Lifetime Maximization for Multicasting in Energy-Constrained Wireless Networks

Patrik Floréen, Petteri Kaski, Jukka Kohonen, and Pekka Orponen, *Member, IEEE*

***Abstract*—We consider the problem of maximizing the lifetime of a given multicast connection in a wireless network of energy-constrained (e.g., battery-operated) nodes, by choosing ideal transmission power levels for the nodes relaying the connection. We distinguish between two basic operating modes: In a *static* power assignment, the power levels of the nodes are set at the beginning and remain unchanged until the nodes are depleted of energy. In a *dynamic* power schedule, the powers can be adjusted during operation.**

We show that while lifetime-maximizing static power assignments can be found in polynomial time, for dynamic schedules the problem becomes NP-hard. We introduce two approximation heuristics for the dynamic case, and experimentally verify that the lifetime of a dynamically adjusted multicast connection can be made several times longer than what can be achieved by the best possible static assignment.

***Index Terms*—Ad hoc networks, computational complexity, energy-awareness, lifetime, multicasting, optimization.**

I. Introduction

WIRELESS "ad hoc" communication networks, consisting of a collection of radio transceivers with no prearranged infrastructure, have been studied intensively during the past few years [1]–[3]. The general area of topology control in such networks [4] is concerned with assigning appropriate transmission power levels to the transceivers so that some desired topological property holds. A natural problem in this area is the maintenance of a multicast connection, i.e., a transmission graph connecting a given source node to a group of sink nodes.

Battery power is a serious limiting constraint in many applications of ad hoc networks, and accordingly much attention has been paid to energy-efficient designs in this area. In a wireless transceiver, power is used for transmitting and receiving data, internal data processing, and simply for being "on" in an idle mode. The power required for transmission and reception, however, dominates [5], [6]. For simplicity, we consider only the power required for transmission, as in, e.g., [5] and [7]–[9], and disregard some physical layer issues such as interference.

We assume that transceiver i can communicate directly to transceiver j, if the transmission power of i exceeds a threshold value d_{ij}, usually approximated as $c \cdot r_{ij}^{\alpha}$, where c is a constant, r_{ij} is the physical distance between nodes i and j, and the exponent α, $2 \leq \alpha \lesssim 4$, models the decay of the radio signal in the intervening medium. In reality, the actual transmission powers and threshold values d_{ij} depend on several environmental and technological factors [12], [13]. We make no constraining assumptions on the values of d_{ij}.

A number of recent papers (e.g., [9], and [14]–[23]) have considered the topic of energy-aware broadcasting in wireless networks, i.e., the problem of maintaining a transmission graph connecting a given source node to all the other nodes. However, with rare exceptions such as [9] and [20], all of these works (as well as many older ones, e.g., [24] and [25]), take it as their goal to minimize the total power consumption of the entire network, i.e., they address the wireless analogue of the classical problem of finding a minimal spanning tree for a network. (Interestingly, in wireless networks this "minimum power broadcast" problem turns out to be NP-complete, as proved in, e.g., [14], [16], and [18].) Some of the papers, e.g., [18], [19], [22], and [23] address also the problem of constructing minimum multicast trees under the same metric.

However, minimal total power consumption does not guarantee maximum lifetime for a network (either for broadcast or multicast), as has been noted for instance in [26] and [27]. In this paper, we address the task of maximizing the network lifetime directly, and also take into account that different nodes may have different initial energy supplies.

Recent work closest to ours consists of the articles [8]–[11]. Our network model and the basic ideas for dealing with the static power assignment case are adapted from [8]. In [9], the problem of broadcast lifetime maximization using both static and dynamic power schedules is introduced, and a graph-theoretic polynomial time algorithm for the static problem is presented in [10]. Reference [11] introduces a distributed heuristic for finding long-lived static multicast trees.

The rest of the present article is organized as follows. In Section II, we formulate our model and the problem of multicast lifetime maximization. Furthermore, we note that in the static setting, an optimal power assignment can be determined in polynomial time. In Section III, we turn to dynamic power schedules, which are the main topic of this paper. We show that multicast lifetime maximization now becomes NP-hard, but remains polynomial time solvable for a fixed number of sink nodes. Whether the problem has a polynomial-time approximation algorithm with bounded, or even polylogarithmic performance ratio, is an

Manuscript received October 14, 2003; revised July 27, 2004. The work of P. Floréen and J. Kohonen was supported in part by the Academy of Finland under Grant 202203. The work of P. Kaski was supported in part by the Academy of Finland under Grant 202205 and in part by the Foundation of Technology, Helsinki, Finland (Tekniikan Edistämissäätiö). The work of P. Orponen was supported in part by the Academy of Finland under Grant 204156.

P. Floréen and J. Kohonen are with the Helsinki Institute for Information Technology (HIIT), Basic Research Unit, Department of Computer Science, University of Helsinki, FIN-00014 Helsinki, Finland (e-mail: Patrik.Floreen@cs.helsinki.fi; Jukka.Kohonen@cs.helsinki.fi).

P. Kaski and P. Orponen are with the Laboratory for Theoretical Computer Science, Department of Computer Science and Engineering, Helsinki University of Technology, FIN-02015 Helsinki, Finland (e-mail: Petteri.Kaski@hut.fi; Pekka.Orponen@hut.fi).

Digital Object Identifier 10.1109/JSAC.2004.837342

open problem connected to some well-known open problems in combinatorial approximation theory [28]–[30].

In Section IV, we suggest two randomized approximation heuristics for finding good dynamic power schedules for multicasting, and extend them also to the related problems of broadcast lifetime maximization and group connectivity lifetime maximization. In Section V, we report on simulation experiments using these techniques, showing that optimal static power assignments can indeed be significantly improved upon by using dynamic power schedules. In Section VI, we conclude with a summary and some ideas for further research. For readability, proofs of the technical results are relegated to the Appendix.

A preliminary conference report on our work appeared in [31]. In the present version, the proofs of the complexity-theoretic results have been fundamentally changed to remove the assumption of discrete scheduling times made in [31]. Also, the variety of experimental tests has been increased, and the experiments have been extended to cover also the broadcast and group connectivity lifetime maximization problems. Since the publication of [31], similar results on the complexity and approximability of the multicast lifetime maximization problem have been independently announced in [7] and [32].

II. Definitions and the Static Case

We write $\mathbb{Q}_+$ for the set of nonnegative rational numbers. We model an ad hoc network as a complete directed graph, the *power threshold graph* $G = (V, A, d)$, where the n transceivers form the set of nodes $V = \{1, \ldots, n\}$, and the values $d_{ij} \in \mathbb{Q}_+ \cup \{\infty\}$ on each arc $(i, j) \in A$ are the power threshold values for transmission from transceiver i to transceiver j. Note that we include ∞ among the possible values to indicate that direct communication is impossible (due to, e.g., an impenetrable obstacle between the nodes).

Each transceiver usually carries a battery with limited energy supply. We denote the energy constraint for node i with $e_i \in \mathbb{Q}_+ \cup \{\infty\}$. Disregarding the details of battery technology, transmission at power p for a time t consumes energy $w = pt$.

