetime achieved by a mobile sink increases dramatically with the network size (Table 18.2). For small-size networks (e.g., $|\mathcal{V}| = 9$ in Table 18.2), the substitution effect dominates the load balancing effect, so the relative improvement is small. With an increasing network size, the number of alternative paths between an *s*-*t* pair is also increasing. Consequently, the load balancing effect becomes increasingly remarkable and thus produces significant improvements on the lifetime.

	Network Lifetime T		
$\lvert\mathcal{V}\rvert$	mobile sink	static sink (optimal)	improvement (%)
9	5.331	4.500	18.47
16	6.509	4.000	62.72
49	11.09	4.084	171.7
121	17.07	4.033	323.2
144	18.71	4.000	376.8
225	23.29	4.018	479.7
289	26.33	4.014	555.9

Table 18.2 Comparing the achievable lifetime between using a mobile sink and a static sink in grid networks of different size.

We illustrate the pause time distribution in four networks in Figure 18.4. Our observation is that the sink tends to move toward the periphery of a network with an increasing n. The intuition is that, for a 3D grid on a sphere, the sink should pause everywhere with the same time period. Therefore, the pause times spread out when the network grows in size and thus appears more and more like a sphere grid to the nodes close to the center. This observation also corroborates the result in [21]:



(a) 25 nodes (b) 49 nodes (c) 81 nodes (d) 121 nodes

Fig. 18.4 Pause time distribution of a mobile sink in grid networks. The x and y axes indicate the location of the nodes, and the z-axis represents the pause time.

the network periphery, as a sink moving trace, is asymptotically optimal. Note that we investigate in [21] the asymptotical case where the node density is large enough to make the necessary radio ranges infinitely small. In that case, the shortest paths between any *s*-*t* pair happen to be straight lines.

We also consider the off-graph sink mobility, where the sink can also move to the vertices of another grid that is complementary to the original network, as shown in Figure 18.5 (a). The results show that, for all the networks shown in Table 18.2, off-graph mobility **does not** further improve the lifetime compared with on-graph mobility. In fact, even the pause time distribution remains to be the same after relaxing the on-graph constraint on the sink mobility. This interesting observation shows that, for networks that are well connected, on-graph sink mobility is sufficient to achieve the maximum lifetime.

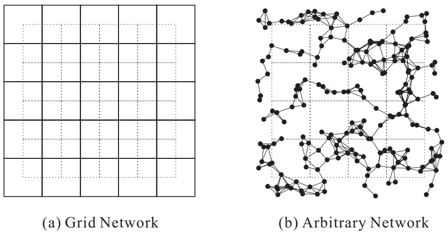


Fig. 18.5 Illustrations of off-graph sink mobility. The original network is represented by (a) the solid grid and (b) the black nodes. The sink, in addition to be able to move on-graph, may also move to locations represented by the vertices of the dash grid.

18.2.5.2 Arbitrary Network

We also perform experiments on arbitrary networks (nodes uniformly distributed within a square). Figure 18.5 (b) shows such a network and the possible off-graph sink locations (represented by the dash grid). We consider both 100-node and 200-node networks with a 10×10 off-graph grid, and each with 30 trials. In Figure 18.6, we compare the maximum lifetime achieved in four cases, namely, static on-graph sink, mobile on-graph sink, static off-graph sink, and mobile off-graph sink. We use the boxplot to summarize the results we have obtained, in which each case is depicted by five quantities: lower quartile (25%), median, upper quartile (75%), and the two extreme observations. It can immediately be seen that moving the sink always improves the lifetime compared with fixing it, whether on-graph or off-graph. Also, it is not a surprise that allowing off-graph sink locations (for both mobile and static sinks) outperforms constraining those locations on-graph, this is, of course, at a cost of higher complexity in solving the problem. Fortunately, our algorithm handles this complexity very well given a reasonable number of the off-graph locations.

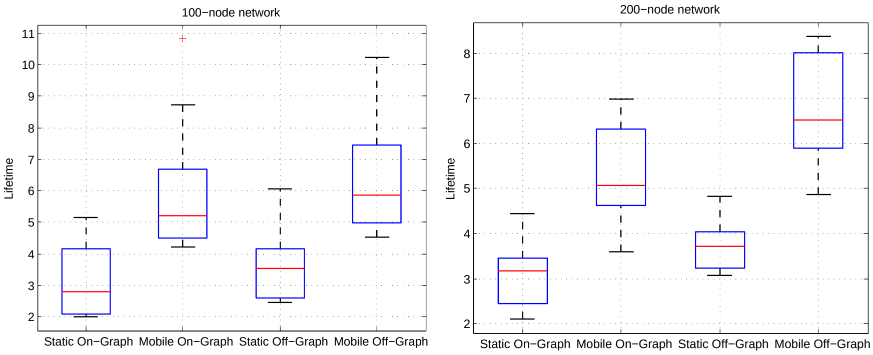
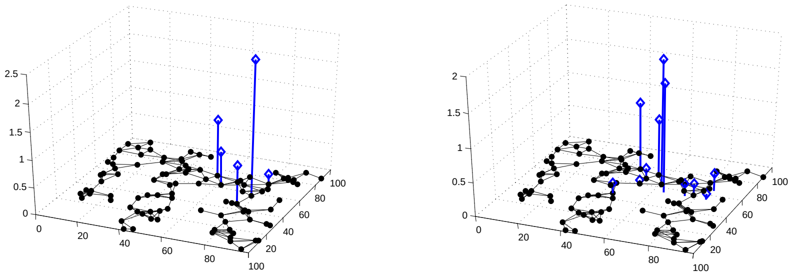


Fig. 18.6 Comparing different sink behaviors in arbitrary networks with 100 and 200 nodes.

It is also interesting to look at the pause time distribution, Figure 18.7 illustrates one such case (other cases exhibit the same trend). A direct observation is that the sink tends to pause at the nodes whose degrees are high (for on-graph locations) and at the off-graph locations around which the node density is high. This is intuitive because the more neighbors a node or a location has, the more a balanced load can be achieved by co-locating the sink with it. A slightly surprising observation is that not many locations are chosen by the optimal sink mobility: only 5 positions for on-graph mobility and 10 positions for off-graph mobility. This is quite different from the grid network. In fact, most arbitrarily deployed networks have a topology close to a tree rather than a mesh. It is quite intuitive to see that the sink mobility will concentrate around the root of a balanced tree.



(a) On-graph mobility

(b) Off-graph mobility

Fig. 18.7 Pause time distribution of a mobile sink in an arbitrary network. The x and y axes indicate the location of the nodes, and the z-axis represents the pause time.

18.2.6 Summary

By far, we have formulated the optimization problem for maximizing network lifetime using mobile sinks. We have analyzed the complexity of the problem and characterized the optimal solution through duality theory. Finally, we have developed an algorithm to solve the problem with one mobile sink and to approximate the solution if there are multiple sinks. In the remainder of this chapter, we will present several extensions to the optimization framework defined in this section. In Section 18.3, we explain how to obtain optimal solution if we do not put any constraints on the sink locations and hence allow them to be chosen within, for example, a 2D Euclidean space. Recall that the reason accounting for the lifetime improvement in the mobile sink approach is twofold, namely load balancing effect and substitution effect, while the former is the dominating factor. In Section 18.4, we will look into another extension where the substitution effect will be fully utilized, which is called the mobile node approach. Finally, we will show, in Section 18.5, that certain slight changes of the terminology allow us to model the mobile relay approach. Also, we will describe a variance of the problem formulation, which, by simplifying several assumptions of the model, may actually lead to more efficient solutions.

