Data Transmission and Base-Station Placement for Optimizing Network Lifetime*

Esther M. Arkin
Applied Math and Statistics, Stony Brook University
estie@ams.stonybrook.edu

Alon Efrat
Computer Science, University of Arizona
alon@cs.arizona.edu

Joseph S. B. Mitchell
Applied Math and Statistics, Stony Brook University
jsbm@ams.stonybrook.edu

Valentin Polishchuk
Information Technology, University of Helsinki
polishch@cs.helsinki.fi

Srinivasan Ramasubramanian
Electrical and Computer Engineering, University of Arizona
srini@ece.arizona.edu

Swaminathan Sankararaman
Computer Science, University of Arizona
swami@cs.arizona.edu

Javad Taheri
Computer Science, University of Arizona
taheri@cs.arizona.edu

ABSTRACT

We study the problem of transmitting data from a set of sensors to a base-station where the data is to be gathered. Each sensor continuously generates data and has to transmit it through the network (via other sensor nodes) to the base-station. Considering the battery limitations of the sensors, our goal is to find an optimum location of the base-station and a corresponding data transmission scheme for routing the data from the sensors, such that the network is operating for the longest possible time. We focus mainly on tree networks for 2-level trees, with at most 2 hops from sensor to the base-station. For such networks we give efficient algorithms for forwarding data from sensors to the base-station and for locating the base-station optimally for maximizing network lifetime. Further, we show that determining a transmission protocol for trees with 3 or more levels is NP-hard. We demonstrate the effectiveness of our methods with experimental results on simulated data, comparing our 2-level tree algorithm with methods based on linear programming.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*; C.2.1 [**Network Architecture and Design**]: Distributed Networks, Wireless Communication

DIALM-POMC'10, September 16, 2010, Cambridge, MA, USA.

General Terms

Algorithms

Keywords

Algorithms, Routing, Sensor Networks, Base-station location, Optimization, Matching

1. INTRODUCTION

Recent advances in building powerful, highly integrated and energy-efficient electronic devices coupled with emerging technologies in digital communication have enabled the creation of large-scale Wireless Sensor Networks (WSNs) (see [4, 24]). WSNs have a vast number of applications, including military surveillance and tracking, environmental monitoring, human-centric applications and robotics (see, e.g., the survey Arampatzis *et al.* [5]).

Such networks consist of a large number of sensor nodes deployed in a region to monitor and gather information, such as video, audio, temperature, etc. The hardware of each node is typically limited in the resources it supports, with limited processing and storage capabilities and limited power/energy (battery life). The information gathered by the sensor nodes is transmitted to a central authority through wireless communication for more advanced processing via a base-station, which is linked to the central authority. Hence, all of the data captured at a sensor node needs to be transmitted to one or more such base-stations. The major bottleneck in maintaining operation of the network for the longest possible time are the batteries of the sensors; which are depleted of their energy primarily by the transmission of data rather than by sensing or computation. Thus, locating the base-station and finding a transmission scheme for the sensor nodes such that the lifetime of the network is maximized is an important optimization problem.

In terms of network topology, the current literature typically distinguishes between two different topologies: *Flat Network Topology*, in which all transmissions are homoge-

neous, i.e., each sensor node sends its data directly to the base-station, and *Multi-hop Network Topology*, in which transmissions are inhomogeneous, i.e., sensors can relay their data to the base-station through other sensor nodes. The cost of transmission is impacted by both the data bit-rate and the distance of the transmission. The lifetime of the network can be defined in various ways; it is defined as the length of time from when the system starts operating until a termination event, which may be defined to be the moment of battery expiration of any one of the sensors, or of all of a specific set of sensors, or of a certain percentage of the sensors. An additional consideration is that the sensor nodes may be capable of recharging their batteries (at a certain rate, e.g., using a solar cell) and this also has to be taken into account.

Sensor networks are often employed for collecting data periodically from an environment. Such networks are typically operated by turning on the sensors for a brief period of time to collect data, to communicate with neighboring sensors in order to transmit the data to the base-station. Therefore, it is quite common to assume that the most significant factor affecting the lifetime of a (battery operated) network is the length of idle periods of the nodes when the receivers are switched off to save energy. Therefore, two nodes that need to communicate (in the future) need to agree on a future time to wake up. Switching sensors between idle and active modes is expensive; frequent switching is to be avoided. Hence, the longer idle time the sensors have, the more energy efficient the network is.

Two factors that are central in the design and deployment of sensor networks are (1) low duty-cycle of the nodes coupled with clock synchronization; and (2) end-to-end delay in data delivery. With clock divergence of different nodes, clock synchronization becomes an important requirement since the communication times need to be synchronized. Clock synchronization in the context of sensor networks has been extensively studied and many relevant issues have been thoroughly explored [23, 21]. Various approaches for local time synchronization include, but are not limited to, passing time-stamp messages [18], third-party event-based synchronization [11], and Reference Broadcast Systems [12], where no time-stamps are used but arrival times at different nodes are compared. In terms of global network synchronization, some diffusion protocols have been proposed to diffuse the time through the entire network [22, 15]. Since global synchronization of the clocks of all nodes requires extensive overhead, it is usually sufficient for a node to track deviations from its own clock, rather than to compare the clocks of all of its neighbors (the nodes with which it communicates). The divergence of clocks is not necessarily a bad characteristic, as it reduces channel contention due to near-simultaneous data transmissions. On the other hand, in a general multi-hop topology, the delay of each hop of a message on its path to the base-station is very long, meaning that the idle periods of nodes on the message's path are very high. This implies that having a large number of hops is undesirable, as the end-to-end delay (the delay between sensing an event and notification of the base-station) is significant. Moreover, the clock synchronization process may involve complicated scheduling of the nodes, which in turn reduces the energy efficiency of the network. Thus, minimizing clock synchronization processes in the network will contribute to more efficient data collection and better network lifetime.

The discussion above motivates our study: It indicates that shallow tree topologies enjoy these benefits: (i) the number of hops a message traverses is bounded by the height of the tree, which is small, while clock synchronization is easier, and (ii) each node only needs to keep track of the deviations between its own and its parent's clock. In terms of a theoretical perspective, it is intriguing to see for which topologies it is possible to obtain polynomial-time algorithms.