A *power assignment* $p \in \mathbb{Q}_+^n$ in a power threshold graph G of n nodes associates to each node in G its transmission power value. A power assignment p induces a *transmission graph* G^p which includes only those arcs (i, j) for which the power level assigned to node i in p is at least d_{ij}, i.e., the arcs corresponding to direct communication from node i to node j. As a consequence, when the power is high enough to reach a certain node, some other nodes may be reached simultaneously with the same transmission. Node j may transmit with power different from i, so a direct connection from i to j does not imply the converse. Note that this is different from the symmetric model employed in, e.g., [33].

A *power schedule* consists of a sequence of N power assignments, together with their time allocations. Formally, a power schedule is a pair $(P, t) \in \mathbb{Q}_+^{n \times N} \times \mathbb{Q}_+^N$, where the kth column of the matrix P indicates the kth power assignment p_k, and the kth element of the vector t is the time allocated to p_k.

A power schedule is *static* if it consists of a single power assignment, and otherwise, *dynamic*. A schedule is *feasible*, if it respects the energy constraints of the nodes. A power schedule that maintains a particular topological property Π in the induced transmission graph is *valid* for Π.

In the *lifetime maximization problem* for property Π, the goal is to find a power schedule (P, t) maximizing the time $\sum_{k=1}^{N} t_k$ during which the schedule is feasible and valid for Π. The property Π of central interest to us is *multicast connectivity* (or briefly *multicasting*), i.e., the condition that the transmission graph contains a directed tree connecting a given source node (indexed as 1) to a set of given sink nodes. We also consider *broadcast connectivity* (*broadcasting*), which is the special case of multicast connectivity with all nodes except the source as sinks, and *group connectivity*, where a given set of "terminal" nodes W is strongly connected, i.e., the transmission graph contains a directed path from each $u \in W$ to each $v \in W$.

To summarize, the multicast lifetime maximization problem takes as input the power threshold graph, the energy constraints at the nodes, and the set of sinks, and gives as output a power schedule that maximizes the time during which there are paths from the source node 1 to the sinks, taking the energy constraints into account.

Lloyd *et al.* [8] give a polynomial time algorithm for optimizing static power assignments that finds, for any polynomially testable, *monotone* graph property Π (i.e., one that is conserved under addition of arcs), a static assignment valid for Π that minimizes the maximum power used at any individual node in the network. The algorithm is based on the observation that for a monotone graph property, it suffices to consider only power levels that correspond to the power thresholds d_{ij}, of which there are at most n^2 different ones. Thus, with a simple search over these values, one can find the optimal static power assignment in polynomial time. In our conference paper [31], we note that for monotone graph properties Π (in particular for multicast connectivity), static lifetime maximization is equivalent to static node power minimization. Thus, we obtain a polynomial-time algorithm for static multicast lifetime maximization as a corollary to the results in [8]. The argument is presented in detail in [31], and we do not repeat it here. As broadcasting and group connectivity are also monotone graph properties, we similarly get polynomial-time algorithms for the static versions of these problems.

III. Dynamic Power Schedules

We now turn our attention to the more difficult task of multicast lifetime maximization with arbitrary power schedules. In this context, one can see that the hardness of the problem lies in determining the power assignments that occur in an optimal power schedule. Once the power assignments are available, it is a standard linear programming (LP) problem [34], [35] to determine the optimal time to allocate to each assignment.

More precisely, suppose we are given N valid power assignments for the nodes. Denoting by p_{ik} the power assigned to node i in the kth power assignment, the lifetime maximization task restricted to these power assignments can be represented as the following LP problem over the variables t_k:

$$\begin{aligned} \text{maximize} \quad & \sum_k t_k \\ \text{subject to} \quad & \sum_k p_{ik} t_k \le e_i, \quad \forall i \in \{1, \ldots, n\}, \\ & t_k \ge 0, \qquad\qquad \forall k \in \{1, \ldots, N\}. \end{aligned} \tag{1}$$

The associated dual problem over the variables y_i is

$$\begin{aligned} \text{minimize} \quad & \sum_i e_i y_i \\ \text{subject to} \quad & \sum_i p_{ik} y_i \geq 1, \quad \forall k \in \{1, \ldots, N\}, \\ & y_i \geq 0, \qquad\quad \forall i \in \{1, \ldots, n\}. \end{aligned} \tag{2}$$

The hardness of the lifetime maximization problem arises from the fact that the power assignments are not given explicitly in the problem instance, but implicitly via the power thresholds d_{ij}, and by specifying the source and sink nodes. In this case, the total number N of different valid power settings can be exponential in n, which makes the problem hard to solve.

Theorem 1: The multicast lifetime maximization problem is NP-hard under polynomial-time Turing reductions. Moreover, the problem remains NP-hard for broadcasting.

In the conference version of this paper [31], we established a weaker version of Theorem 1, quantizing the time allocations t_k to multiples of some discrete unit. The current proof, which takes a different approach than that in [31] and relies on ideas arising from the context of fractional Steiner tree packing [36, Sec. 4], is presented in the Appendix. This theorem has also recently been announced, without proof, in [7].

For any fixed number of sink nodes, the multicast lifetime maximization problem is polynomial time solvable via the ellipsoid algorithm. This is due to the fact that the separation problem (see [34]) for the dual (2) reduces to the problem of finding a valid power assignment with the minimum total power (multicast total power minimization), which can be solved in polynomial time for a fixed number of sink nodes. This is because the multicast total power minimization problem is polynomial-time equivalent to directed Steiner tree minimization, which is solvable in polynomial time for a fixed number of sink nodes, but NP-hard for an arbitrary number of sink nodes. However, we omit the details of this approach (see [36, Th. 4.1] for an analogous situation in the context of fractional Steiner tree packing), because recently a more practical algorithm has been given in [7], using techniques from [37].

As far as approximability is concerned, it is noted in [7] that multicast total power minimization is approximation-preserving equivalent to directed Steiner tree minimization. Furthermore, given a polynomial time algorithm for multicast total power minimization with approximation ratio α, the techniques in [7] and [37] give a polynomial time algorithm for multicast lifetime maximization with approximation ratio $(1+\epsilon)\alpha$ for any $\epsilon > 0$.

For directed Steiner tree minimization it is known that no polynomial-time approximation algorithm can achieve a performance ratio of $\Omega(\log^{2-\epsilon} n)$ for any $\epsilon > 0$, unless $\text{NP} \subseteq \text{ZTIME}(n^{\text{polylog}(n)})$, and the existence of a polylogarithmic-ratio approximation algorithm remains an open problem [29], [30].

A theoretically interesting open question that remains is to what extent lifetime maximization and total power minimization are "equivalent" with respect to polynomial time approximability; that is, is it also the case that an approximation algorithm for lifetime maximization would give an approximation algorithm for total power minimization.

Theorem 1 shows that multicast lifetime maximization with arbitrary power schedules is NP-hard, but still leaves open the possibility that more restricted versions of the problem could be solvable in polynomial time. A natural restriction is to limit the number N of valid power assignments allowed in a power schedule. In particular, for $N = 1$, we know from Section II that the problem is solvable in polynomial time. It turns out that already for $N = 2$ the problem becomes NP-hard. Another NP-hard variation, noted in Theorem 3 below, is to require integral time allocations to all power assignments in a power schedule.