18.3 Balancing Traffic Load with Mobile Sinks: The Case of Unconstrained Mobility

Although it seems that the problem formulation we present in Section 18.2 is limited to the case where a finite set of potential sink locations is provided, we show in this section that it is not true: given a continuous space (e.g., a Euclidean one), we only need to search among a finite number of locations in order to obtain the optimal solution or to closely approximate the optimal solution. In the following, we first discuss, in Section 18.3.1, the case where the transmission energy is associated with a node (as assumed in Section 18.2.1); the approach we present in Section 18.2 still yields the optimal solution even after relaxing the constraints on the potential locations. In Section 18.3.2, we slightly change one of the assumptions by associating the transmission cost with a link. The problem does become harder under this circumstance, but we will describe an extension to the approximation scheme presented in [30],[5] which may give a solution that is arbitrarily close to the optimal one for a single mobile sink and provide good approximation for multiple mobile sinks, at an increasingly (sometimes drastically) computational complexity.

[5] The original work by Shi and Hou [30] is only designed for a single mobile sink. In this section, we extend their approach by combining it with MNL_ALGO presented in Section 18.2.4 to deal with multiple mobile sinks.

18.3.1 Node-Associated Transmission Energy

If the transmission energy is associated with individual nodes, we could always come up with a virtual circle around a certain node, such that a link can be established iff the destination node falls within this circle. As shown in Figure 18.8, the potential locations for the mobile sinks are constrained by these circles. In particular, these circles partition the continuous space into many subareas, whose boundaries are one or more portions of certain circles. Most importantly, we only need one sink location to represent each of these subareas (which can be arbitrarily chosen within a subarea), as the resulting network topology graph remains the same if we move the sink within a subarea. In a nutshell, given a WSN along with its transmission energy assignment and a specific way of defining the mapping **c**, we can compute the radius of the circle for each node. Drawing all these circles will give us a finite set of subareas and hence a finite set $\mathscr{A}$ of potential sink locations. Let $\mathscr{V} = \mathscr{N} \cup \mathscr{A}$, we are back to the off-graph mobility problem addressed in Section 18.2.

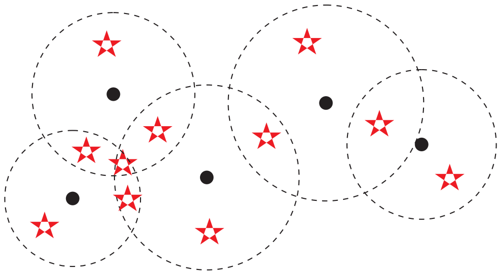


Fig. 18.8 All possible off-graph sink locations (represented by the stars) given a certain assignment of the node-associated transmission energy.

We consider the model that associates the transmission energy with individual nodes to be more realistic than the link-associated version, because, although nodes may have a tunable transmission power, it is not cost-effective to dynamically tune the power for destinations at different distances. In addition, tuning transmission power according to transmission distances is not always feasible either, as a node might not know the distances. Therefore, a reasonable scenario, in our opinion, is that each node sets up a transmission power according to certain topology control mechanisms [18, 19] at the network initialization phase and fixes this power until some topology changes happen. However, since the link-associated model is also popularly used, we will treat the MNL problem under that model in Section 18.3.2.

18.3.2 Link-Associated Transmission Energy

Under on-graph mobility, it is pretty straightforward to switch from the node-associated model to the link-associated version: we simply need to replace e_i^{T} with $e_{i,j}^{\mathrm{T}}$.[6] However, allowing off-graph mobility drastically increases the complexity of the problem. As the (link) transmission energy can be tuned freely, any location in a continuous space is virtually unique as it may yield a different transmission energy from some sensor node to that location.

Whereas it is true that finding the optimal solution will incur a tremendous complexity, obtaining good approximation is still possible [30]. To better illustrate the idea, we first reformulate the MNL problem by modifying the constraint (18.14) as follows (constraint (18.13) is not affected by switching to the link-associated version).

$$\sum_{k:i\notin\mathscr{L}_k}\sum_{p\in P_i^k} f(p)\left(\mathrm{I}_{(i,j)\in p,j\notin\mathscr{L}_k}\cdot e_{i,j}^{\mathrm{T}}+\mathrm{I}_{(i,j)\in p,j\in\mathscr{L}_k}\cdot e_{i,j}^{\mathrm{T}}+\mathrm{I}_{p\notin P_{is}^k,\forall s}\cdot e^{\mathrm{R}}\right)\le E_i \quad \forall i \tag{18.24}$$

Note that, apart from replacing e_i^{T} with $e_{i,j}^{\mathrm{T}}$, we also split the first term in the parentheses into two parts: while the first part is independent of the sink locations in the current epoch, the second part is not. We call this problem *maximizing network lifetime with link associated transmission energy* (MNL–LATE). In theory, the possible choices of $e_{i,j}^{\mathrm{T}}$ for arbitrary $i\in\mathscr{N}$ and $j\in\mathscr{L}_k$ are infinite, this makes the searching for an optimal solution enormously hard. Fortunately, it is possible to develop a $(1+\kappa)$-approximation (where κ is the error bound) for a single mobile sink or a $(1+\kappa)(3+\omega)(1-\varepsilon)^{-2}$-approximation for multiple mobile sinks based on the algorithm given in Section 18.2. In the following, we first briefly present the basic idea of the approach in Section 18.3.2.1, then we describe the algorithm along with propositions that strictly prove its correctness in Section 18.3.2.2.

18.3.2.1 Parameterization Using Geometric Sequence

We first parameterize MNL–LATE by taking $e_{i,j}^{\mathrm{T}}$ in the second term of (18.24) as the parameters. Instead of letting each $e_{i,j}^{\mathrm{T}}$ to be any possible real number, we limit the choice to be a sequence of numbers $\mathbf{e}_{i,j}=\{e_{i,j;0}^{\mathrm{T}},e_{i,j;1}^{\mathrm{T}},\cdots,e_{i,j;h}^{\mathrm{T}},\cdots\},\forall i\in\mathscr{N},j\in\mathscr{S}$, where $e_{i,j;0}^{\mathrm{T}}=a$ and $e_{i,j;h}^{\mathrm{T}}=a(1+\kappa)^h$. Obviously, this sequence is a geometric sequence with factor a and common ratio $(1+\kappa)$. Assume that there is a set $\mathscr{A}$ of the off-graph locations such that, for each $j\in\mathscr{A}$, we have $e_{i,j}^{\mathrm{T}}\in\mathbf{e}_{i,j},\forall i\in\mathscr{N}$ and for each $j:e_{i,j}^{\mathrm{T}}\in\mathbf{e}_{i,j},i\in\mathscr{N}$, we have $j\in\mathscr{A}$. In other words, $\mathscr{A}$ enumerates any possible off-graph location j whose incurred transmission energy from any node i to itself is given in $\mathbf{e}_{i,j}$. If we could also make $\mathscr{A}$ to be finite, then letting $\mathscr{V}=\mathscr{N}\cup\mathscr{A}$ would

[6] A byproduct of this change is that the cost assignment $\mathbf{c}$ is not needed anymore, as any link (i,j) is feasible given a sufficiently high transmission energy $e_{i,j}^{\mathrm{T}}$.

bring us back to the off-graph mobility problem addressed in Section 18.2. We call this problem MNL–DLATE. Of course, the solution to MNL–DLATE may not be optimal to MNL–LATE, as $\mathscr{A}$ does not include all possible off-graph locations. However, it is intuitive to see that the solution should not be far from optimal if the granularity $(1+\kappa)$ we use to discretize the continuous space is sufficiently small.

