The Power Balancing Problem in Energy Constrained Multi-hop Wireless Networks

Randeep Bhatia† Abhishek Kashyap‡ Li (Erran) Li†
†Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ, USA
Email: {randeep, erranlli}@research.bell-labs.com
‡ ECE Dept., University of Maryland, College Park, USA
Email: kashyap@eng.umd.edu

***Abstract*— Power efficient operation is very critical in energy constrained multi-hop wireless networks. One important technique is to intelligently assign transmission powers to nodes while maintaining network connectivity. Previous work has focused on assigning a single transmission power to each node. This often leads to "power imbalance", where some nodes use much more power than other nodes. This can reduce network lifetime. In this paper, we investigate the problem of two power assignments where nodes alternate the use of these assigned powers. We rigorously formulate the problem of two power assignment under the constraint that the network connectivity is maintained. The objective here is to minimize the maximum average power used by the nodes. We show that, in general, the problem is not just NP-hard but also hard to approximate. We then propose a distributed localized heuristic to compute the two power assignments. We perform extensive simulations to show that the algorithm can reduce the average power significantly when compared with algorithms that assign a single power. By assuming some properties on radio propagation, we also present a centralized algorithm with bounded worst case guarantees for the two power assignment problem.**

I. INTRODUCTION

In energy-constrained multi-hop wireless networks such as sensor networks, it is very important to minimize the power expended in transmission for prolonged network lifetime. However, a node cannot use arbitrarily small transmission power as the connectivity of the network depends on the power each node uses. Depending on whether traffic demand is known or not, prior work can be classified into two categories: the first [3] tries to optimize network lifetime directly based on various notions of network lifetime. The second [9], [6], [7] uses heuristic metrics such as minimizing the maximum transmission power or the sum of the transmission power over all nodes, in an attempt to reduce transmission power consumption. The latter approach is often referred to as topology control.

In this paper, we take the latter approach as in many practical settings, it is hard to obtain the traffic demand, and considerable overhead is incurred to obtain global topology information. The literature on topology control has focused on configuring one transmission power. With only local topology information, e.g. the visibility graph (the topology that a node sees using its maximum transmission power), a node independently assigns a transmission power to itself, attempting to optimize certain metrics such as the min-max power while ensuring the network is connected (which is the case as long as the union of visibility graph over all nodes is connected).

One important drawback of this approach is that some nodes always use a very large transmission power compared to other nodes since the transmission power assignment is fixed. In this paper, we explore the benefit of multiple transmission power assignments. We maintain network connectivity with each power assignment. We focus on two transmission power assignments. A node will alternate the use of these two transmission powers. Some radio technologies may incur a switching delay when changing transmission powers. The most efficient is when each node is equipped with two radios. Each source node can stamp IP packet header with a bit indicating the power assignment the packet requires. For route request control packets, the bit will indicate whether a route should be found on the network topology defined by the first power or second power assignment. The benefit of multiple power assignments is "power balancing". This is analogous to load balancing. We seek to minimize the maximum over the nodes of the sum of the two powers.

In this paper, we formally define the multiple power assignment problem. We prove that it is NP-hard even for the case of two powers if we do not assume any property of radio propagation. Without any property of radio propagation, we show that there does not exist any polynomial time algorithm that will not be worse than an optimal single power in all problem instances. With the assumption of power decays with the distance to the power of 2 to 4, we present a constant approximation algorithm for the problem. We then focus on designing a "localized" distributed algorithm for the two power assignment problem. As the "localized" algorithm can be arbitrarily worse than the optimal in the single power case, we focus on comparing with a "localized" MST based algorithm [7]. An important property of the LMST algorithm is that it is the best one can do given the visibility graph. We design a two power algorithm that will either improve on LMST or have the same performance as LMST. That is, it will not be worse than LMST for min-max power assignment. Our extensive simulation results demonstrate that the algorithm can effectively "power balance" among nodes.

The rest of the paper is organized as follows. In Section II, we present a rigorous formulation of the power balancing problem. In Section V, we show that the general problem is NP-hard. In Section III, we present the details of centralized algorithm and show that it has a constant approximation factor. In Section IV, we present a localized algorithm that will never be worse than LMST [7]. In Section V, we show that the general problem is NP-hard. In Section VI, we evaluate our algorithms using simulation. In Section VII, we present related

work. Finally, in Section VIII, we present our conclusions.

II. Problem Formulation

We are given a network of nodes V and edges E. The edges E may be directed or undirected. Since each undirected edge can be thought of as a pair of directed edges, we describe the problem for directed edges E. The extension to undirected edges is straightforward. Each edge $e = (u, v) \in E$ is labelled with a power $P(e)$, such that $P(e)$ is the minimum power needed at node u to send a packet to node v. Thus, if P_u is the power used by a node u then directed edge $e = (u, v)$ is active iff $P_u \geq P(e)$.