We show NP-hardness of these problems by transformation from the following *node-disjoint Steiner tree packing problem*: given an undirected graph $G = (V, E)$, a set of terminal nodes $W \subseteq V$, and a positive integer N, decide whether there exists a set of N trees in G such that 1) every tree contains all the nodes in W and 2) every node in $V - W$ occurs in at most one tree. In [31, Lemma 2], we show that the node-disjoint Steiner tree packing problem is NP-complete if the number of terminal nodes is constant (four); whether this holds also for three terminal nodes is open. For two terminal nodes, the problem reduces to finding a maximum flow in a network and is thus solvable in polynomial time. If we allow a nonconstant number of terminal nodes, then packing even two node-disjoint trees is hard, as the following lemma shows.

Lemma 2: The node-disjoint Steiner tree packing problem is NP-complete. Moreover, the problem remains NP-complete even if $N = 2$.

As a consequence of this lemma we obtain the following.

Theorem 3: The multicast lifetime maximization problem remains NP-hard in both of the following special cases.

Case 1) The number of different power assignments in a power schedule is fixed to any constant greater than one.

Case 2) It is required that the time allocated to each power assignment in a power schedule is integral.

For proofs, we again refer to the Appendix. Results similar to those of Theorem 3 have been announced recently also by Krumke *et al.* [32]. Hardness and approximation of packing Steiner trees is investigated further in [38].

IV. Algorithms

We present two randomized algorithms for multicast lifetime maximization with dynamic power assignments, and a method for bounding the lifetime from above. We also modify the algorithms to apply to broadcast and group connectivity lifetime maximization.

A. Algorithm RNDGREEDY

The first algorithm is based on the polynomial-time static power assignment algorithm mentioned in Section II. The algorithm for the static case gives a solution where all nodes will run out of energy simultaneously, and where many nodes may be assigned transmission powers higher than actually necessary for multicast connectivity. Algorithm RNDGREEDY modifies this by reducing the powers. With the remaining energy, another power assignment may be generated, and so on. This can be continued for as long as the source node has any energy left; eventually, there remains the trivial multicast connection where the source 1 transmits directly to all sinks j (assuming $d_{1j} < \infty$).

Since the algorithm is a randomized one, it can be iterated several times, and the best solution chosen; in our experiments in Section V, we take the best solution obtained in 100 iterations of the algorithm.

```
Algorithm RNDGREEDY
1. While energy of source > 0, do:
    1.1. Choose an optimal power assign-
         ment with the static algorithm.
    1.2. For all nodes $i$ in random order:
         Reduce the power assigned to
         node $i$ to be as low as possible
         without breaking the multicast
         connectivity. (Binary search
         over all $d_{ij}$ that lie between 0
         and the power assigned to node $i$
         in Step 1.1; for each value,
         check connectivity with depth-
         first search.)
    1.3. Run the network with this power
         assignment until some node is
         depleted of energy. Update the
         energy supplies of all nodes
         according to the consumption.
```

B. Algorithm LPSCHEDULE

The second algorithm is based on the LP formulation described in Section III. It generates a collection of valid power assignments as columns in a matrix P. Time is then allocated to these assignments by solving the linear program (1).

```
Algorithm LPSCHEDULE
1. Initialize: $P \leftarrow [\ ]$, $e_{\mathrm{red}} \leftarrow e$ (the true
   energy constraints of each node).
2. Repeat for a number of iterations:
    2.1. Sampling: Generate a set of power
         assignments with RNDGREEDY, using
         reduced energy constraints $e_{\mathrm{red}}$.
         Append all of them as new columns
         to $P$.
    2.2. Scheduling: $t \leftarrow$ optimal times for
         $P$ with full energy $e$ (solve as
         LP).
    2.3. Consumption: $w \leftarrow Pt$.
    2.4. Energy for next iteration:
         $r \leftarrow$ uniform random number in
         [0,1], $e_{\mathrm{red}} \leftarrow e - r \cdot w$.
3. Return the power schedule $(P, t)$.
```

The difficulty lies in generating the collection of valid power assignments. As pointed out in [9], simply generating individually good power assignments may not result in a good collection. Intuitively, a good collection would contain assignments that are as "energy-disjoint" as possible, i.e., that exploit the energy supplies of different subsets of all nodes.

Algorithm LPSCHEDULE generates P iteratively: new power assignments are obtained using RNDGREEDY and accumulated. Power assignments are never discarded from the collection; we rely on the scheduling step to allocate zero time for inferior assignments. As long as P remains reasonable in size, scheduling is computationally cheap, compared with the cost of generating new valid power assignments.

In order to encourage the new power assignments to exploit other nodes than those heavily used by the current P, the algorithm RNDGREEDY is given a reduced energy constraint vector. In fact, the randomization of r in Step 2.4 of algorithm LPSCHEDULE is not critical to the operation of the algorithm.

C. Upper Bounds on Lifetime

Finally, we attack the problem from the opposite direction and formulate upper bounds for the maximum dynamic multicast lifetime achievable in a given network. Such an upper bound can be used, e.g., for assessing the quality of the solutions found by the algorithms.

Consider an arbitrary cut $C \subset V$ that contains the source, but not all the sinks. Replace multicast connectivity with a "path-out" property that only requires at least one path to exist from the source to any node outside C. Since this property is weaker than multicasting, its best achievable lifetime gives an upper bound on multicast lifetime.

On the other hand, the path-out property only depends on the power levels assigned to nodes in C. Thus, for small C, this weaker lifetime problem can be solved exactly by enumerating all such power assignments (inside C) that fulfill the path-out property and optimizing the time allocation among them using the linear program.

Experiments suggest that quite small cuts (e.g., the source and its six nearest neighbors) can give quite tight bounds on the dynamic multicast lifetime. Indeed, for multicast trees in a large network, the bottleneck condition is to escape a small neighborhood of the source, constrained by its limited energy supply. After that, the large number of nodes in the rest of the network provides an abundance of alternative routes to the sinks.

D. Broadcasting and Group Connectivity

As broadcasting is a special case of multicasting, all algorithms given, including the method for obtaining an upper bound, can be used without modification for the problem of broadcast lifetime maximization.

For group connectivity, as already noted, the polynomial-time static algorithm for multicasting applies; only the property check needs to be changed. In algorithm RNDGREEDY, the loop condition is changed to "While *all* nodes in W have energy > 0." Algorithm LPSCHEDULE needs no modifications. The multicast upper bound algorithm can be run once for each $u \in W$ as a source and $W \setminus \{u\}$ as sinks; the minimum of these upper bounds gives an upper bound for the group connectivity lifetime.