Contributions. In this paper, we study shallow tree topologies where the trees are limited in height, and then we compare these to other approaches. We refer to a tree of height k as a k-tree. We present several contributions: (1) In the context of 2-trees, we present some polynomial-time algorithms for finding an optimal location of the base-station and an associated optimal transmission scheme; (2) We show that the above problem is NP-Hard for trees whose height are 3 or more; (3) We formulate a linear programming problem that finds an optimal topology to maximize the lifetime of the system while guaranteeing that each node's messages reach the base-station within two hops. We call this topology the *Constrained LP* **(CLP)**; and, finally, (4) We have conducted extensive simulations comparing the lifetime and number of hops of the 2-trees algorithm with other approaches (which we also present).

Related Work

Efrat *et.al.* [9] addressed the problem of finding an optimal base-station location, along with a transmission scheme for the sensor nodes under the objective of maximizing lifetime. They split the problem into two parts: (i) for a given base-station location, finding an optimal transmission scheme, and (ii) finding an optimal base-station location by using part (i) over a discrete set of locations. They show that problem (i) can be expressed as a linear programming problem, which can be solved in polynomial time. For problem (ii), they show that restricting the base-station to the locations of the sensor nodes achieves a constant-factor approximation for maximizing lifetime and present a $(1-\varepsilon)$-approximation by discretizing the search space into a limited number of points. In general, Shi *et al.* [20] provide a framework for general base-station location problems with different objectives and show that their approach achieves a $(1-\varepsilon)$-approximation for the objective of maximizing lifetime.

In [17], Pan *et al.* have considered the network to consist of clusters of nodes, with three types of nodes: *sensor nodes (SN)*, which generate and capture some real-time data, *application nodes (AN)*, which are responsible for collecting data from SNs, and a *base-station (BS)*, which collects data from all ANs. Thus, the network has a two-tier hierarchical topology, and the aim is to maximize the lifetime, where several different measures are considered. They have introduced multiple definitions for topological lifetime, based on the criticality of the network: if any of the ANs dies, if K out of a total of M ANs die, or if a specific set of ANs dies. The conclusion of their paper is that the ANs are the most important nodes in the length of system lifetime. However, they do not obtain any theoretical bounds for the time complexity of the algorithm. Another interesting work by Buragohain *et al.* [7] deals with the sub-problem of finding

an optimal routing scheme to maximize the lifetime of the network when the base station's location is fixed. Using the assumption that transmission and reception of 1 bit of data uses 1 unit of energy irrespective of the distance of transmission/reception, they prove that the problem is NP-complete. Under their scheme, the number of levels of the transmission tree is not restricted.

Bogdanov *et al.* [6] tackle the problem of multiple base-station positions, where there are two equivalent objectives: (i) minimizing the recharging rate of the batteries of the sensors, while having a specified data generation rate, and (ii) maximizing the data generation rate, while respecting some fixed battery recharging rate. They aim to find a *base-station location layout* for different numbers of base-stations, by considering a regular grid of unit cells in the sensor network region.

Some works also consider a model of a mobile base-station. Hou *et al.* [19] have studied the extension of the life of the network by moving the base-station to different locations. In their model, as is typically the case, the base-station has an unlimited supply of power. The idea is that since the sensors around the base-station consume more energy than others (because they work as the relay for other sensors), one can move the base-station to different locations, to balance the battery consumption among all sensors. They considered two versions of the problem: (i) The number of base-station locations is finite (C-MB), and (ii) The number of base-station locations is infinite (U-MB). They provide a polynomial-time algorithm for the C-MB problem and a $(1-\varepsilon)$-approximation algorithm for the U-MB based on discretizing the space into a set of finite locations. The interested reader may refer to the survey paper by Akkaya *et al.* [3] for a detailed study of different base-station positioning works in the literature categorized into different approaches.

2. PROBLEM FORMULATION

A set $S \subset \mathbb{R}^2$ of n sensors is deployed in the plane. The sensors have gathered some information, which must be transmitted to the *base-station* located at a given point $g \in \mathbb{R}^2$ in the plane. Assume that the amount of data generated by each sensor within the same time period is the same. A sensor can either transmit its data directly to g or forward (relay) the data via one other sensor. Each sensor transmits all of its data to the same node, and cannot split it between different nodes. The rationale for these assumptions is simple: Sensors commonly are deactivated for longer periods, and synchronizing the transmission times requires non-trivial handling and learning of clocks drifting. The fewer edges that exist, the less the resources that are needed for the synchronization.

The sensors $F \subset S$ that do not transmit their data directly to g are called *followers*; the sensors $L \subseteq S$ transmitting directly to g (possibly also forwarding the data from some followers) are called *leaders*. The links from followers to leaders and from leaders to the base-station form a tree of height 2, with g as the root, leaders on the first level, and followers on the second (leaf) level.

The *degree* $\mathrm{d}(l)$ of a leader $l \in L$ is defined as its degree in the tree, i.e., as 1 plus the number of followers relaying their data via l. We sometimes call the tree the *forwarding protocol* since it defines (protocols) on how the data is forwarded to the base-station.

Transmitting data a distance D requires energy proportional to D^α for some $2 \le \alpha \le 4$; assume without loss of generality that the coefficient of proportionality is 1. That is, if a follower $f \in F$ sends its data to a leader $l \in L$, then f spends energy $|fl|^\alpha$, where $|fl|$ is the Euclidean distance from f to l. Similarly, a leader l spends energy $\mathrm{d}(l)|lg|^\alpha$. The battery capacity of each sensor is normalized to 1 unit. For simplicity of exposition we assume that receiving the data comes at no cost; e.g., one can assume that the reception cost is included in that of sending. As will soon be apparent, this assumption can be changed easily.

Given a location of the base-station g, we consider two basic problems for finding an optimal data transmission scheme. The first problem is that in which each sensor has a finite amount of data to transmit. Given the locations of the sensors S and the base-station g, find the protocol such that the number of sensors that can transmit their data is maximized. Formally, denoting by $L(f)$ the leader to which a follower $f \in F$ is transmitting, our problem is

$$\max(|L| + |F|) \qquad \text{s.t.} \qquad \mathrm{d}(l)|lg|^\alpha \le 1 \quad \forall l \in L \quad \text{and}$$
$$|fL(f)|^\alpha \le 1 \quad \forall f \in F.$$

The second problem is that in which each sensor is continuously transmitting data at a certain rate. In this problem, the goal is to maximize the network's lifetime, i.e., the time t when the first sensor runs out of battery, while ensuring that all sensors are transmitting their data until that time. Being formal, the problem is

$$\max t \qquad \text{s.t.} \qquad t\mathrm{d}(l)|lg|^\alpha \le 1 \quad \forall l \in L \quad \text{and}$$
$$t|fL(f)|^\alpha \le 1 \quad \forall f \in F.$$

Finally, we consider the analogous problems for optimal base-station location. Specifically, we want to find the location of the base-station g and an associated transmission protocol such that: (1) the number of sensors that transmit their data to g is maximized, in the case of finite data transmissions; and, (2) the network lifetime is maximized, in the case of continuous data transmission. Note that this is an "all-or-nothing" setting: Each sensor either fully transmits its data to the base-station or does not transmit at all; there are no partial transmissions.