It is worth noting that the aforementioned discrete parametrization applies to many optimization problems, especially those with linear constraints. In Section 18.3.2.2, we first show that this discretizaton is possible within a Euclidean space and $\mathscr{A}$ can indeed be made finite. Then we show that the solution to MNL–DLATE is a good approximation to that of MNL–LATE and we also give the approximation ratio. We note that, if the error bound κ, thus the common ratio $(1+\kappa)$, is small , the cardinality of $\mathscr{A}$ (hence that of $\mathscr{V}$) becomes huge, which makes the problem much harder to solve than the node-associated version presented in Section 18.3.1.

18.3.2.2 Algorithm Implementation

Based on what is described in Section 18.3.2.1, a practical algorithm to approximate the solution of MNL–LATE needs three components: 1) A finite set $\mathscr{A}$ defined by a set of geometric sequences $\{\mathbf{e}_{i,j}\}_{i\in\mathscr{N},j\in\mathscr{A}}$, 2) The algorithm MNL_ALGO we have presented in Section 18.2.4, and 3) A proof to show the approximation ratio. Since the second component is ready, we present in the following the procedures that justify both the first and the third components, assuming a Euclidean space and a link transmission power assignment that is strictly increasing in the Euclidean distance from the source to the destination.

As we assume that $e^{\mathrm{T}}_{i,j}$ is an increasing function of $d(i,j)$, the Euclidean distance between node i and node j, the sequence $\mathbf{e}^{\mathrm{T}}_{i,j}$ can be generated by properly drawing concentric circles around i and let $e^{\mathrm{T}}_{i,j;h}$ be the transmission energy to reach from i to the h-th circle, as shown in Figure 18.9. Although to reach any location between the $(h-1)$-th and the h-th circles only requires a transmission energy in the interval $(e^{\mathrm{T}}_{i,j;h-1}, e^{\mathrm{T}}_{i,j;h})$, we deliberately amplify it to $e^{\mathrm{T}}_{i,j;h}$. According to (18.24), such an amplification tightens the constraint. Therefore, the optimal solution to this special

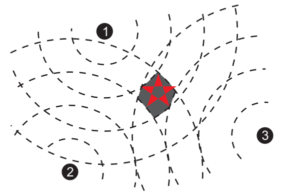


Fig. 18.9 Generating geometric sequences using concentric circles. According to our definition of MNL-DALATE, any potential sink location j within the shaded region is assigned with a transmission energy vector $(e^{\mathrm{T}}_{1,j;2}, e^{\mathrm{T}}_{2,j;3}, e^{\mathrm{T}}_{3,j;3})$.

version of MNL–DLATE, called MNL–DALATE, is bounded to be feasible under the general MNL–DLATE where each location is associated with the actual transmission energy derived from the exact distance. Consequently, the optimal solution to MNL–DLATE is no smaller than that of MNL–DALATE. This is summarized by the following proposition.

Proposition 5. *If the solution to MNL–DALATE is an α-approximation of the solution to MNL-LATE, where $\alpha > 1$, the approximation ratio given by the solution to MNL-DLATE is at most α.*

The finiteness of $\mathscr{A}$ is the direct consequence of a bounded search region and of the increasing radius of the circles that generate the geometric sequences, and the boundedness of the search region is shown by the following proposition.

Proposition 6. *An optimal solution of MNL–LATE always has its sink locations constrained within the smallest enclosing disk (SED) that contains $\mathscr{N}$.*

Proof: Assume that the statement in the proposition is not true, then there exists at least one optimal sink location that is outside of the SED. We first draw a line segment from this location to the center of the SED, which yields a (point) intersection with the boundary of the SED. Then we mirror the location with respect to the tangent (of the boundary of the SED) that goes through the intersection point. It can be easily seen that this new location is superior to the old one, as it is closer to all the nodes in terms of Euclidean distance. This outcome contradicts the optimality of the assumed "out of SED" location, and hence proves the proposition. Q.E.D. ⊠

Now, the algorithm construction becomes pretty clear; we illustrate the algorithm MNL–LATE_ALGO as follows. Note that, if there is only one mobile sink, we can also replace the 4th step by an LP, which removes the (albeit negligible) approximation ratio $(1-\varepsilon)^{-2}$. The performance of MNL–LATE_ALGO is shown by the following propositions.

```
Algorithm 15 MNL–LATE_ALGO
Input: $\mathscr{N}, \Lambda, \mathbf{E}$
1: Compute the SED that contains $\mathscr{N}$, and generate $e^{\mathrm{T}}_{i,j}$ for all $i, j \in \mathscr{N}$;
2: Generate the geometric sequence $\mathbf{e}^{\mathrm{T}}_{i,j}$ for all $i \in \mathscr{N}$ using the sequence of concentric circles, and produce the location set $\mathscr{A}$, within the boundary of SED, according to the subareas demarcated by these circles;
3: Applying the amplification rule to generate $e^{\mathrm{T}}_{i,j}$ for all $i \in \mathscr{N}, j \in \mathscr{A}$;
4: Call MNL_ALGO upon the instance $(\mathscr{V}, \mathscr{E}, \Lambda, \mathbf{E}, \mathbf{e})$, where $\mathscr{V} = \mathscr{N} \cup \mathscr{A}$, $\mathscr{E}$ includes all the edges among $\mathscr{N}$ and those from $\mathscr{N}$ to $\mathscr{A}$ and $\mathbf{e} = [e^{\mathrm{T}}_{i,j}]$, and get the return value $\tilde{T}$.
5: return maximum network lifetime: $\tilde{T}$
```

Proposition 7. *With a single mobile sink, the value returned by MNL–LATE_ALGO, $\tilde{T}$, is a $(1+\kappa)$-approximation to the optimal value $\hat{T}$ of MNL–LATE, in other words, $\tilde{T} \geq (1+\kappa)^{-1}\hat{T}$.*

Proof: The proof goes very similar to the discussion made before **Proposition 5**. We first notice that any location that belongs to an optimal solution of MNL–LATE must fall into one of the subareas demarcated by the concentric circles and the SED. Then it is straightforward to see that the amplification rule simply exaggerates the transmission energy incurred by an optimal location, which, in effect, tightens the constraint (18.24). The other important fact is that the amplification is bounded: it is at most $(1+\kappa)$ to the value that would have been incurred by an optimal location. Given $\hat{T}$ as the optimal value of MNL–LATE, we can scale it down by a factor $(1+\kappa)$, which effectively scales down all $f(p)$s in (18.24), and makes it a feasible solution of MNL–DALATE. As $\tilde{T}$ is the optimal value of MNL–DALATE (assume the use of LP for the 4th step), we have $\tilde{T} \geq (1+\kappa)^{-1}\hat{T}$. Q.E.D. ⊠

Similar arguments lead us to the approximation ratio for multiple mobile sinks.

Proposition 8. *With multiple mobile sink, $\tilde{T}$ is a $(1+\kappa)(3+\omega)(1-\varepsilon)^{-2}$-approximation to the optimal value $\hat{T}$ of MNL–LATE.*

Note that the amplification procedure is simply used to facilitate the proof of approximation ratio. In a practical implementation of MNL–LATE_ALGO, we may skip it and simply choose an arbitrary location within a particular subarea to represent that subarea. In other words, we solve MNL–DLATE rather than MNL–DALATE. The solution, though bearing the same worst case performance as the one with amplification, is in general better, i.e., closer to the optimal value.