For other topological properties, such as k-connectivity, the LP formulation of lifetime maximization is still valid. If the

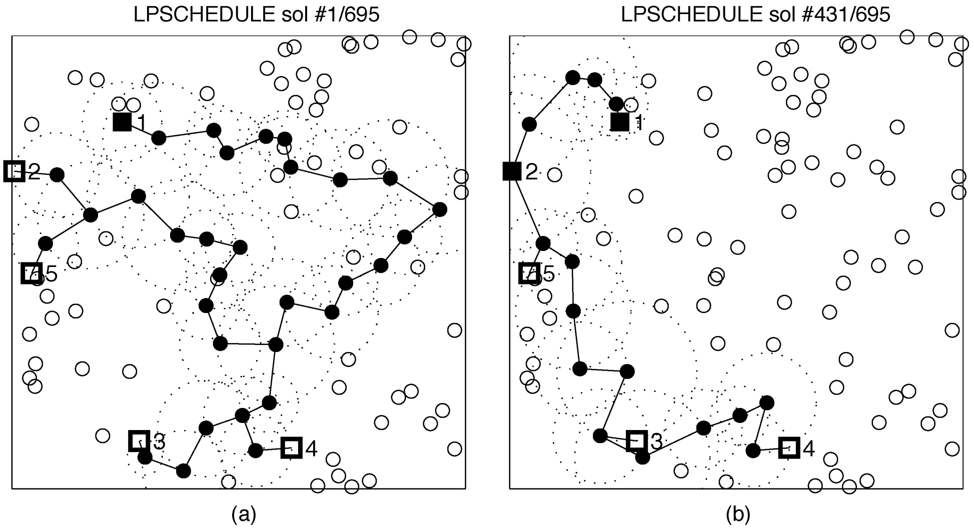

Fig. 1. Multicast trees in two valid power assignments in a random network. Source 1 and sink nodes 2, 3, 4, and 5 shown as squares. Filled nodes are transmitting at ranges indicated by the dotted circles; hollow nodes idle. (a) Optimal static assignment; it could be used for 59.7 units of time, but was allocated only 17.9 in the dynamic schedule. (b) The assignment that was allocated the most time in the dynamic schedule, 26.5.

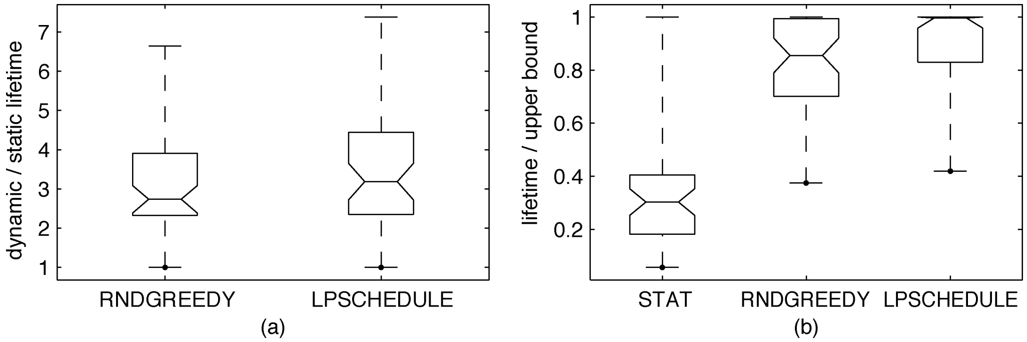

Fig. 2. Distribution of multicast lifetimes for 50 random networks of 100 nodes and 4 sinks. Notched boxes indicate median and quartiles; whiskers indicate minimum and maximum values. (a) Lifetime relative to the static solution. (b) Lifetime relative to the upper bound.

property is monotone and can be checked in polynomial time, both the static algorithm and RNDGREEDY can be used.

V. Experiments

The algorithms were implemented in MATLAB 6.5, using the Optimization Toolbox 2.2 for linear program solving. Experiments were run on a workstation with a 1333-MHz AMD Athlon processor.

In the experiments, 100 nodes (including one source and four sinks) were placed uniformly at random in the unit square. Power thresholds were computed as $d_{ij} = r_{ij}^2$, and all nodes were initially given one unit of energy. An upper bound for the lifetime was computed using a cut consisting of the source and its six nearest neighbors.

An example network, with a maximum static multicast lifetime of 59.7 units, is shown in Fig. 1. Both dynamic algorithms were run for 100 iterations on this network. For algorithm RNDGREEDY, we take the best result obtained in these iterations. Algorithm LPSCHEDULE generated a total of 695 different power assignments, of which 92 were allocated nonzero time. Two of them are illustrated in Fig. 1. The resulting power schedule has a lifetime of 236.8 units, a factor of 3.96 better than the static lifetime, and achieves 92.6% of the upper bound.

To gain some idea of the performance of the algorithms in general, we next generated 50 random networks. For each network, we ran the static algorithm and the two dynamic ones for 50 iterations each. The results are shown in Fig. 2. It can be seen that the dynamic power schedules have, in general, lifetimes about three times as long as the static ones. Also, algorithm LPSCHEDULE often finds near-optimal solutions; in fact, in more than half of the networks the resulting lifetime was within 1% of the upper bound.

The results show that dynamic power schedules can be clearly superior to static ones and that our algorithm LPSCHEDULE can find very good power schedules.

We also performed a similar experiment for broadcast and strong connectivity lifetime maximization. The results are shown in Figs. 3 and 4. For broadcasting, the ratios to the upper bounds are inferior to those of multicasting, although some of this might be caused by the upper bounds being too loose. For group connectivity, the ratios to the upper bounds are better than for multicasting. Because there is a "bottleneck region" around each terminal, the upper bounds are lower than for multicasting.

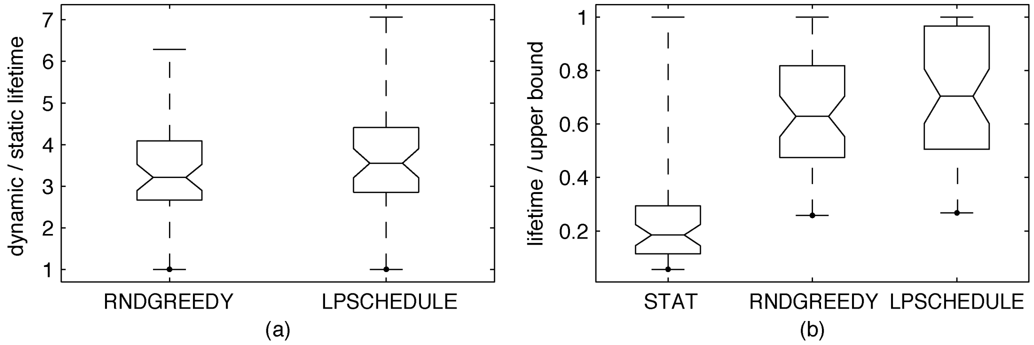

Fig. 3. Distribution of broadcast lifetimes for 50 random networks of 100 nodes. (a) Lifetime relative to the static solution. (b) Lifetime relative to the upper bound.