3. PRELIMINARIES

Maximum $\vec{b}$-matchings and augmenting paths with respect to them are central to our algorithms, so we review them here.

Matchings. A *matching* in a bipartite graph $G = (U \cup V, E)$ is a subset of edges such that every vertex is incident to at most one edge in the subset. A maximum-cardinality matching in G can be found in $O(|E|\sqrt{n})$ time, $n = |U \cup V|$, using the algorithm of Hopcroft and Karp [14]. Because our algorithms follow the classical schema of [14], we outline some details of the Hopcroft–Karp algorithm (HK, for short) and its analysis.

Let $M \subset E$ be a matching. An *augmenting path* is an odd-length path that starts and ends with an unmatched vertex and whose every other edge is not in M. An augmenting path exists if and only if M is not maximum, and the path can be found using a combination of breadth-first search (BFS) and depth-first search (DFS), as follows. The BFS starts from all unmatched vertices in U. It proceeds

along unmatched edges from U to V, and along matched edges when going from V to U. The BFS stops at a level when reaching one or more vertex in the set, $V' \subseteq V$, of unmatched vertices; at such a stopping point, an augmenting path has been discovered. The DFS's go back along the edges traversed during the BFS, starting from a vertex of V'; after an augmenting path is found, the path's vertices are removed, and a DFS is started from another vertex of V'. In this way, a set of *vertex-disjoint* augmenting paths is obtained; the edges along the augmenting paths are flipped (i.e., each edge in M is removed from M, while each edge not in M is added to M), resulting in a larger cardinality matching.

The analysis of the running time of the HK algorithm is based on the fact that after the kth BFS+DFS phase, all length-k augmenting paths are discovered. Moreover, at each phase at least one path is discovered; thus, the remaining number of phases is at most $|U \cup V|/k$. The running time follows from setting $k = \sqrt{n}$ and observing that one BFS+DFS phase takes linear time.

$\vec{b}$-Matchings. Let $\vec{b}$ be a vector of $|U \cup V|$ integers, with a component, $\vec{b}(v)$, associated with each vertex $v \in U \cup V$. A *$\vec{b}$-matching* is a subset $M \subseteq E$ of edges such that each vertex $v \in U \cup V$ is incident to at most $\vec{b}(v)$ edges of M. The concept of a $\vec{b}$-matching is a generalization of matchings. The HK algorithm and its analysis extends to finding a maximum-cardinality $\vec{b}$-matching: We say that a vertex v is *exposed* if its degree in M is less than $\vec{b}(v)$. The BFS starts from all exposed vertices in U, proceeding along unmatched edges, and continues along matched edges when going from V to U. The BFS stops upon reaching one or more vertex in the set of exposed vertices of V, at which time the DFS starts. The DFS proceeds verbatim as for the case of matchings. Finally, edges are flipped along the alternating paths obtained.

The correctness of the extension of the HK algorithm to $\vec{b}$-matching follows from the fact that, analogously to matchings, the symmetric difference between two $\vec{b}$-matchings is a set of alternating paths and cycles. The running time analysis is identical to the analysis for matchings, establishing the following result (which we believe has been known before, although we did not find it in the literature):

THEOREM 1. [Likely folklore] *A maximum-cardinality $\vec{b}$-matching in a bipartite graph can be found in $O(|E|\sqrt{n})$ time.*

Maintaining a maximum $\vec{b}$-matching.

Suppose we are given a maximum $\vec{b}$-matching M in G, and suppose that for some vertex $v \in V$, $\vec{b}(v)$ is increased by 1. We can update M efficiently to get the maximum $\vec{b}$-matching for the modified $\vec{b}$. For that, observe that an augmenting path π (if one exists) in G must, without loss of generality, start from v (or else M was not maximal). One BFS+DFS is enough to find π and to flip edges along it. The obtained matching is necessarily optimal in the modified instance, since there are no other augmenting paths, as we now argue. Indeed, if an augmenting path π' still exists, it must use some edges from π (or else M was not maximum). Let a be a vertex were π' meets with π. The subpath of π' from v to a, concatenated with π from a to one of π's endpoints, is an augmenting path also in the original instance (G with M), contradicting maximality of M.

Similarly, suppose that $\vec{b}(v)$ is decreased by 1. As above, we can update M efficiently to get a maximum $\vec{b}$-matching for the modified $\vec{b}$. Indeed, if M is still feasible, it is also optimal. Otherwise, remove any edge $uv \in M$ incident to v from the matching. An augmenting path π (if one exists) in G must, without loss of generality, start from u (or else M was not maximal). As above, one BFS+DFS phase finds the path and flips edges along it; also, as above, no other augmenting paths exist. Thus, we have

THEOREM 2. *After one component, $\vec{b}(v)$, of $\vec{b}$ is changed by ± 1, the maximum cardinality $\vec{b}$-matching can be updated with one BFS+DFS phase.*

4. COMPUTING AN OPTIMAL FORWARDING PROTOCOL

We are now ready to give algorithms for the data forwarding protocol, given a location of the base-station g, in order to (i) maximize the number of sensors able to forward their data when each sensor has a finite amount of data, and to (ii) maximize the network lifetime when each sensor is continuously transmitting data.

Consider the case of finite data per sensor. Splitting the sensors into followers and leaders is easy: If a sensor $s \in S$ is "far" from the base-station g, precisely if $|sg|^\alpha > 1$, then s cannot transmit to g, and, thus, $s \in F$. Moreover, by an exchange argument, in an optimal tree, any sensor $s \in S \setminus F$, without loss of generality, can transmit directly to the base-station. Thus, $L = S \setminus F$.