A set of node powers $P_v,\ v \in V$ is said to be feasible if the resulting network of active edges is strongly connected. Likewise, we define a k set of node powers $P_v^1, P_v^2, \ldots P_v^k,\ v \in V$ to be feasible if for the set of node powers $P_v^i, v \in V$, the resulting network of active edges is strongly connected, for every $1 \leq i \leq k$. In this paper we are interested in a feasible power assignment problem for $k = 2$. This problem is defined as follows.

The two power assignment problem: The objective of the two power assignment problem is to find a feasible 2 set of node powers $P_v^1, P_v^2, v \in V$ and fractions f_1 and f_2 adding up to 1 (i.e. $f_1 + f_2 = 1$) such that the maximum average node power $\max_v f_1 P_v^1 + f_2 P_v^2$ is minimized.

In a more realistic Euclidean setting the nodes V are points in the Euclidean plane. The edge path (power) loss is inversely proportional to a power α of the distance for some $2 \leq \alpha \leq 4$ [8]. Thus the power received at node v from node u transmitting at power P_0 on an edge $e = (u, v)$ of length d is proportional to $\frac{P_0}{d^\alpha}$. In this setting the label of edge e of length d is proportional to d^α. Thus $P(e) = Kd^\alpha$, for some constant K. Also, in realistic settings, each node v has a bounded maximum power which we denote by $P_{\max}(v)$. In this setting directed edge $e = (u, v) \in E$ iff $P_{\max}(u) \geq P(e)$.

For simplicity we assume that all $P_{\max}(v)$ are equal and the common maximum power is denoted by $P_{\max}$. Our results also extend to the more general case the details of which we leave for a full version of the paper. Note that in the Euclidean setting with the assumption of equal maximum node power, the edges E in the network (possible set of active edges) are undirected. Thus, in the rest of the paper, we will *mainly work with undirected networks* $G = (V, E)$.

III. Centralized Algorithm

A. Preliminaries and Notations

We assume that there are m network edges E ($|E| = m$), which are indexed from 1 to m. Thus, we denote the set of edges by $e_1, e_2, \ldots e_m$. For a given set of edge weights $w(e),\ e \in E$ we say that edge e_i is heavier than edge e_j if either $w(e_i) > w(e_j)$ or $i > j$ and $w(e_i) = w(e_j)$. Thus, we use a tie breaking rule to differentiate among equal weight edges. We denote by $N(v)$ the set of one hop neighbors of node v.

B. The Algorithm

The centralized algorithm is assumed to have a global view of the network and its topology. The algorithm has two phases. In the first phase the algorithm finds a spanning tree of the network as follows. For each vertex u, let $P_{\min}(u)$ be the minimum power needed to communicate with at least one neighbor. Thus, $P_u^{min} = \min\{P(e) | e = (u, v), v \in N(u)\}$. In this phase the weight of edge $e = (u, v)$ is then set as $w(e) = \max\{P_u^{min}, P_v^{min}\} + P(e)$. The algorithm then finds a Minimum Spanning Tree (MST) of the network using these edge weights with the tie breaking rule described earlier to differentiate among equal weight edges.

Given the MST, we denote a node u to be the "buddy" $B(v)$ of node v if the edge (v, u) is the lightest among all the edges incident on node v in the MST. Due to the tie breaking rule, $B(v)$ is unique for every node v. It can also be shown (proof omitted) that $w(e) = 2P(e)$, where $e = (v, u)$.

Next, the algorithm labels the vertices V with two labels 0 and 1 such that for any edge e in the MST its two end-nodes do not have the same label. Note that this is equivalent to a 2-vertex coloring of the MST and is done by selecting an arbitrary node r as the root of the MST. All nodes that are at even (hop) distance from r in the MST are assigned label 0, and all the other nodes are assigned label 1. It can be verified that this is indeed a 2-vertex coloring of the MST.

The second phase of the algorithm involves repeatedly running the following steps for different values of a parameter γ in the range $1/2 \leq \gamma \leq 1$. Each choice of γ results in a pair of strongly connected networks $T_1(\gamma)$ and $T_2(\gamma)$ over the node set V. In the end the algorithm outputs the best pair of strongly connected networks $T_1(\gamma)$ and $T_2(\gamma)$ among the different choices for γ (i.e. for which the maximum average node power is minimized).

Let W_{MST} be the largest edge weight in the MST. Let E_1 and E_2 be a partition of the edges of the MST with $e \in E_1$ iff the edge power $P(e)$ is at most γW_{MST}. Intuitively, E_2 is the set of long edges of the MST and E_1 is the set of short edges of the MST. Note that the set of short edges (E_1) can be thought of as forming a set C of connected components such that the end-nodes of any long edge e ($e \in E_2$) lie in different connected components of the set C. It can be shown (proof omitted), based on the way the sets E_1 and E_2 are defined, that if u is a buddy of v ($B(v) = u$) then the edge (v, u) must be in E_1 and hence the two nodes v and u must be in the same connected component of C.