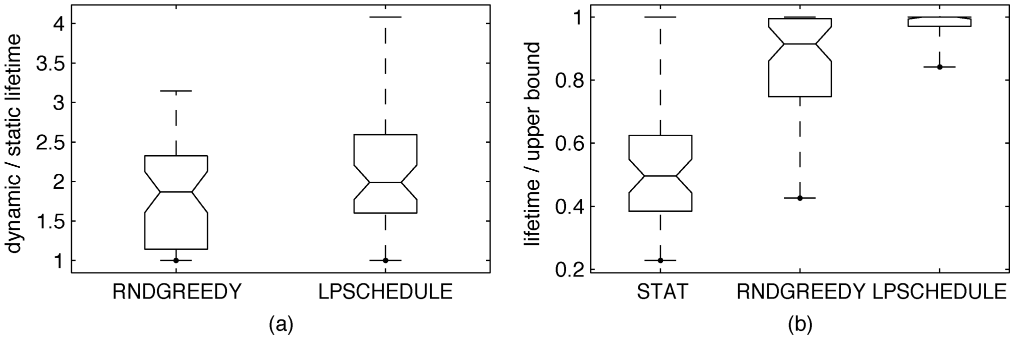

Fig. 4. Distribution of group connectivity lifetimes for 50 random networks of 100 nodes and 5 terminals. (a) Lifetime relative to the static solution. (b) Lifetime relative to the upper bound.

VI. Conclusion

A large amount of recent work has been directed toward energy minimization in wireless ad hoc networks, with the underlying goal of maximizing the lifetime. Our approach has been to optimize the lifetime directly. In addition, most previous work has been directed toward static power assignments; we have worked with more general dynamic ones.

We have proved that finding optimal power schedules for multicasting is NP-hard in the dynamic case and thus not likely to be exactly solvable by a polynomial-time algorithm. This inherent complexity of the problem notwithstanding, the two heuristic approximation algorithms we have presented for determining dynamic power schedules are able to consistently find schedules achieving multicast lifetimes 3–4 times as long as the optimal static power assignments. The approximation techniques are easily adaptable to related topology control problems, such as broadcast lifetime maximization and group connectivity lifetime maximization, in which cases similar performance improvements seem to be achieved.

We have assumed that the nodes are immobile, as in, e.g., sensor networks. In addition, our power assignment algorithms require some degree of centralized control of the network. Either all the nodes need to be aware of the network's complete structure and initial energy state, or they need to communicate with some central coordinating node. The problems of node mobility and distributed approximate optimization of the power schedules remain to be studied.

Appendix
Proofs

A. *Proof of Theorem 1*

We follow [34] in notation and terminology unless explicitly indicated otherwise.

Let O be an oracle for the lifetime maximization problem that behaves as follows. Given a power threshold graph $G = (V, A, d)$, the energy constraints $e = (e_1, \ldots, e_n) \geq 0$, a set of sink nodes $U \subseteq V$, and a rational accuracy parameter $\epsilon > 0$ as input, the oracle O outputs a rational number t such that $|t - t^*| \leq \epsilon$, where t^* is the maximum objective function value for the primal program (1). We assume that the encoding length $\langle t \rangle$ of t is bounded by a polynomial of $n + \sum_i \langle e_i \rangle + \sum_{i,j} \langle d_{ij} \rangle + \langle \epsilon \rangle$.

We describe an oracle algorithm $\mathcal{A}$ that solves the multicast total power minimization problem using O as an oracle. More precisely, given a power threshold graph $G = (V, A, d)$, a set of sink nodes $U \subseteq V$, and an accuracy parameter $\epsilon > 0$ as input, the algorithm $\mathcal{A}$ computes a rational number p such that $|p - p^*| \leq \epsilon$, where p^* is the minimum total power used by a valid power assignment in G. The running time of $\mathcal{A}$ and the number of oracle calls made are bounded by a polynomial in $n + \sum_{i,j} \langle d_{ij} \rangle + \langle \epsilon \rangle$.

The algorithm $\mathcal{A}$ enables us to decide in oracle polynomial time between YES and NO instances of set cover with integral weights (minimum cover [39, p. 222]). First, transform an instance of set cover into an instance of minimum broadcast cover [14] using the transformation in [14, Th. 1]. Then, input the resulting power threshold graph G to $\mathcal{A}$ so that all other nodes except the source are sink nodes. It is easy to see that all the power thresholds in G are integral. Thus, any accuracy $\epsilon < 1/2$ in Algorithm $\mathcal{A}$ suffices to decide between YES and NO instances of set cover. From this it follows that the multicast lifetime maximization problem is NP-hard under polynomial time Turing reductions [39, Ch. 5], and remains so in the special case when all the other nodes except the source are sinks. This proves Theorem 1.

It remains to describe the algorithm $\mathcal{A}$ in detail. Before proceeding with the description, we require some definitions and results. A polyhedron $P \subseteq \mathbb{R}^n$ is of *blocking type* [35, p. 114] if it is up-monotone and $P \subseteq \mathbb{R}^n_+$. If $c_1, \dots, c_M \in \mathbb{R}^n_+$ are the vertices of P, then

$$P = \operatorname{conv}(\{c_1, \dots, c_M\}) + \mathbb{R}^n_+ =: \{c_1, \dots, c_M\}^\uparrow.$$

The *blocker* of a polyhedron $P \subseteq \mathbb{R}^n_+$ is the polyhedron $\operatorname{bl}(P) = \{z \in \mathbb{R}^n_+ : z^T x \geq 1 \ \forall x \in P\}$.

Theorem 4: [35, Th. 9.2] Let $P \subseteq \mathbb{R}^n$ be a polyhedron of blocking type. Then:

1) $\operatorname{bl}(P)$ is a polyhedron of blocking type;
2) $\operatorname{bl}(\operatorname{bl}(P)) = P$;
3) if $P = \{c_1, \dots, c_M\}^\uparrow$, then $\operatorname{bl}(P) = \{z \in \mathbb{R}^n_+ : z^T c_i \geq 1 \ \forall i \in \{1, \dots, M\}\}$, and conversely.

We now proceed with the description of algorithm $\mathcal{A}$. Let a power threshold graph $G = (V, A, d)$ be given as input together with a set $U \subseteq V$ of sink nodes. To avoid degenerate cases, we may without loss of generality assume $d_{ij} > 0$ for all i, j. Denote by $p_k = (p_{1k}, \dots, p_{nk})$ the kth power assignment in G that satisfies the multicast connectivity property for the set of sinks U, $k \in \{1, \dots, N\}$.

We first require some bounds. Let Q be the polyhedron defined by the constraints of the dual program (2). Clearly, Q is of blocking type. The vertices $c_1, \dots, c_M \in \mathbb{Q}^n_+$ of Q are intersections of n hyperplanes of the form $\sum_i p_{ik} y_i = 1$ or $y_i = 0$. Consequently (see [35, Ch. 3]), the encoding length $\langle c_i \rangle$ for all $i \in \{1, \dots, M\}$ is bounded from above by a polynomial in $n + \sum_{i,j} \langle d_{ij} \rangle$. Thus, we can in time bounded by a polynomial in $n + \sum_{i,j} \langle d_{ij} \rangle$ find rational numbers ρ, σ, ρ', $\sigma' > 0$ with similarly polynomially bounded encoding length such that

$$\rho < \min_i \|c_i\|, \quad \max_i \|c_i\| < \sigma,$$
$$\rho' < \min_k \|p_k\|, \quad \max_k \|p_k\| < \sigma'.$$

We also observe that $N \leq n^n$ and $M \leq \binom{N+n}{n} \leq n^{n(n+1)}$. Thus, both $\langle N \rangle$ and $\langle M \rangle$ are bounded by a polynomial in n.