Define the *capacity* of a leader $l \in L$ to be

$$\vec{b}(l) = \min\{n, \lfloor 1/|lg|^\alpha \rfloor\} - 1.$$

The capacity shows how many followers can forward their data via l. Set $\vec{b}(f) = 1$ for all $f \in F$. Let $G = (L \cup F, E)$ be the bipartite graph on the leaders and followers whose edges connect a follower f to a leader l whenever f is close enough to l to transmit to it: $|fl|^\alpha \leq 1$. The crucial observation is the following.

OBSERVATION 3. *Finding the optimal protocol is equivalent to computing maximum-cardinality $\vec{b}$-matching in G.*

Thus, by Theorem 1, the optimal protocol can be found in $O(n^{2.5})$ time. In the remainder of this section, we show how to improve the running time to $O(n^{1.5} \log n)$ by exploiting the geometry of the problem. Our main ingredient in doing so is the following.

LEMMA 4. *The BFS can be performed in $O(n \log n)$ time.*

PROOF. Start the BFS from unmatched (exposed) vertices of F. Then the L-F step of the BFS is straightforward: just follow the matched edges from the reached vertices of L. We now describe how to perform the F-L step of the search. That is, we show how to do the following: Given a set of points $F' \subseteq F$ (exposed followers), find all leaders $L' \subseteq L$ such that a follower in F' can transmit to a leader in L'. This is equivalent to finding all leaders l such that the unit disk centered at l contains an exposed follower. We discover all such disks one by one, spending $O(\log n)$ time per disk. Upon discovery, the disk is deleted, so that every disk is discovered only once. Thus, overall, discovering and deleting the disks takes $O(n \log n)$ time, since there are $O(n)$

disks. The discovery of the disks is done by scrolling through the exposed followers: For each exposed follower $f' \in F'$, in $O(\log n)$ time, we either discover (and delete) a disk that contains f' or report that no such disk exists (in which case we proceed to the next follower). Using the data structure from the paper by Efrat *et al.* [10, Section 5.1], we can perform the discovery and deletion of disks in $O(\log n)$ time per disk. The structure has exactly the requisite properties: the capability to discover and delete one disk in $O(\log n)$ time. □

Now, as in the original HK algorithm (and Theorem 1), the BFS+DFS phase is invoked at most $2\sqrt{n}$ times. Clearly, each DFS takes $O(n)$ time. Thus, by Lemma 4,

THEOREM 5. *Given a set of sensors, each with a finite amount of data to transmit, and given the location of the base-station g, the optimal data forwarding protocol in order to maximize the number of sensors that can transmit to g can be found in $O(n^{1.5} \log n)$ time.*

Note that the number of edges in G can be quadratic in n. Nevertheless, we can find a maximum $\vec{b}$-matching in subquadratic time; of course, this is achieved by not building the graph explicitly. Now, consider the case in which each sensor is continuously transmitting data. It is useful to consider a variant of this problem, which is used as a subroutine in the algorithm for optimal location of the base-station: Given a target lifetime t and a location of the base-station g, determine whether there exists a protocol in which all sensors can continuously transmit their data to g for time t. In this case, if a sensor sends the data of k other sensors, it is spending $(k+1)|sp|^\alpha$ per unit time. The problem now becomes a decision problem satisfying the following constraints:

$$t\mathrm{d}(l)|lg|^\alpha \leq 1 \quad \forall l \in L, \qquad t|fL(f)|^\alpha \leq 1 \quad \forall f \in F.$$

The solution to this problem is analogous to that for finite data transmission. Hence, we get

THEOREM 6. *Given a target lifetime t and a location for the base-station g, we can decide if there exists a protocol in which each sensor continuously transmits data to g at a fixed rate in $O(n^{1.5} \log n)$ time.*

Hence, when maximizing lifetime, the objective is to find a protocol maximizing the time t before the first sensor runs out of battery while ensuring that all sensors send in their data for t units of time.

5. OPTIMIZING THE LOCATION OF THE BASE-STATION

The case of a fixed t. In the previous section it was assumed that the location of the base-station g is given. Next we consider the corresponding problem of finding an *optimal location* g. Given the sensors S, with a finite amount of data to transmit, we show how to find a location g for the base-station and the tree T, maximizing the number of sensors from S that can transmit their data to g. We also address the variant of the problem for continuous data transmission by the sensors: find a location g and the tree T, maximizing the network lifetime during which all sensors can transmit to the base-station without running out of battery. To do so, we first assume that t is known (so the data generated by each sensor is known), and refer to this problem as a *decision problem*, or an oracle. This will later be used to find the maximum possible t.

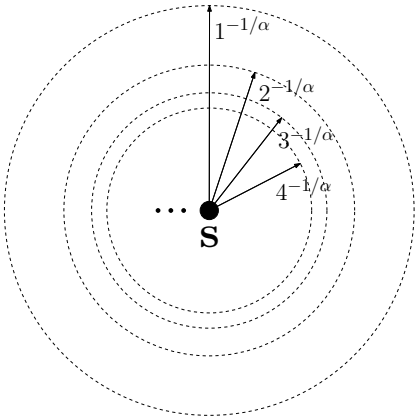
This problem can be solved by observing that, without loss of generality, it is enough to consider only a polynomial number of candidate locations for the base-station. Specifically, consider circles of radii $1^{-1/\alpha}, 2^{-1/\alpha}, \ldots, (n-1)^{-1/\alpha}$ centered at each sensor from S; the i^{th} circle is the boundary of the region to which the sensor can transmit its data and the data from $i-1$ additional sensors (see Figure 1a). Let A denote the arrangement of the circles. Then, without loss of generality, the base-station can be placed at a vertex of the arrangement (see Figure 1b). Indeed, if g is not at a vertex, one can move g while not intersecting any circle, and still ensure that every sensor can still transmit the necessary data to g. There are $O(n^4)$ vertices in A; for each we can invoke the algorithm from Theorem 5, and choose the overall best vertex. Thus, finding an optimal location for the base-station, and a corresponding forwarding protocol, can be done in $O(n^{5.5} \log n)$ time. In order to improve the running time, we note that the problem does not have to be solved "from scratch" at every vertex of A. Instead, we start by computing the optimal protocol T_p for g being at an arbitrary vertex p of A. Then, we move g to an adjacent vertex, q. Suppose that p is the intersection of circles centered at some sensors i, j, and that q is the intersections of circles centered at i, k (possibly, $k = j$). As g is moved from p to q, the only changes in the graph $G = (F \cup L, E)$ concern the sensors j and k: they change their capacity by 1 (including the possibility of switching between F and L). Thus, by Theorem 2, the optimal protocol T_q can be obtained from T_p by one BFS+DFS phase, which takes $O(n \log n)$ time by Lemma 4. Hence, in overall $O(|A|)$ steps we can walk over all vertices of A, moving between adjacent vertices, spending time $O(n \log n)$ per vertex to update the optimal protocol:

THEOREM 7. *An optimal location of the base-station, and a corresponding forwarding protocol, can be computed in time $O(n^5 \log n)$.*

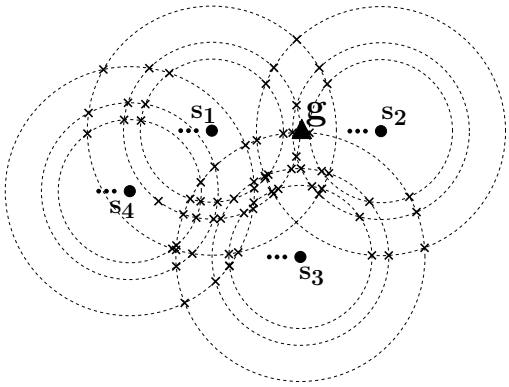
LEMMA 8. *Given a target lifetime t, we can decide in time $O(n^5 \log n)$ whether or not there exists a location of the base-station g and an associated transmission protocol in which each sensor can continuously transmit data to g for a duration t.*

Maximizing the lifetime. Here we use the procedure of Lemma 8 as an oracle, $\mathcal{A}(\sqcup)$ that determines for a specific t if it is larger, smaller or equal to the maximum lifetime, denoted by t_{OPT}.

One feasible solution is to set $t = 1/|sg|^\alpha$ and have all sensors transmit directly to g; here s is a sensor farthest from g. This, in fact, is a $1/2^\alpha$-approximation because either s or its leader has to transmit to distance at least $|sg|/2$. We can increase the approximation factor to $(1-\varepsilon)$, for any $\varepsilon > 0$, by a binary search: given an interval $[a, b]$ for t (which means having a b/a-approximation), query the point $\sqrt{ab}$ to get a $\sqrt{b/a}$-approximation. Since we start from $b/a = 2^\alpha$, $\log(\alpha \ln 2 / \varepsilon)$ steps are enough; hence, the maximum lifetime can be $(1-\varepsilon)$-approximated in $O(n^5 \log n \log \frac{1}{\varepsilon})$ time (for any $\varepsilon > 0$). For finding an exact solution, we can use the following approach, which is based on the parametric search method of Megiddo [16]. The main idea here is to imagine a slow process of increasing t from 0 until it exceeds



(a) *The circles around sensor s of radii $1^{-1/\alpha}, 2^{-1/\alpha}, \ldots, (n-1)^{-1/\alpha}$. The i^{th} circle is the boundary of the region to which the sensor can transmit its data and the data of $i-1$ additional sensors.*



(b) *The vertices of the arrangement of circles, which are the candidate locations of the base-station, are marked with $\times$. The optimal location of the base-station is at g.*

Figure 1: The circles around each sensor and the corresponding arrangement.

t_{OPT}, while keeping track of the vertices of the arrangement of disks discussed above. Special events occurs when two vertices coincide and switch their order along the boundary of a third disk. So we can use the parallel sorting networks of AKS [2], as described by Cole [8], as the generic algorithm. We refer the reader to [16] for further reading.

THEOREM 9. *An optimal location of the base-station and a corresponding data transmission protocol can be found in $O(n^5 \log^2 n)$ time.*

Remark: It is not unreasonable to assume that a leader would relay data only from a small constant K of other sensors. In addition, even if this is not the case and a sensor s does relay data from K other sensors, the boundaries of the disks centered at s corresponding to larger values of K tend to cluster very close to each other around s, and distinguishing between them is probably unnecessary, implying that the complexity of Theorem 9 drops to $O(n^3 K^2 \log^2 n)$.

6. NP-HARDNESS FOR 3-LEVEL TREES

In the previous sections, we considered finding optimal protocols with at most 2 levels. We now justify our focus on only 2-level trees by proving that if we are allowed to have 3 or more levels, our problems become NP-hard. We show that the problem is hard even in the case of finite data transmission from a sensor (i.e., there is no issue of maximizing system lifetime). This implies that the problem is NP-hard in the case of continuous data transmission as well. As before, the only rule is that a sensor s can transmit to a point p if $\mathrm{d}(s)|sp|^\alpha \leq 1$, where $\mathrm{d}(s)$ is the total number of sensors (including s itself) whose data is forwarded by s.

Our decision problem is: Given S, g, does there exist a protocol (with possibly arbitrarily many levels) enabling all sensors in S to transmit their data to g? To show the hardness, we reduce from 3PARTITION: given a set of integers $a_1 \ldots a_{3n}$, decide if it is possible to split them into n groups of 3 so that the sum of the integers in every group is the same. The problem is hard even if each integer is between $B/4$ and $B/2$, and $B = O(1)$; here $B = \sum a_i/n$ [13]. Given such an instance of 3PARTITION, we create an instance S, g of the forwarding problem as follows (Fig. 2).

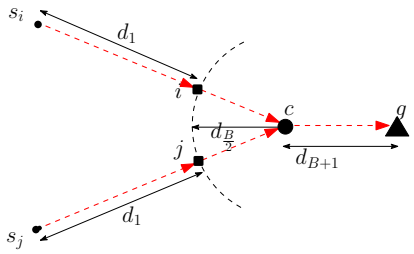


Figure 2: c is a cluster-head; i, j are two elements. s_i, s_j are atoms associated with i and j.

The base-station g is at the origin. The sensors are of 3 types: n *cluster-heads*, $3n$ *elements*, and B *atoms*. For an integer m, let $d_m = m^{-1/\alpha}$ denote the distance for which the information of m sensors can be forwarded. We place all cluster-heads on the x-axis at distance d_{B+1} to the left of g.

The elements correspond to the integers in the 3PARTITION. The $3n$ elements are placed on the semicircle of radius $d_{B/2}$ centered at the cluster-heads; the whole semicircle is to the left of the cluster-heads (we elaborate on the exact placement of the elements below). For each element i, we place $a_i - 1$ atoms at the point at distance d_1 from i on the ray from the cluster-heads through i. If the 3PARTITION instance is feasible, the sensors can transmit the data as follows: The atoms transmit to their elements. The elements are split into n groups of 3 and each group transmits to one of the n cluster-heads. Finally, the cluster-heads transmit to g.