18.3.3 Summary

We demonstrate in this section that, though the formulation and solution presented in Section 18.2 have a constrained set of locations for mobile sinks to choose, it is not difficult, in theory, to extend them by relaxing the constraint. However, we believe that all the results presented in the section are more for pure theoretical purpose, as the improvement (in terms of the absolute value of the lifetime) can be very marginal and the cost to obtain this improvement is huge. Especially in the link associated case, the cardinality of $\mathscr{A}$ is in the order of $\left(\frac{n}{\kappa}\right)^2$, which can be enormous if we are chasing an accurate approximation with very small κ. Therefore, while appreciating the beauty of the theory, we do caution the readers for any practical use of it.

18.4 Energy Conservation with Mobile Nodes:

The Extreme Usage of The Substitution Effect

Although the substitution effect (see Section 18.2.1 and 18.2.3) is overwhelmed by the load balancing effect in the mobile sink approach, one might still wonder if it is possible to fully exert the benefit of this effect. As illustrated by Figure 18.10, such an approach does exist [35], where the sink is kept static while some powerful mobile nodes are changing their location from time to time to replace certain overloaded (static) nodes. However, the analysis performed in [35] is based on a fluid model (similar to [21]), hence the results hold only in an asymptotic sense and cannot be applied whenever a specific network topology is given. Therefore, we reformulate the problem following the general framework presented in Section 18.2. The great benefit is that, as we will show later, almost all the results presented in Section 18.2 apply here, only with a change of the separation oracle.

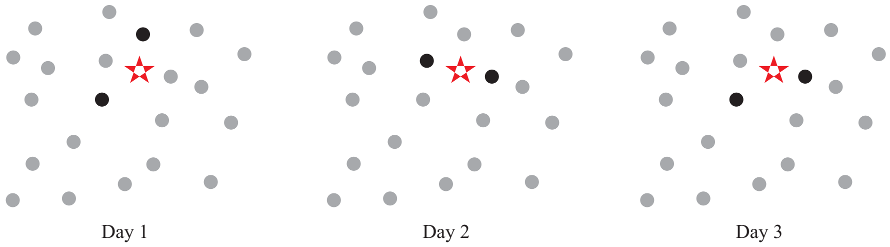


Fig. 18.10 Using mobile nodes to balance the traffic load within a WSN. The star represents the static sink, and nodes with darker color are the mobile nodes. Note that the mobility is (again) slow, as the mobile nodes may change their locations very infrequently.

18.4.1 MNL with Multiple Mobile Nodes (MNL–MMN)

We keep using the model and terminologies presented in Section 18.2.1. The differences are: 1) there is only one static sink $s \in \mathcal{N}$, and 2) $\mathcal{L}_k \in \mathcal{N}$ denotes the mobile node (rather than sink) locations during the kth epoch. Note that, as the substitution effect demands co-locations of mobile nodes with certain sensor nodes, the problem only has an on-graph version, i.e., $\mathcal{V} = \mathcal{N}$, where $\mathcal{V}$ represents potential locations of the mobile nodes. Let us directly formulate the MNL–MMN problem into its Path-Flow form:

$$\text{maximize} \quad T = \sum_k t_k \tag{18.25}$$

$$\text{subject to} \qquad \sum_{p\in P_{is}^k} f(p) \;\ge\; \lambda_i t_k \qquad \forall i \neq s, k \tag{18.26}$$

$$\sum_{k:i\notin\mathscr{L}_k}\sum_{p\in P_i^k} f(p)\left(e_i^{\mathrm{T}} + \mathrm{I}_{p\notin P_{is}^k}\cdot e^{\mathrm{R}}\right) \;\le\; E_i \qquad \forall i \neq s \tag{18.27}$$

$$t_k, f(p) \;\ge\; 0 \qquad \forall k, p \tag{18.28}$$

where I_A is the indicator function of event A, p is a path between a node and the sink s, and $f(p)$ is the flow going through that path. Furthermore, we denote by P_{is}^k the path set from node i to the sink s and by P_i^k the set of paths going through node i, both in the kth epoch. Splitting a flow among a set of paths implies that we allow multi-path routing strategy to be taken by the optimal solution. As this formulation is very similar to that of MNL (18.13)–(18.14), the same happens to the dual problem of MNL–MMN.

$$\text{minimize} \quad G(\mathbf{w}) = \sum_i E_i w(i) \tag{18.29}$$

$$\text{subject to} \qquad \sum_{i\neq s} \lambda_i W(i,k) \;\ge\; 1 \qquad \forall k \tag{18.30}$$

$$\sum_{j\in p\in P_{is}^k, j\notin\mathscr{L}_k, j\neq s} w(j)\left(e_j^{\mathrm{T}} + \mathrm{I}_{j\neq i}\cdot e^{\mathrm{R}}\right) - W(i,k) \;\ge\; 0 \qquad \forall i \neq s, k, p \tag{18.31}$$

$$w(i), W(i,k) \;\ge\; 0 \qquad \forall i \neq s, k \tag{18.32}$$

where the $w(i)$ is the weight assigned to node i, representing the marginal cost of using an additional unit energy of node i; and $W(i,k)$ is the weight of a commodity, i.e., data flow going from node i to sink s during epoch k; it indicates the marginal cost of rejecting a unit demand of the commodity. The striking similarity between MNL and MNL–MMN immediately suggests the validity of applying most of the results obtained in Section 18.2 to MNL–MMN, with, of course, certain exceptions, as we will discuss in Section 18.4.2.

18.4.2 Theorem, Complexity, and Algorithm

We directly use the following theorem to characterize the optimal solution of MNL–MMN; detailed analysis is omitted as it basically follows the same line as Sections 18.2.2 and 18.2.3.

Theorem 3 (MAX-LIFETIME MIN-POTENTIAL RATIO THEOREM RELOADED). *Given the lifetime maximization problem formulated in (18.25)–(18.28), the optimal lifetime $\hat{T}$ is such that*

$$\hat{T} = \min_{\mathbf{w}} \left[\frac{G(\mathbf{w})}{\rho(\mathbf{w})}\right]$$

where $G(\mathbf{w}) = \sum_i E_i w(i)$ is a linear combination of the energy reserves of all nodes with coefficients $w(i)$, and

$$\rho(\mathbf{w}) \equiv \min_k \rho_k(\mathbf{w}) = \min_k \left[\sum_i \lambda_i \left(\sum_{j \in \min\{p | p \in P_{is}^k\}, j \notin \mathscr{L}_k, j \neq s} w(j)(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}}) \right) \right]$$

is the minimum "potential" (computed as the sum of the minimum "cost", given $w(i)$, to route λ_i from node i to the sink s) achieved among all possible mobile node layouts $\{\mathscr{L}_k\}$.

Similar to the on-graph MNL, the answer to TMNTM is also positive for MNL–MMN. We omit the proof here; it follows the same line as the proof for **Proposition 2**. Note that the question asked by TMNTM for MNL–MMN is whether moving the mobile nodes is superior to keep them static. Moreover, for certain cases where (off-graph) sink mobility does not improve network lifetime, adding (on-graph) mobile nodes may still benefit network lifetime. Taking the networks shown in Figure 18.3 as examples, moving some mobile nodes (even just one) around the rings to replace those static nodes in turn is bounded to improve the network lifetime. The latter fact seems to suggest that it would be better to combine mobile nodes and mobile sinks.