Algorithm $\mathcal{A}$ turns the optimization oracle O for the primal program (1) into a strong membership oracle for Q using a sequence of steps that are described next. Once the strong membership oracle is available, it is a simple matter to complete the proof.

In the following lemma, we observe that the optimization oracle O directly gives us a weak validity oracle for Q.

Lemma 5: There exists an oracle algorithm that solves the weak validity problem for Q using a single query to the optimization oracle O. The running time of the algorithm is bounded by a polynomial in $n + \sum_{i,j} \langle d_{ij} \rangle + \langle c \rangle + \langle \gamma \rangle + \langle \epsilon \rangle$.

Proof: Let $c \in \mathbb{Q}^n$, $\gamma \in \mathbb{Q}$, and a rational number $\epsilon > 0$ be given. If any coordinate of c is positive, then we can by up-monotonicity and nonemptiness of Q assert that there exists an $y \in S(Q, \epsilon)$ such that $c^T y \geq \gamma - \epsilon$. Otherwise, put $e = -c$ and query O with input $G = (V, A, d)$, e, U, ϵ. Let t be the answer returned by the oracle. If $t \geq -\gamma$, assert that $c^T y \leq \gamma + \epsilon$ for all $y \in S(Q, -\epsilon)$; call this case i). If $t < -\gamma$, assert that there exists an $y_0 \in S(Q, \epsilon)$ such that $c^T y_0 \geq \gamma - \epsilon$; call this case ii).

The definition of O and the LP duality imply that $e^T y \geq t - \epsilon$ holds for all $y \in Q$. Moreover, there exists an $y_0 \in Q$ such that $e^T y_0 \leq t + \epsilon$. In case i), we have $t \geq -\gamma$. Thus, $-c^T y \geq -\gamma - \epsilon$ holds for all $y \in Q$, which shows that the assertion made in case i) is correct. In case ii), we have $t < -\gamma$. Thus, $-c^T y_0 \leq -\gamma + \epsilon$, which shows that the assertion made in case ii) is correct. ■

The weak validity oracle for Q constructed in Lemma 5 gives us a weak membership oracle for the blocker $\operatorname{bl}(Q)$; take $P = Q$ and $r \leq \rho/M$ in Lemma 6 to obtain such an oracle.

Lemma 6: Let $P \subseteq \mathbb{R}^n_+$ be a nonempty up-monotone polyhedron given by a weak validity oracle. Suppose there exists a rational number $r > 0$ such that $S(0, r) \cap P = \emptyset$. Then, there exists an oracle algorithm that solves the weak membership problem for $\operatorname{bl}(P)$ with one query to the weak validity oracle for P. The running time of the algorithm is bounded by a polynomial in $n + \langle y \rangle + \langle \delta \rangle + \langle r \rangle$.

Proof: Let $y \in \mathbb{Q}^n_+$ and a rational number $\delta > 0$ be given. Select any $\epsilon > 0$ so that $\epsilon < \delta r / (2\sqrt{n}(1 + \|y\|_1))$ and $2\epsilon < r$. Query the weak validity oracle with $c = -y$, $\gamma = -1$, and ϵ. If the oracle asserts that $y^T x \geq 1 - \epsilon$ for all $x \in S(P, -\epsilon)$, then assert that $y \in S(\operatorname{bl}(P), \delta)$; call this case i). Otherwise (i.e., when the oracle asserts that $y^T x \leq 1 + \epsilon$ for some $x \in S(P, \epsilon)$), assert that $y \notin S(\operatorname{bl}(P), -\delta)$; call this case ii).

We first prove correctness of the assertion made in case i). Denote by $\mathbb{1}$ the vector with all components equal to one. Since P is up-monotone, we clearly have $x + \epsilon \mathbb{1} \in S(P, -\epsilon)$ for all $x \in P$. Thus, for all $x \in P$, $y^T x \geq 1 - \epsilon(1 + \|y\|_1)$. Put $y_0 = y - (\delta/\sqrt{n})\mathbb{1}$. Now, for all $x \in P$

$$\begin{aligned} y_0^T x &= \left(y - \frac{\delta}{\sqrt{n}} \mathbb{1}\right)^T x \\ &\geq 1 - \epsilon(1 + \|y\|_1) + \frac{\delta}{\sqrt{n}} \|x\|_1 \\ &\geq 1 - \epsilon(1 + \|y\|_1) + \frac{\delta}{\sqrt{n}} r \\ &> 1 \end{aligned}$$

by our choice of ϵ and $\|x\|_1 \geq \|x\|_2 \geq r$. Thus, $y_0 \in \operatorname{bl}(P)$, and since $\|y - y_0\| = \delta$, $y \in S(\operatorname{bl}(P), \delta)$.

It remains to show that the assertion made in case ii) is correct. Let $x \in S(P, \epsilon)$ such that $y^T x \leq 1 + \epsilon$. Put $x_0 = x + \epsilon \mathbb{1}$ and $y_0 = y - (\delta/(2\sqrt{n}))\mathbb{1}$. Because P is up-monotone, $x_0 \in P$. Now

$$\begin{aligned} y_0^T x_0 &= \left(y - \frac{\delta}{(2\sqrt{n})} \mathbb{1}\right)^T (x + \epsilon \mathbb{1}) \\ &\leq 1 + \epsilon(1 + \|y\|_1) - \frac{\delta}{(2\sqrt{n})} \|x\|_1 - \frac{\delta \epsilon n}{(2\sqrt{n})}. \end{aligned}$$

We, clearly, have $y_0^T x_0 < 1$ if $(\delta/(2\sqrt{n}))\|x\|_1 > \epsilon(1 + \|y\|_1)$. The last inequality holds because $x \in S(P, \epsilon)$ and our choice of ϵ imply $\|x\|_1 \geq \|x\|_2 \geq r - \epsilon > r/2$. Thus, $y_0 \notin \operatorname{bl}(P)$, which together with $\|y - y_0\| = \delta/2$ implies that $y \notin S(\operatorname{bl}(P), -\delta)$. ■

From the weak membership oracle for $\mathrm{bl}(Q)$ given by Lemma 6, we obtain via a theorem of Yudin and Nemirovskii [40] (as cited in [34, Sec. 4.3]), a weak validity oracle for $\mathrm{bl}(Q)$.

Theorem 7: There exists an oracle-polynomial time algorithm that solves the weak violation problem for every centered convex body $(K; n, R, r, a_0)$ given by a weak membership oracle [34, Th. 4.3.2].

Take $P = \mathrm{bl}(Q)$ and $\alpha \geq \sigma'$ in the following lemma to obtain a weak validity oracle from the weak membership oracle.

Lemma 8: Let $P \subseteq \mathbb{R}_+^n$ be a nonempty up-monotone polyhedron given by a weak membership oracle. Suppose there exists a rational number $\alpha > 0$ such that all the vertices of P occur in $S(0, \alpha)$. Then, there exists an oracle algorithm that solves the weak validity problem for P. The running time of the algorithm and the number of oracle calls made are bounded by a polynomial in $n + \langle c \rangle + \langle \gamma \rangle + \langle \epsilon \rangle + \langle \alpha \rangle$.