It is easy to see that the above protocol is the only possible solution, for appropriately chosen locations of the elements (described below). First, the data from an atom can reach g only via its element; otherwise, if an atom s transmits to an atom s', the latter will not be able to forward its and s's data via its element. Each element i has to transmit a_i units of data ($a_i - 1$ from its atoms plus its own unit). Now,

in our construction above, we are free to place the elements *anywhere* on the semicircle. We choose to place them very close to the x-axis, so that the distance from any element to g is (almost) $d_{B+1} + d_{B/2}$. Because $a_i > B/4$, no element can transmit directly to g as soon as:

$$d_{B+1} + d_{B/2} > d_{B/4} \quad (1)$$

Assuming (1) holds, the data from any element cannot reach the base-station directly, and has to go via a cluster-head. No cluster-head can transmit more then $B + 1$ units of data, and hence elements have to be split evenly among the cluster-heads.

THEOREM 10. *Deciding whether there exists a protocol (with possibly arbitrarily many levels) enabling all sensors to transmit their data to the base-station, is NP-hard.*

PROOF. We only have to show that (1) holds. This follows from simple arithmetic:

$$\begin{aligned}
(1) \Leftrightarrow \quad & (B+1)^{-\frac{1}{\alpha}} + \left(\frac{B}{2}\right)^{-\frac{1}{\alpha}} > \left(\frac{B}{4}\right)^{-\frac{1}{\alpha}} \\
\overset{B\geq 1}{\Longleftarrow} \quad & (2B)^{-\frac{1}{\alpha}} + \left(\frac{B}{2}\right)^{-\frac{1}{\alpha}} > \left(\frac{B}{4}\right)^{-\frac{1}{\alpha}} \\
\Leftrightarrow \quad & 2^{\frac{3}{\alpha}} - 2^{\frac{2}{\alpha}} - 1 < 0 \qquad \Leftarrow \quad \alpha \geq 2
\end{aligned}$$

□

7. EXPERIMENTAL RESULTS

In this section, we describe our simulations on finding the optimal data transmission scheme under various parameters while measuring various metrics.

Simulation Setup. In all simulations, we consider a square sensor field of side-length 10 units, in which the base-station is placed at the center. The parameters of the experiments are the number of sensors n and the *Rayleigh fading factor* denoted by α. The sensors are placed randomly in the field by using a uniform random location generator. We have conducted experiments for $n = 10$ to $n = 150$ in steps of 10 and for $\alpha = \{2, 3, 4\}$. For each combination of the parameters, each experiment is repeated 20 times and the results are averaged. For all experiments, the cost function of sending a unit of data, from s_1 to s_2 is calculated as $cost(s_1, s_2) = max\{c_{min}, |s_1, s_2|^\alpha\}$ in which, the parameter c_{min} reflects that even if $|s_1|$ and $|s_2|$ are very close, a certain minimum energy is still required. We set $c_{min} = 1$.

Algorithms. Our experiments consist of implementation and comparison of three algorithms for finding a data forwarding protocol:

Two-level Tree (TT): This is the main algorithm that we have introduced in Section 4.

Linear Programming (LP): The problem of determining an optimal forwarding protocol for the system without the restrictions on splitting the data and the path length to the base-station can be formulated as a Linear Programming problem, as in [9, Section 2.1.2].

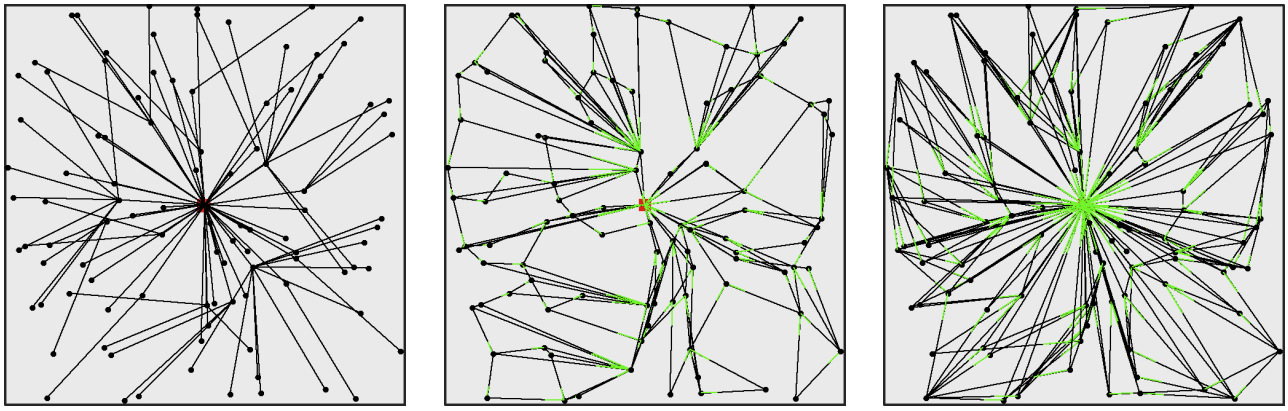
Constrained LP (CLP): In the LP approach, there is no guarantee that the path length of a data packet is small; a data packet might have many hops before reaching the base-station. Since we want the sensors to remain idle for the maximum possible time, the resulting multi-hop paths may introduce significant delays in transmitting data to the base-station. One approach to this issue is to check in the LP what is the average and maximum delay obtained. Another approach is to further constrain the LP, by adding constraints to guarantee that every data unit in all nodes, reaches the BS in at most 2 hops. In the resulting LP problem, referred to as the *Constrained LP (CLP)*, we force each sensor to send its incoming data directly to the base-station, but we let it to freely select the next hop for its own generated data.

In our simulations, we are primarily interested in the *Lifetime* of the system defined as the length of time from the initiation of system operation until the first sensor runs out of battery. We have also studied two other metrics: *Out-degree* and *Average Hop Count*. See discussion in the introduction about the rules of these parameters in the network's performances, when activation and deactivation (awake and idle states) are needed. The *Out-degree*, which is the average number of receivers to which a node transmits its data and the *Average Hop Count (AHC)* is hop counts for a data packet from its origin sensor to the base-station averaged over all sensors. In the case of the LP and CLP algorithms, the average hop count for a sensor node is calculated by computing by recursively using the average hop counts of its parents (nodes it forwards data to), and by computing a sum of these weighted with the amount of data transmitted to each parent.