Note that $\rho(\mathbf{w})$ is also the separation oracle of the dual MNL-MMN. Let $K = |\mathscr{L}_k| = m$, $\omega(i) = \lambda_i$, $\ell(j) = w(j)\left(e_j^{\mathrm{T}} + \mathrm{I}_{j \neq i} \cdot e^{\mathrm{R}}\right)$, and $d(i) = \sum_{j \in \min\{p|p \in P_{is}^k\}, j \neq s} \ell(j)$, then the separation oracle is equivalent to the following decision problem:

INSTANCE: A graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, a weight assignment $\omega(i) : \mathscr{N} \to \mathbb{R}_0^+$, a length assignment $\ell(i) : \mathscr{N} \to \mathbb{R}_0^+$, positive integer $K \leq |\mathscr{N}|$, a special vertex $s \in \mathscr{N}$, and positive real number B.

QUESTION: Is there a set $\mathscr{P}$ of K points on $\mathscr{G}$ such that, if we set $\ell(i) = 0, i \in \mathscr{P}$ and let $d(i)$ be the length of the shortest path from i to s, then $\sum_i \omega(i) \cdot d(i) \leq B$?

If we could solve this decision problem or give a proper approximation to its optimization version, we would be able to apply MNL_ALGO introduced in Section 18.2.4 to solve MNL–MMN. However, we show in the following that the separation oracle of dual MNL–MMN is NP-hard, and hence MNL–MMN is also NP-hard, again due to [26] (**Theorem 3.3**).

Proposition 9. *The separation oracle of MNL–MMN is NP-hard.*

Proof: The NP-hardness of the separation oracle can be shown by transforming from the p-median problem on special cases such as the one shown in Figure 18.11. It is straightforward to see that $\mathscr{P}$ has to be chosen among $\mathscr{V}_2$, in order to get a maximum weight reduction $M(K + |\mathscr{V}_1|)$ in $\sum_i \omega(i) \cdot d(i)$. However, which $\mathscr{P}$ vertices to be chosen among $\mathscr{V}_2$ are determined by the solution of the p-median problem on $\tilde{\mathscr{G}}(\mathscr{V}_1, \mathscr{E}_1)$, where $\mathscr{E}_1$ refers to the set of edges whose both ends are in $\mathscr{V}_1$. This is so because the answer to the separation oracle is true iff the answer to the p-median problem with $B' = B - M(|\mathscr{V}_2| - K)$ is true. Conversely, if we could address

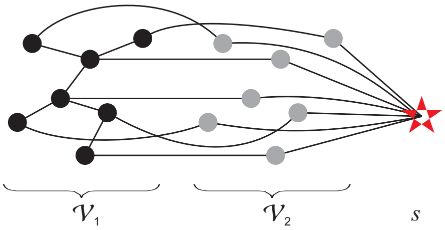


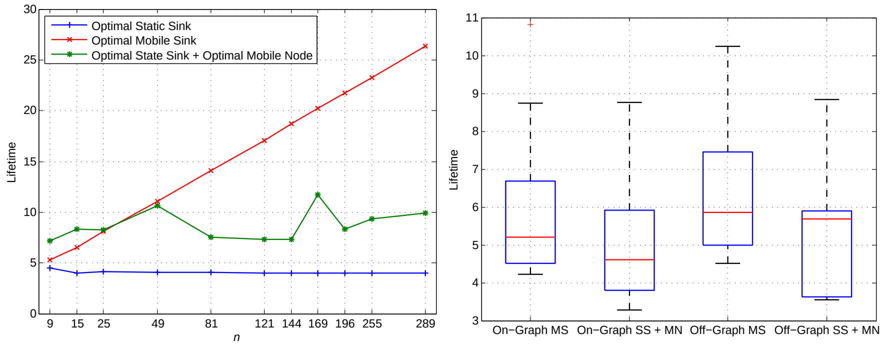
Fig. 18.11 A graph $\mathscr{G}(\mathscr{V}_1 \cup \mathscr{V}_2 \cup \{s\}, \mathscr{E})$, where vertices in $\mathscr{V}_1$ are connected to s only through some vertices in $\mathscr{V}_2$, and vertices in $\mathscr{V}_2$ are not directly connected to each other. We assign uniform $\omega(i) = 1$ to all vertices in both $\mathscr{V}_1$ and $\mathscr{V}_2$, and we assign $\ell(i) = 1$ to vertices in $\mathscr{V}_1$ and $\ell(i) = M$ to vertices in $\mathscr{V}_2$, where $M > |\mathscr{V}_1|$.

the separation oracle for $\mathscr{G}(\mathscr{V}_1 \cup \mathscr{V}_2 \cup \{s\}, \mathscr{E})$, we would actually solve the p-median problem in $\tilde{\mathscr{G}}(\mathscr{V}_1, \mathscr{E}_1)$, as the medians are indicated by the chosen $\mathscr{P} \subset \mathscr{V}_2$. Q.E.D. ⊠

The existence of efficient PTAS with provable approximation ratio to this separation oracle, to our best knowledge, is unfortunately unknown, although heuristics with good empirical performance can be derived based on short-path algorithms. Therefore, we restrict ourselves to the single mobile node case when making comparisons between the mobile node and mobile sink approach in Section 18.4.3.

18.4.3 Numerical Results

We apply the same settings as those in Section 18.2.5, and we compare the two approaches, namely mobile sink and mobile node, for both grid networks of different sized and arbitrary networks with 100 nodes. We illustrate these comparisons in Figure 18.12. Figure 18.12(a) shows that, whereas the improvement (against the static sink approach) brought by the mobile node approach is more or less a constant, the mobile sink approach yields an increasing improvement in larger networks. The better performance for the mobile node approach in small networks is not a surprise: the substitution effect is the dominating factor of the lifetime improvement in these networks, as already explained Section 18.2.5. in In Figure 18.12(b), we use MS, SS, and MN to refer to mobile sink, static sink and mobile node, respectively. When the mobile node approach is used, the static sink is always put at its optimal location. Although the mobile node approach is still inferior to the mobile sink approach in most cases, the difference is much less significant than the cases of grid networks.



(a) Grid networks of different size

(b) Arbitrary Networks with 100 nodes

Fig. 18.12 Comparing mobile node with mobile sink.

18.4.4 Summary

In this section, we have shown the close relation between the mobile sink and mobile node approaches, by unifying their problem formulations and solution techniques. The other benefit of formulating the mobile node approach into our general optimization framework is to allow detailed investigations on particular network topologies. This actually brings us a slight surprise: Although the mobile sink approach appears to be superior to the mobile node approach in regular topologies (or their asymptotic version [35]), their performances are not very different from each other in arbitrary topologies.