Proof: Let $c \in \mathbb{Q}_+^n$, $\gamma \in \mathbb{Q}$, and a rational number $\epsilon > 0$ be given. If any one coordinate of c is positive, then we can by up-monotonicity and nonemptiness assert that there exists an $x \in S(P, \epsilon)$ such that $c^T x \geq \gamma - \epsilon$. Similarly, if $c \leq 0$ and $\gamma \leq -\alpha \|c\|$, then we know that there exists a vertex $x \in P$ with $c^T x \geq \gamma \geq \gamma - \epsilon$ and can assert accordingly. Thus, the nontrivial case is when $c \leq 0$ and $\gamma > -\alpha \|c\|$.

Select rational $r, R > 0$ such that $r < \sqrt{n}\alpha$ and $R > 3\sqrt{n}\alpha$. Let $P' = P \cap S(0, R)$. Using the weak membership oracle for P, we can build a weak membership oracle for P': given $y \in \mathbb{Q}^n$ and $\delta > 0$, we proceed as follows. If $\|y\| > R$, we assert that $y \notin S(P', -\delta)$; otherwise, we run the weak membership oracle for P with input y, δ and assert accordingly.

In the nontrivial case, we run the algorithm of Theorem 7 for the centered convex body $(P'; n, R, r, 2\alpha\mathbb{1})$ given by the weak membership oracle for P' and assert accordingly. ■

Next, we apply Lemma 6 with $P = \mathrm{bl}(Q) = \{p_1, \ldots, p_N\}^\uparrow$ and $r \leq \rho'/N$ to obtain a weak membership oracle for $\mathrm{bl}(\mathrm{bl}(Q)) = Q$. Applying the technique in [34, Th. 6.3.2(b)] with $a_0 = \sigma\mathbb{1}$, we can transform the weak membership oracle for Q into a strong membership oracle whose running time together with the number of queries made to O is bounded by a polynomial in $n + \sum_{i,j} \langle d_{ij} \rangle$.

It remains to complete the description of algorithm $\mathcal{A}$. Let λ^* be the minimum $\lambda > 0$ such that $\lambda \mathbb{1} \in Q$. In particular, it is easy to see that $1/\lambda^*$ is the minimum total power required for multicasting in G. Let $\epsilon > 0$ be the accuracy parameter given to algorithm $\mathcal{A}$ as input. Algorithm $\mathcal{A}$ first computes the bounds ρ, σ, ρ', σ' based on the power thresholds d_{ij}, and then selects a rational $\tau > 0$ so that

$$\tau \leq \frac{\epsilon \left(\frac{1}{n\sigma'} \right)^2}{1 + \frac{\epsilon}{\rho'}}.$$

Then, using $\ell = \lceil -\log_2 \rho' \tau \rceil \leq 1 + \langle \rho' \rangle + \langle \tau \rangle$ steps of binary search in the interval $1/(n\sigma') \leq \lambda \leq 1/\rho'$, algorithm $\mathcal{A}$ finds a $\lambda > 0$ such that $|\lambda - \lambda^*| \leq \tau$. It then outputs $p = 1/\lambda$ and halts. By the choice of τ

$$\frac{1}{\lambda} - \epsilon \leq \frac{1}{\lambda + \tau} \leq \frac{1}{\lambda^*} \leq \frac{1}{\lambda - \tau} \leq \frac{1}{\lambda} + \epsilon.$$

Thus, $|1/\lambda - 1/\lambda^*| \leq \epsilon$. It is straightforward to check that the overall running time of algorithm $\mathcal{A}$ is bounded by a polynomial in $n + \sum_{i,j} \langle d_{ij} \rangle + \langle \epsilon \rangle$. ■

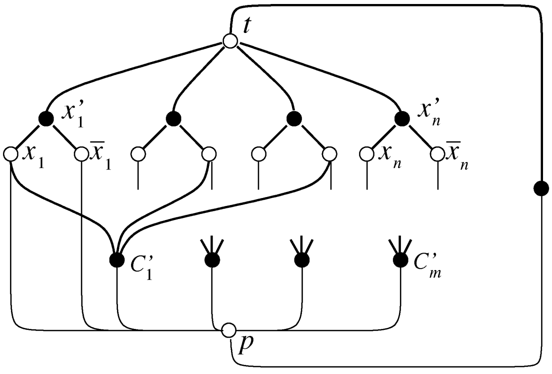

Fig. 5. Graph G in the proof of Lemma 2.

B. Proof of Lemma 2

The problem is clearly in NP. To show completeness, we use transformation from 3SAT [39, p. 48]. Let $C_1, \ldots, C_m$ be the m clauses over n variables $x_1, \ldots, x_n$. The graph G consists of $2n + 2$ nonterminal nodes and $m + n + 1$ terminal nodes. One terminal node c' (*core*) is connected by an edge to nonterminal nodes t (*truth assignment*) and p (*pad*). Associated with each clause C_j there is one terminal node C'_j. Each variable x_i is represented by a *variable gadget* consisting of three nodes: one terminal node x'_i adjacent to two nonterminal nodes x_i and $\bar{x}_i$. The node p is adjacent to c' and all the nodes $x_i, \bar{x}_i, C'_j, 1 \leq i \leq n, 1 \leq j \leq m$. The node t is adjacent to c' and all the terminals x'_i, $1 \leq i \leq n$. Finally, each node C'_j is adjacent to p and to the three literals (i.e., nonterminal nodes of the form x_i or $\bar{x}_i$) that occur in the clause C_j. It is straightforward to check that G admits two node-disjoint Steiner trees connecting the terminals if and only if the 3SAT instance is satisfiable. ■

C. Proof of Theorem 3

Consider the graph G in the proof of Lemma 2 (Fig. 5). Let c' be the source and suppose the other terminal nodes are the sinks. Suppose that c' has two units of energy and that all the other nodes have one unit of energy. Each node may transmit to any adjacent node with power one; the threshold to other nodes is infinite. This defines a power threshold graph together with the energy constraints on the nodes.

Consider Case 1) in the theorem. It is easy to see that a power schedule with a lifetime greater than one must contain at least two different power assignments. This is because the time allocated to a power assignment cannot exceed one due to the energy constraints on the nodes. Furthermore, if a power schedule with lifetime greater than one contains exactly two different power assignments, then no other node except the source c' can transmit in both power assignments. Thus, the sets of nonterminals that transmit in each power assignment induce two node-disjoint Steiner trees connecting the terminals. Conversely, we can transform two node-disjoint Steiner trees into two power assignments, in which no node except the source transmits in both assignments.

Let $K \geq 2$ be an integer. If we now modify G so that c' has K units of energy and add $K-2$ unit-energy nonterminal nodes adjacent to all the terminals, then a similar argument shows that achieving a lifetime of over $K-1$ with a power schedule consisting of at most K different power assignments requires that each nonterminal transmits in at most one power assignment. This proves Case 1).