In order to better compare the three algorithms, we wrote a Java applet [1] enabling the user to visualize the resulting tree, i.e., the forwarding protocol. The applet allows the user to input a set of sensor locations and visualize all three algorithms. Some interesting results are shown in Fig. 3. The black disks and the red squares represent the sensors and the base-station respectively. Moreover, there is an edge connecting nodes n_1 and n_2 if and only if n_1 transmits some data to n_2. In Fig. 3a, all edges are directed towards the base-station. In Figs 3b and 3c, the green end of an edge indicates its direction.

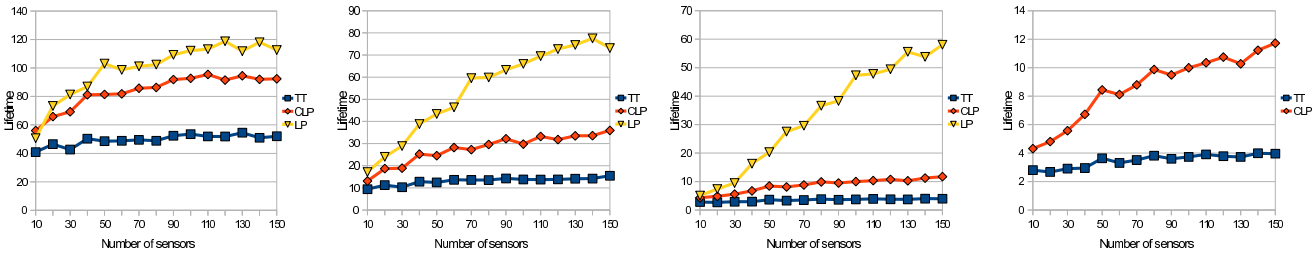
Discussion of Results. As can be seen in Fig. 3, the TT algorithm produces quite a sparse tree. The LP approach introduces more edges in the graph, and the average hop-count of a data unit is higher, in comparison with TT. Not surprisingly, LP results in some nodes whose out-degrees are large, with quite a few nodes at 5 hops or more away from the BS. For the CLP algorithm, however, the edges are more directed towards the base-station, in comparison with LP, but still the density of the edges is high compared with TT.

Fig. 4 shows the simulation results for *Lifetime* for the different algorithms and parameters. In addition, Fig. 4d shows the same results as Fig. 4c, with the exception that the results for LP are omitted, in order to make the comparison of TT and CLP clearer. As can be seen, the gap between the TT algorithms and the other two methods increases with increasing α. This can be attributed to the fact that as we increase α, the advantages in splitting data from a node are more apparent, making more complicated, but also more power efficient, paths viable. In the case of $\alpha = 2$, the graph of the LP and CLP seems to approach saturation. This leads us to infer that these algorithms perform only a constant factor better than the TT algorithms without the added benefits of synchronization, leading to better end-to-end delay. It is interesting that this does not seem to hold



(a) *Two-level tree* (b) *LP* (c) *Constrained LP*

Figure 3: Instances of the Two-level Tree algorithm (TT), the linear programming method (LP), and the Constrained LP method (CLP)



(a) $\alpha = 2$ (b) $\alpha = 3$ (c) $\alpha = 4$ (d) $\alpha = 4$

Figure 4: The lifetime of the system as a function of α.

for greater values of α. In particular, in Fig. 4d, we see that the TT algorithm is perhaps not as effective as the CLP algorithm, which has better lifetime. One area of future research lies in exploring the effect of α on the lifetime.

		α		
		2	**3**	**4**
CLP/TT	***avg***	1.69	2.09	2.45
	max	1.84	2.42	2.96
LP/TT	***avg***	1.99	4.03	9.13
	max	2.32	5.46	14.91

Table 1: The average and maximum ratios of lifetime using methods CLP and LP to that using TT.

The average and maximum of the ratios of system lifetime in CLP and LP to system lifetime in TT are shown in Table 1. These values also seem to support the above observations.

In Table 2 we show the average and maximum out-degree values for different scenarios, when 100 sensors are deployed in the field and $\alpha = 4$. As expected, the LP has the highest out-degrees of the three algorithms. The higher the out-degree, the more complicated is the forwarding scheme, making synchronization between nodes more complicated. Hence, the TT algorithm is seen to be best in this respect.

The simulation results for AHC are shown in Fig. 5. The second row basically shows the same results of the first row, without the results of LP algorithm in order to better compare TT and CLP. The average and maximum ratios of system hop-count in CLP and LP algorithms over TT are shown in Table 3.

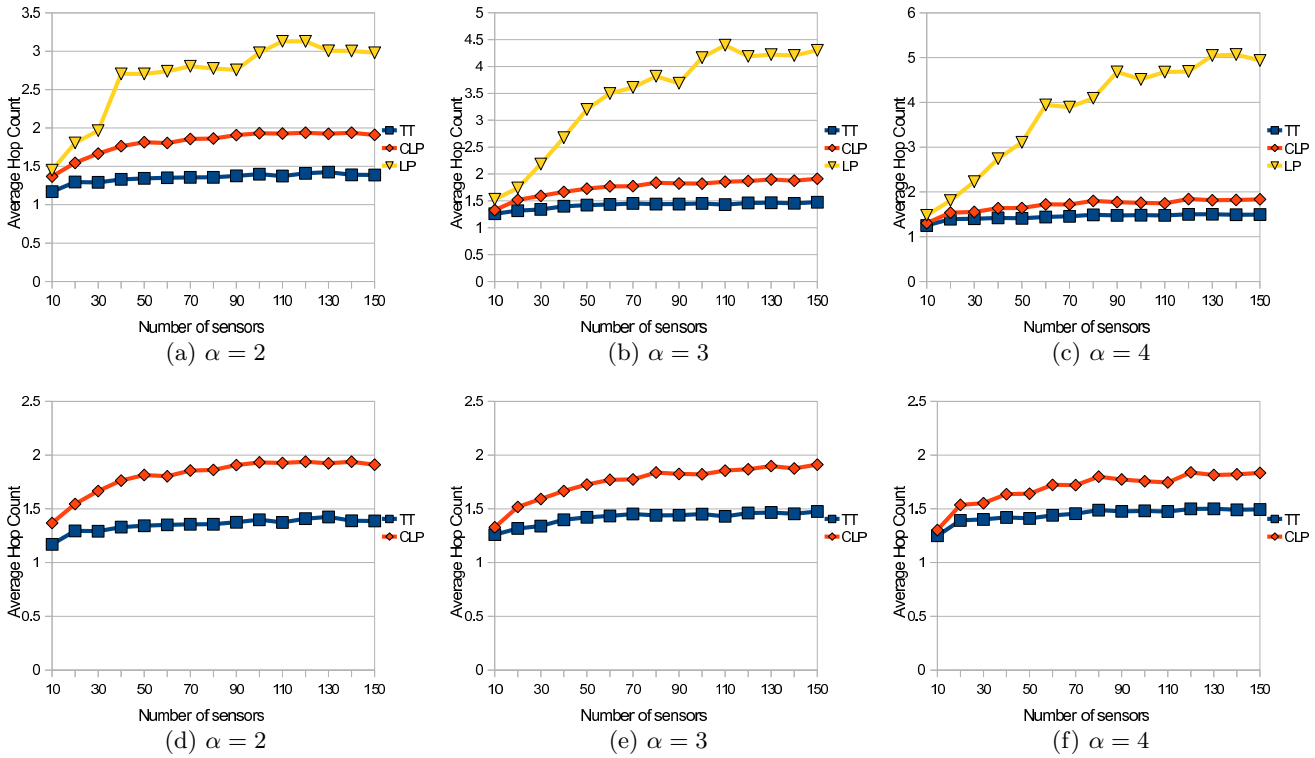
	TT	**LP**	**CLP**
average out-degree	1	2.48	1.95
max out-degree	1	8	4