18.5 Energy Conservation with Mobile Relays: Using Mechanical Data Transportation Smartly

Although running WSNs under the slow mobility regime (such as the mobile sink and mobile node approaches presented in the previous sections) may significantly improve the network lifetime, the need for dynamic routing configurations according to the specific locations of the mobile entities could incur additional maintenance overhead in practice. An alternative solution is to use mobile relays to “pick up” data from node through one-hop transmissions and then to transport the data with mechanical movements [5, 28]. Unfortunately, this extreme approach incurs a large transmission delay due to the limited speed of mechanism movements. In this section, we discuss a good compromise made between the aforementioned two extremes: A hybrid approach that jointly considers multi-hop transmissions and mechanical data transportation [32]. As shown in Figure 18.13, certain locations in

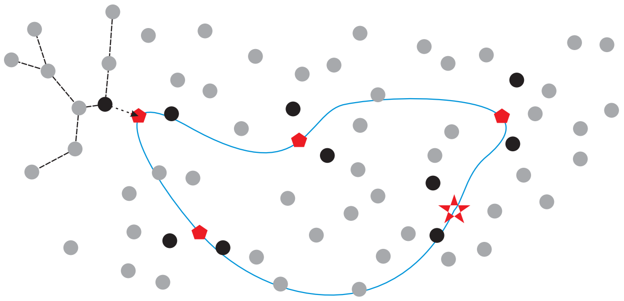


Fig. 18.13 Using a mobile relay to reduce the energy consumption of sensor nodes. The star represents the relay, the pentagons are its RPs, and nodes with darker color are those that buffer or aggregate data sent from other nodes. In other words, a darker colored node is the root of one data collection tree; one of such trees is illustrated at the upper-left corner. The mobility is fast in this case, as the mobile relay is usually required to finish the tour through all RPs within a given time period.

the Euclidean plane where the considered WSN locates are chosen as *rendezvous points* (RPs) [36]. A mobile relay periodically travels along a predefined tour and picks up data at RPs, while nodes that may directly reach a RP buffer (or even aggregate[7]) data originated from other nodes and transfer the data to the relay when it arrives at the RP. The advantage of this approach, compared with those slow mobility approaches, is that the multi-hop routing can be configured offline, as it does not change with different locations of the mobile relays.

18.5.1 The Single Mobile Relay Positioning (SMRP) Problem

We discuss the problem of identifying the RPs for one mobile relay, which we term *single mobile relay position* (SMRP) problem. Our discussion will be based on the general framework introduced in Section 18.2.1. This means that there is a set of potential locations $\mathscr{V}$ within which we would like to identify the optimal subset $\mathscr{S}$ as the RPs. The relay mobility is on-graph if $\mathscr{V} = \mathscr{N}$, or is off-graph if $\mathscr{V} \supset \mathscr{N}$. Usually, $|\mathscr{S}|$ is bounded by a certain integer, as the relay can only visit up to that number of locations given a certain data delivery deadline. Also, we assume there is no data aggregation involved in the routing configuration. The reason we focus only on a single mobile relay is twofold: first, it is already very hard to solve the problem involving only a single relay, and second, solutions for multiple mobile relays could always be based on those for a single mobile relay.

To simplify the exposition, we present the problem formulation as a decision problem rather than an optimization problem. It is well known that these two for-

[7] By aggregation, we refer to any transformation that summarizes or compresses the data acquired and received by a certain node and hence reduce the volume of the data to be sent out, e.g., [25].

mulations are equivalent as there always exists a polynomial-time reduction from one to the other.[8] We present the problem for on-graph mobility first and then show how to extend it to off-graph mobility.

> INSTANCE: A set of nodes $\mathscr{N}$, a cost assignment $\mathbf{c} : c(i,j) = e, \forall i,j \in \mathscr{N}$, a set $\mathscr{S}$ of *virtual sink locations* with $|\mathscr{S}| < |\mathscr{N}|$, and for each $i \in \mathscr{N}$, a transmission energy e_i^{T}, a receiving energy e^{R}, an energy reserve E_i, a rate λ_i, a constraint that i sends data to only one $s \in \mathscr{S}$, and a positive real number t.
>
> QUESTION: Is there a *rendezvous schedule* $\{\delta_{is}\}$, where $\delta_{is} : \mathscr{N} \times \mathscr{S} \to \{0,1\}$, $\sum_i \delta_{is} = 1$ (only one mobile relay is allowed) and $\sum_i \sum_s \delta_{is} \leq |\mathscr{S}|$, such that the lifetime T is at least t?

In order to extend the formulation to accommodate off-graph mobility, we simply need to assign a zero rate to those vertices in $\mathscr{V} \backslash \mathscr{N}$. It is obvious that the nodes that take a higher traffic load are those that are one-hop from one of the RPs (those darker colored ones shown in Figure 18.13). Therefore, the objective of the optimization problem is, again, to minimize the maximum load, or, in other words, to balance the traffic load. If we assume the same transmission energy e^{T} for all nodes, the bottleneck node that constrains the lifetime is the one that serves as the root of the "heaviest" data collection tree, where the weight of a tree is the total traffic load generated by all the nodes in the tree. It is straightforward to see that SMRP is equivalent to the notoriously hard *base station placement* (BSP) problem [4], which has been shown as NP-hard and whose PTAS is not known by far.[9]

18.5.2 A Variation of SMRP

The difficulty of the problem formulated in Section 18.5.1 stems from the fact it is the property of individual data collection trees that needs to be optimized. In a recent proposal, Xing et al. [36] suggest a simplified problem formulation by 1) assuming

[8] As an example, the decision problem related to MNL (on-graph) is the following:

> INSTANCE A set of nodes $\mathscr{N}$, a cost assignment $\mathbf{c} : c(i,j) = e, \forall i,j \in \mathscr{N}$, a set $\mathscr{S}$ of sinks with $|\mathscr{S}| < |\mathscr{N}|$, and for each $i \in \mathscr{N}$, a transmission energy e_i^{T}, a receiving energy e^{R}, an energy reserve E_i, a rate λ_i, and a positive real number t.
>
> QUESTION: Is there a *sink layout schedule* $\{(sl_k, t_k)\}$ (sl_k is a vector of $[\delta_{is}^k]$ where $\delta_{is}^k : \mathscr{N} \times \mathscr{S} \to \{0,1\}$ and $\sum_i \sum_s \delta_{is}^k = |\mathscr{S}|$) such that the lifetime $T = \sum_k t_k$ is at least t?

This problem can be shown as NP-hard by, for example, a polynomial-time reduction from the DOMINATING SET on a *unit disk graph* [22].

[9] Although Shi et al. [31] have proposed an approximation algorithm for BSP, using a technique similar to the one presented in Section 18.3.2.1, that algorithm is only pesudo-polynomial in time, as the time complexity is actually exponential in the number of base stations.

a uniform data rate from all nodes that produce sensory data (or source nodes)[10], 2) identical transmission energy and zero receiving energy for all nodes, 3) allowing "many-to-one" data aggregation, and 4) optimize total energy consumption instead of lifetime. While the first three lead to identical traffic load on each node,[11] the last change shifts the objective from individual data collection trees to the WSN as a whole. The direct consequence of these simplifications is the following:

- The problem formulation may explicitly involve the data delay factor as a design constraint. This eventually translates to the constraint on the length of the tour through all RPs, which appears to be at least as hard as the EUCLIDEAN TRAVELING SALESMAN (ETS) problem.
- The problem objective becomes minimizing the number of links that are needed to connect all the source nodes to the RPs. Due to the identical transmission energy for every node, the energy consumption of transmitting data along a routing path can be approximated by the Euclidean distance between the source and destination, which in turn suggests an approximation of the objective by the Euclidean length of all routing trees. This approximated objective is actually a GEOMETRIC STEINER TREE (GST) problem.

Although both ETS and GST are NP-complete, ETS actually has a straightforward 2-approximation algorithm given by GST, while GST admits a PTAS whose ratio is pretty close to 1 (can be, in fact, smaller than $\frac{2}{\sqrt{3}}$ [8]). Based on these observations, efficient approximation algorithms can be designed to identify RPs within an unconstrained Euclidean space, unlike SMRP whose RPs are limited to $\mathscr{V}$. In the original proposal [36], two algorithms are given, respectively, for choosing RPs in 2D (e.g., the region covered by a WSN) and 1D (e.g., a fixed track) spaces. We only discuss the first algorithm in this section.