Case 2) follows using a similar construction: a lifetime of $K \geq 2$ time units can be achieved if and only if G contains K node-disjoint Steiner trees connecting the terminals. ■

REFERENCES

[1] M. Ilyas, *The Handbook of Ad Hoc Wireless Networks*. Boca Raton, FL: CRC Press, 2003.

[2] C. E. Perkins, *Ad Hoc Networking*. Reading, MA: Addison-Wesley, 2001.

[3] I. Stojmenović, *Handbook of Wireless Networks and Mobile Computing*. New York: Wiley, 2002.

[4] R. Rajaraman, "Topology control and routing in ad hoc networks: a survey," *ACM SIGACT News*, vol. 33, no. 2, pp. 60–73, Jun. 2002.

[5] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless ad-hoc networks," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw. (MOBICOM 2001)*, 2001, pp. 97–107.

[6] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. 4th Annu. Int. Conf. Mobile Comput. Netw.*, 1998, pp. 181–190.

[7] G. Calinescu, S. Kapoor, A. Olhevsky, and A. Zelikovsky, "Network lifetime and power assignment in ad-hoc wireless networks," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2003, vol. 2832, Proc. 11th Annu. Eur. Symp. Algorithms, pp. 114–126.

[8] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks," in *Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2002, pp. 123–134.

[9] R. J. Marks II, A. K. Das, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Maximizing lifetime in an energy constrained wireless sensor array using team optimization of cooperating systems," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2002, pp. 371–376.

[10] I. Kang and R. Poovendran, "Maximizing static network lifetime of wireless broadcast ad hoc networks," in *Proc. IEEE Int. Conf. Commun.*, pp. 2256–2261.

[11] B. Wang and S. K. S. Gupta, "Extending the lifetime of multicast trees in WANETs," *J. Inf. Sci. Eng.*, vol. 20, pp. 425–447, 2004.

[12] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2000, pp. 404–413.

[13] T. S. Rappaport, *Wireless Communications: Principles & Practice*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[14] M. Čagalj, J.-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," in *Proc. 8th Annu. Int. Conf. Mobile Comput. Netw.*, 2002, pp. 172–182.

[15] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos, "A logarithmic approximation algorithm for the minimum energy consumption broadcast subgraph problem," *Inf. Process. Lett.*, vol. 86, pp. 149–154, 2003.

[16] A. E. F. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca, "On the complexity of computing minimum energy consumption broadcast subgraphs," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2001, vol. 2010, Proc. 18th Symp. Theor. Aspects Comput. Sci., pp. 121–131.

[17] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: Integer programming formulations," in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2003, pp. 1001–1010.

[18] W. Liang, "Constructing minimum-energy broadcast trees in wireless ad hoc networks," in *Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2002, pp. 112–122.

[19] R. J. Marks, II, A. K. Das, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: Optimizing using the viability lemma," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. I, 2002, pp. 273–276.

[20] I. Papadimitriou and L. Georgiadis, "Energy-aware broadcasting in wireless networks," in *Proc. WiOPT'03: Modeling Opt. Mobile, Ad Hoc Wireless Netw.*, 2003, pp. 267–277.

[21] P.-J. Wan, G. Călinescu, X.-Y. Li, and O. Frieder, "Minimum-energy broadcasting in static ad hoc wireless networks," *Wireless Netw.*, vol. 8, pp. 607–617, 2002.

[22] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2000, pp. 585–594.

[23] ——, "Algorithms for energy-efficient multicasting in static ad hoc wireless networks," *Mobile Netw. Appl.*, vol. 6, pp. 251–263, 2001.

[24] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, "Power consumption in packet radio networks," *Theor. Comput. Sci.*, vol. 243, pp. 289–305, 2000.

[25] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, pp. 1333–1344, Aug. 1999.

[26] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks," in *Proc. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2000, pp. 22–31.

[27] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Resource management in energy-limited, bandwidth-limited, transceiver limited wireless networks for session-based multicasting," *Comput. Netw.*, vol. 39, pp. 113–131, 2002.

[28] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Berlin, Germany: Springer-Verlag, 1999.

[29] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner problems," *J. Algorithms*, vol. 33, pp. 73–91, 1999.

[30] E. Halperin and R. Krauthgamer, "Polylogarithmic inapproximability," in *Proc. 35th ACM Symp. Theory Comput.*, 2003, pp. 585–594.

[31] P. Floréen, P. Kaski, J. Kohonen, and P. Orponen, "Multicast time maximization in energy constrained wireless networks," in *Proc. 2003 Joint Workshop Found. Mobile Comput. (DIALM-POMC'03)*, 2003, pp. 50–58.

[32] S. O. Krumke, R. Liu, E. L. Lloyd, M. V. Marathe, R. Ramanathan, and S. S. Ravi, "Some Preliminary results on maximizing lifetimes of ad hoc networks," Comput. Sci. Dept., Univ. Albany-State Univ. New York, Tech. Rep. SUNYA-CS-03-04, Aug. 2003.

[33] M. Adler and C. Scheideler, "Efficient communication strategies for ad hoc wireless networks," *Theory Comput. Syst.*, vol. 33, pp. 337–391, 2000.

[34] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, 2nd ed. Berlin, Germany: Springer-Verlag, 1993.

[35] A. Schrijver, *Theory of Linear and Integer Programming*. New York: Wiley, 1986.

[36] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing Steiner trees," in *Proc. 14th ACM-SIAM Symp. Discrete Algorithms*, 2003, pp. 266–274.

[37] N. Garg and J. Könemann, "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," in *Proc. 39th Annu. Symp. Found. Comput. Sci.*, 1998, pp. 300–309.

[38] J. Cheriyan and M. R. Salavatipour, "Hardness and approximation results for packing Steiner trees problems," in *Proc. 12th Annu. Eur. Symp. Algorithms*, 2004, pp. 180–191.

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

[40] D. B. Yudin and A. S. Nemirovskii, "Informational complexity and effective methods for the solution of convex extremal problems" (in Russian), *Ékonomika i Matematicheskie Metody*, vol. 12, pp. 357–369, 1976.

Patrik Floréen received the Ph.D. degree in computer science from the University of Helsinki, Helsinki, Finland, in 1992.

He is currently a Senior Research Scientist at Helsinki Institute for Information Technology (HIIT), Helsinki, Finland. His past research includes theoretical work on neural networks and genetic algorithms, and current research interests focus on theoretical aspects of ad hoc networks and on theory and practice of context-aware computing.

Petteri Kaski is working towards the Ph.D. degree at the Laboratory for Theoretical Computer Science, Helsinki University of Technology (HIIT), Espoo, Finland.

His current research is focused on combinatorial algorithms and computational complexity.

Jukka Kohonen is working towards the Ph.D. degree in computer science at the University of Helsinki, Helsinki, Finland.

He is a Researcher at Helsinki Institute for Information Technology (HIIT), Helsinki, Finland. His current research is focused on theory for ad hoc networks.

Pekka Orponen (M'87) received the Ph.D. degree in computer science from the University of Helsinki, Helsinki, Finland, in 1986.

He is currently a Professor of Theoretical Computer Science at the Helsinki University of Technology, Espoo, Finland. His research interests are in the area of computational complexity and its application to the design of efficient algorithms, in particular, in the context of wireless communication networks.