Table 2: average and maximum out-degree values of nodes, using different algorithms

		α		
		2	**3**	**4**
CLP/TT	***avg***	1.34	1.23	1.17
	max	1.4	1.3	1.23
LP/TT	***avg***	1.96	2.4	2.6
	max	2.28	3.08	3.4

Table 3: The average and maximum hop-count ratios of CLP and LP algorithms over TT

Again, as in the case of lifetime, it appears that the gap in hop counts between LP and TT are higher for $\alpha = \{3, 4\}$ than for $\alpha = 2$. The TT algorithm has the lowest hop count, indicating that in terms of end-to-end delay, the TT algorithm has much better performance than the LP algorithm. Again, the gap between TT and CLP is not very wide. However, since, in both of these algorithms, the data is guaranteed to reach the base-station in two hops, it appears that



(a) $\alpha = 2$ (b) $\alpha = 3$ (c) $\alpha = 4$

(d) $\alpha = 2$ (e) $\alpha = 3$ (f) $\alpha = 4$

Figure 5: The average hop count of a data packet as a function of α. In the second row, we have omitted LP results, in order to clarify the comparison of TT and CLP.

the CLP algorithm is reaching the maximum point of two hops with an increasing number of sensors, whereas the TT algorithm stays at approximately 1.5 hops per sensor, with increasing α. This may indicate that, in practice, no more than half the nodes are designated as cluster-heads in the TT algorithm.

In summary, the simulation results indicate that for the case of $\alpha = 2$, the deficiency in terms of lifetime of the TT algorithm in comparison with LP and CLP may be offset by the improved values of AHC, i.e., better average end-to-end delays. In particular, the LP and CLP algorithms are only a constant factor better than the TT algorithm in terms of lifetime. However, for higher values of α, this is not so apparent. The CLP algorithm appears to be more effective than the TT algorithm and seems to be the best choice for higher values of α.

8. ACKNOWLEDGMENTS

E. Arkin and J. Mitchell are partially funded by the National Science Foundation (CCF-0729019, CCF-1018388). Alon Efrat is supported by NSF CAREER grant 0348000.

9. REFERENCES

[1] `http://www.cs.arizona.edu/people/taheri/?tab=research.`

[2] M. Ajtai, J. Komlós, and E. Szemerédi. An $0(n \log n)$ sorting network. In *Proc. 15th Annual ACM Symposium on Theory of Computing*, pp. 1–9, 1983.

[3] K. Akkaya, M. Younis, and W. Youssef. Positioning of base stations in wireless sensor networks. *IEEE Communications Magazine*, 45(4):96–102, 2007.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[5] T. Arampatzis, J. Lygeros, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proc. 13th Mediterranean Conference on Control and Automation*, pp. 719–724, 2005.

[6] A. Bogdanov, E. Maneva, and S. Riesenfeld. Power-aware base station positioning for sensor networks. In *Proc. 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, 2004.

[7] C. Buragohain, D. Agrawal, and S. Suri. Power aware routing for sensor databases. In *Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1747–1757, 2005.

[8] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *Journal of the ACM*, 34(1):200–208, 1987.

[9] A. Efrat, S. Har-Peled, and J. Mitchell. Approximation algorithms for two optimal location problems in sensor networks. In *Proc. 2nd International Conference on Broadband Networks*, pp. 714–723, 2005.

[10] A. Efrat, A. Itai, and M. J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31:1–28, 2001.

[11] J. Elson and D. Estrin. Time synchronization for wireless sensor networks. In *Proc. 15th International*

Parallel & Distributed Processing Symposium, page 186, IEEE Computer Society, 2001.

[12] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Operating Systems Review*, 36(SI):147–163, 2002.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, 1979.

[14] J. E. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

[15] Q. Li and D. Rus. Global clock synchronization in sensor networks. *IEEE Transactions on Computers*, 55(2):214–226, 2006.

[16] N. Megiddo. Linear-time algorithms for linear programming in r^3 and related problems. *SIAM Journal on Computing*, 12:759, 1983.

[17] J. Pan, L. Cai, Y. Hou, Y. Shi, and S. Shen. Optimal base-station locations in two-tiered wireless sensor networks. *IEEE Transactions on Mobile Computing*, 4(5):458–473, 2005.

[18] K. Römer. Time synchronization in ad hoc networks. In *Proc. 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp. 173–182, New York, NY, 2001.

[19] Y. Shi and Y. Hou. Theoretical results on base station movement problem for sensor network. In *Proc. 27th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 8, pp. 1–5, 2008.

[20] Y. Shi, Y. Hou, and A. Efrat. Algorithm design for base station placement problems in sensor networks. In *Proc. 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*, page 13, ACM, 2006.

[21] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *Network, IEEE*, 18(4):45–50, jul. 2004.

[22] W. Su and I. F. Akyildiz. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 13(2):384–397, 2005.

[23] B. Sundararaman, U. Buy, and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

[24] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.