Assume the tolerable data delivery delay is D, meaning that the total length of the relay tour must be no more than $L = D\bar{v}$, where $\bar{v}$ is the average speed of the mobile relay. The problem (decision version) can be specified as:

INSTANCE: A set of **source** nodes $\mathscr{N}$, two positive real numbers L and C.

QUESTION: Is there a mobile relay tour $\mathscr{U}$ no longer than L and a set of geometric trees $\{\mathscr{T}_k(\mathscr{V}_k, \mathscr{E}_k)\}$ rooted on $\mathscr{U}$ such that 1) $\mathscr{V}_k \subseteq \mathscr{V}$ and $\mathscr{N} \subseteq \bigcup_k \mathscr{V}_k$, and 2) $\sum_k \sum_{(i,j)\in\mathscr{E}_k} d(i,j)$ (where $d(i,j)$ is the Euclidean distance between node i and node j) is no greater than C?

Note that a tree edge $(i,j) : i,j \in \mathscr{V}_k$ does not necessarily represent a physical link; it instead may represent a routing path that goes through other non-source nodes (nodes not in $\mathscr{N}$ but belong to the WSN). Obviously, the optimization version of the problem aims at minimizing the total Euclidean length of all routing paths that

[10] This is a special case of our general formulation, which assigns a uniform data rate to the source nodes and a zero rate to others.

[11] The "many-to-one" data aggregation implies that, no matter how many unit of flows converge at an intermediate node, that node only send one unit flow out. These are cases where special aggregation functions such as AVERAGE, MAX, or MIN are used.

are involved in data transmission. This problem can be shown to be NP-hard by a reduction from the ETS problem. Fortunately, due to the reason explained earlier, there exists good approximation algorithms to solve this problem.

To motivate the algorithm, let us first consider an extreme situation where the mobile relay is replaced by a static sink. In this case, the optimal routing tree connecting all source nodes with minimum total length is given by the GEOMETRIC STEINER TREE (GST). Now, we let the sink start moving and thus serve as the mobile relay. As GST provides a lower bound of ETS, moving the relay along the GST appears to be a reasonable choice: as it may strike a good balance between minimizing the transmission cost and limiting the tour length. Based on these observations, the approximation algorithm *rendezvous design for variable tracks* (RD–VT) makes use of approximate GST and ETS solvers as the oracles to address the problem of minimizing $\sum_k \sum_{(i,j)\in\mathcal{E}_k} d(i,j)$ under constrained tour length L. The algorithm first

Algorithm 16 RD–VT

```
Input: Node set $\mathcal{N}$, tour length bound $L$, and threshold $\sigma$
1: Find an approximate GST of $\mathcal{N}$: $\mathcal{T}_N = (\mathcal{V}_N, \mathcal{E}_N)$, where $\mathcal{V}_N \supseteq \mathcal{N}$;
2: Initialize the tour length $\Gamma = L/2$ and a starting point $\Theta \in \mathcal{V}_N$;
3: repeat
4:     Traverse $\mathcal{T}_N$ in depth-first manner from $\Theta$ until the length visited is $\Gamma$,
       denote the subtree traveled as $\mathcal{T}_R = (\mathcal{V}_R, \mathcal{E}_R)$;
5:     Let $\mathcal{R} = \{r_i | r_i$ is the first intersection between $\mathcal{T}_R$ and the path from the $i$th node in $\mathcal{N}$
       to $\Theta$ on $\mathcal{T}_N\}$;
6:     Find the ETS tour $\mathcal{U}$ that goes through $\mathcal{R}$ and denote its length by $|\mathcal{U}|$;
7:     Update trial step: $\Gamma = \Gamma + \Delta$, where $\Delta = (L - |\mathcal{U}|)/2$;
8: until $\Delta \leq \sigma$
9: return a set $\mathcal{R}$ of RPs and a set of trees $\{\mathcal{T}_k(\mathcal{V}_k, \mathcal{E}_k)\} = \mathcal{T}_N \setminus \mathcal{T}_R$
```

constructs an approximate minimum GST, and then recursively traverses it in depth-first manner to find proper RPs. Initially, the length to be traversed on the GST is set as $L/2$. For each recursion, the visited subtree is expanded according to the length to be traversed and RPs are identified on the subtree (5th step), then an approximate EST oracle is called to connect the current RPs, finally the additional length to be traversed in the next recursion is set to be half of the difference between L and the EST tour length (computed by the EST oracle) in the current recursion (7th step). The algorithm terminates if this difference becomes no larger than a threshold σ. The following proposition confirms the performance of RD–VT.

Proposition 10. *Let α be the best known approximation ratio for the GST problem and $\beta = L/\left(\sum_{(i,j)\in\mathcal{E}_N^*} d(i,j)\right)$, where $\mathcal{T}_N^* = (\mathcal{N}, \mathcal{E}_N^*)$ is the minimum GST of $\mathcal{N}$, the approximation ratio of the RD-VT algorithm is no greater than $\frac{\alpha-\beta/2}{1-\beta}$.*

Proof: For simplicity, we represent the total edge length of a tree $\mathcal{T}(\mathcal{V}, \mathcal{E})$ by $c(\mathcal{T}) = \sum_{(i,j)\in\mathcal{E}} d(i,j)$. Suppose the optimal set of RPs is $\mathcal{R}^*$ and its minimum GST is $\mathscr{T}_R^*$, and the optimal set of routing trees is $\{\mathscr{T}_k^*\}$, we have $\sum_k c(\mathscr{T}_k^*) + c(\mathscr{T}_R^*) \geq c(\mathscr{T}_N^*)$, as the union of $\{\mathscr{T}_k^*\}$ and $\mathscr{T}_R^*$ is a GST. As we have discussed, GST gives a lower bound of EST and the tour length of EST is bounded above by L, meaning $c(\mathscr{T}_R^*) \leq L$, thus $\beta = L/c(\mathscr{T}_N^*) \geq L/(\sum_k c(\mathscr{T}_k^*) + c(\mathscr{T}_R^*)) \geq L/(\sum_k c(\mathscr{T}_k^*) + L)$, which leads to $L \leq \frac{\beta}{1-\beta} \sum_k c(\mathscr{T}_k^*)$. Therefore we have:

$$\begin{aligned}\sum_k c(\mathscr{T}_k^*) &\geq c(\mathscr{T}_N^*) - c(\mathscr{T}_R^*)\\ &\geq \frac{c(\mathscr{T}_N)}{\alpha} - L\\ &= \frac{c(\mathscr{T}_N) - c(\mathscr{T}_R)}{\alpha} + \frac{c(\mathscr{T}_R)}{\alpha} - L\\ &\geq \frac{\sum_k c(\mathscr{T}_k)}{\alpha} + \frac{L/2}{\alpha} - L\\ &\geq \frac{\sum_k c(\mathscr{T}_k)}{\alpha} + \frac{1-2\alpha}{2\alpha}\frac{\beta}{1-\beta}\sum_k c(\mathscr{T}_k^*)\end{aligned}$$

The approximation ratio is hence $\frac{\sum_k c(\mathscr{T}_k)}{\sum_k c(\mathscr{T}_k^*)} \leq \alpha + \frac{\beta(2\alpha-1)}{2(1-\beta)} = \frac{\alpha-\beta/2}{1-\beta}$. Q.E.D. ☒

For $\beta \leq 0.56$ and $\alpha = \frac{2}{\sqrt{3}}$, the ratio is smaller than 2. Since β is usually small (otherwise the mobile sink may almost visit every source nodes), the performance of RD–VT is pretty satisfactory. In particular, if $\mathscr{N}$ includes the whole WSN, the minimum GST is actually a MINIMUM SPANNING TREE (MST) and hence $\alpha = 1$ (MST can be perfectly solved efficiently), in which case the performance of RD–VT is further improved.

18.5.3 Summary

In this section, we focus on exploiting entity mobility in the fast mobility regime. We first formulate the mobile relay problem based on the optimization framework described in Section 18.2. As the resulting problem is very hard and there is no known PTAS for it, we discuss instead a variation appearing in the literature. This variation admits a PTAS with satisfactory performance at a cost of several simplifying assumptions. It is still an open question if we can handle the problem efficiently in more general settings where some of these simplifying assumptions may not hold.

18.6 Conclusion

No matter how advanced the stage we are in designing WSNs, network lifetime and energy efficiency will always be recurring issues pertaining to WSNs. To fully utilize the limited energy reserve of sensor nodes, we have to really "think out of the box" and explore new approaches. In our opinion, actively exploiting entity mobility in WSNs is such an approach, and an optimization framework is a powerful tool to guide the designs using this approach. By providing an in-depth description of the construction and application of such a general optimization framework as well as the engineering insights we can acquired from it, we are hoping to stimulate the invention of new design methodologies for WSNs.

References

1. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A Survey on Sensor Networks. *IEEE Communication Mag.*, 40(8):104–112, 2002.
2. V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local Search Heuristics for k-Median and Facility Location Problems. *SIAM J. on Computing*, 33(3):544–562, 2004.
3. M.A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G.S. Sukhatme, W.J. Kaiser, M.Hansen, G.J. Pottie, M.Srivastava, and D. Estrin. Call and Response: Experiments in Sampling the Environment. In *Proc. of the 2nd ACM SenSys*, 2004.
4. A. Bogdanov, E. Maneva, and S. Riesenfeld. Power-aware Base Station Positioning for Sensor Networks. In *Proc. of the 23rd IEEE INFOCOM*, 2004.
5. A. Chakrabarti, A. Sabharwal, and B. Aazhang. Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks. In *Proc. of the 2nd IEEE IPSN*, 2003.
6. J.-H. Chang and L. Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In *Proc. of the 19th IEEE INFOCOM*, 2000.
7. I Chatzigiannakis, A. Kinalis, S. Nikoletseas, and J. Rolim. Fast and Energy Efficient Sensor Data Collection by Multiple Mobile Sinks. In *Proc. of the 5th ACM MobiWAC*, 2007.
8. D.-Z. Du, Y. Zhang, and Q. Feng. On Better Heuristic for Euclidian Steiner Minimum Trees. In *Proc. of the 32nd IEEE FOCS*, 1991.
9. R.W. Floyd. Algorithm 97: Shortest Path. *Commun. ACM*, 5(6):345, 1962.
10. L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, N.J., 1962.
11. S.R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations. In *Proc. of IEEE Globecom*, 2003.
12. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.
13. N. Garg and J. Könemann. Faster and Simpler Algorithms for Multicommodity Flow and other Fractional Packing Problems. In *Proc. of the 38th IEEE FOCS*, 1997.
14. M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. on Networking*, 10(4):477–486, 2002.
15. Y. Gu, D. Bozda¿, E. Ekici, F. Ozguner, and C. Lee. Partitioning-Based Mobile Element Scheduling in Wireless Sensor Networks. In *Proc. of the 2nd IEEE SECON*, 2005.
16. H.S. Kim, T.F. Abdelzaher, and W.H. Kwon. Minimum Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks. In *Proc. of the 1st ACM SenSys*, 2003.
17. A. Kinalis and S. Nikoletseas. Adaptive Redundancy for Data Propagation Exploiting Dynamic Sensory Mobility. In *Proc. of the 11th ACM MSWiM*, 2008. Also in Journal of Interconnection Networks (JOIN), 2010.
18. L. Li, J.Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. Analysis of A Cone-based Distributed Topology Control Algorithm for Wireless Multi-hop Networks. In *Proc. of the 20th ACM PODC*, 2001.

19. N. Li and J. Hou. Topology Control in Heterogeneous Wireless Networks: Problems and Solutions. In *Proc. of the 23rd IEEE INFOCOM*, 2004.
20. Q. Li, M. De Rosa, and D. Rus. Distributed Algorithms for Guiding Navigation across a Sensor Network. In *Proc. of the 9th ACM MobiCom*, 2003.
21. J. Luo and J.-P. Hubaux. Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. In *Proc. of the 24th IEEE INFOCOM*, 2005.
22. J. Luo and J.-P. Hubaux. Joint Sink Mobility and Routing to Increase the Lifetime of Wireless Sensor Networks: The Case of Constrained Mobility. *IEEE/ACM Trans. on Networking*, 2010 (to appear).
23. J. Luo, J. Panchard, M. Piórkowski, M. Grossglauser, and J.-P. Hubaux. MobiRoute: Routing towards a Mobile Sink for Improving Lifetime in Sensor Networks. In *Proc. of the 2nd IEEE/ACM DCOSS*, 2006.
24. M. Ma and Y. Yang. SenCar: An Energy-Efficient Data Gathering Mechanism for Large-Scale Multihop Sensor Networks. *IEEE Trans. on Parallel and Distributed Systems*, 18(10):1478–1488, 2007.
25. S. Madden, M.J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A Tiny Aggregaton Service for Ad-Hoc Sensor Networks. In *Proc. of the 5th USENIX OSDI*, 2002.
26. G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.
27. V. Rodoplu and T. H. Meng. Minimum Energy Mobile Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, 1999.
28. R.C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Mobeling a Three-tier Architecutre for Sparse Sensor Networks. In *Proc. of the 1st IEEE SNPA*, 2003.
29. F. Shahrokhi and D.W. Matula. The Maximum Concurrent Flow Problem. *J. ACM*, 37(2):318–334, 1990.
30. Y. Shi and Y.T. Hou. Theoretical Results on Base Station Movement Problem for Sensor Network. In *Proc. of the 27th IEEE INFOCOM*, 2008.
31. Y. Shi, Y.T. Hou, and A. Efrat. Algorithm Design for A Class of Base Station Location Problems in Sensor Networks. *Springer Wireless Networks*, 15(1):21–38, 2009.
32. A. Somasundara, A. Kansal, D.D. Jea, D. Estrin, and M.B. Srivastava. Controllably Mobile Infrastructure for Low Energy Embedded Networks. *IEEE Trans. on Mobile Computing*, 5(8):958–973, 2006.
33. A.A. Somasundara, A. Ramamoorthy, and M.B. Srivastava. Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines. In *Proc. of the 25th IEEE RTSS*, 2004.
34. G. Wang, G. Cao, and T. La Porta. Movement-Assisted Sensor Deployment. *IEEE Trans. on Mobile Computing*, 5(6):640–652, 2006.
35. W. Wang, V. Srinivasan, and K.-C. Chua. Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks. In *Proc. of the 11th ACM MobiCom*, 2005.
36. G. Xing, T. Wang, W. Jia, and M. Li. Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station. In *Proc. of the 9th ACM MobiHoc*, 2008.
37. F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks. In *Proc. of the 8th ACM MobiCom*, 2002.