For every long edge $e = (u, v) \in E_2$, the algorithm does the following in the second phase. Let u and v be in (distinct) connected components c_1 and c_2 of C respectively. Let $B(u) = u_1$ and let $B(v) = v_1$. Note that u_1 and v_1 must be in c_1 and c_2 respectively. If there do not exist both the edges $e_1' = (u, v_1) \in E$ and $e_2' = (v, u_1) \in E$ then edge e is moved from the set E_2 to the set E_1. Note that this also results in the components c_1 and c_2 getting merged into one connected component in set C. We denote these new sets with this transformation also as E_2 and E_1. We denote by

$E_{MST} = E_1 \cup E_2$ the set of MST edges.

Let $P_{MST}(u)$ denote the power for node u in the MST (the maximum power on any edge incident on u in the MST). The algorithm assigns an ordered pair of powers (P_v^1, P_v^2) to each node v as follows. Let S_v be the set of nodes whose buddy is node v. Thus $S_v = \{u : B(u) = v\}$. Consider the case when either v has an edge $e = (v, u) \in E_1$ incident on it that belonged to the original set E_2 (got moved to the set E_1 by the algorithm), or the case when all the nodes in the set $S_v \cup \{v\}$ have either an edge $e \in E_1$ incident on them that belonged to the original set E_2 or have no edge from the set E_2 incident on them. In both these cases the two powers P_v^1 and P_v^2 are both set to $P_{MST}(v)$. Let W be the nodes whose power is set this way. Now, consider a node $v \notin W$. Let Y_v denote the set of nodes in $S_v \cup \{v\}$ that are not in W. We define a high power $P_h(v)$ for node v as follows. $P_h(v)$ is at least $P_{MST}(v)$. $P_h(v)$ is also at least $P(e)$, where edge $e = (v, w)$ such that for $u \in Y_v$ and $u \neq v$ there is an MST edge (u, w). $P_h(v)$ is chosen as the lowest power to satisfy these constraints. Likewise we define a low power $P_l(v)$ for node v as follows. $P_l(v)$ is at most $P(e)$, where edge $e = (v, B(v))$ if $v \in Y_v$. $P_l(v)$ is also at most $P(e)$, where edge $e = (u, v)$ for $u \in Y_v, u \neq v$. $P_l(v)$ is chosen as the highest power to satisfy these constraints. Depending on whether node v is labelled 0 or 1, the power P_v^1 or P_v^2 is set to $P_h(v)$ and the power P_v^2 or P_v^1 is set to $P_l(v)$ respectively.

Next, for the choice of γ used above, the algorithm outputs two networks $T_1(\gamma)$ and $T_2(\gamma)$ as follows. $T_1(\gamma)$ is the (directed) network obtained by setting the power of every node v to P_v^1. $T_2(\gamma)$ is the (directed) network obtained by setting the power of every node v to P_v^2. We can show (proof omitted) that both these networks are strongly connected. Intuitively, in $T_1(\gamma)$, a node v with label 0 may only use its small power. However then for node v there must exist a MST neighbor u (u is in the set Y for node v) that node v can reach with its small power and where node u with its high power is able to reach all the MST neighbors of node v. Since u and v are MST neighbors u has label 1. Thus, u must use its high power in $T_1(\gamma)$. Therefore, v is able to reach its MST neighbors by a 2 hop path in $T_1(\gamma)$. Thus, the MST connectivity is maintained in $T_1(\gamma)$, and also by the same reasoning in $T_2(\gamma)$.

Note that the algorithm only needs to try out at most $m = |E|$ possible values for γ to find the best pair of networks $T_1(\gamma)$ and $T_2(\gamma)$. This is because two values of γ would result in the same pair of networks, $T_1(\gamma)$ and $T_2(\gamma)$, if they lead to the same partition E_1 and E_2 of the edges of the MST (based on the weight threshold γW_{MST}).

C. Analysis

Now we establish that the algorithm has a non-trivial bounded worst case performance. We say that the network is well "powered" if the maximum node power $P_{\max}$ is much higher than the minimum power $P_{\min}$ needed at each node so that the resulting topology is connected. In particular we assume that $P_{\max}$ is at least $2^{(\alpha+1)}P_{\min}$, where α is the path loss exponent. The implication of a network being well powered is that for every long edge $e = (u, v) \in E_2$ in the first phase MST of the centralized algorithm the edges $(u, B(v)), (v, B(v))$ are both in E. Hence no edge is moved from the set E_2 to the set E_1.

Theorem 3.1: For a well powered network, compared to an optimal two power solution, the solution output by the algorithm is guaranteed to use no more than 1.76 times the node power for $\alpha = 2$ and no more than 1.966 times the node power for $\alpha = 3$.

Proof: Since the network G is well powered, all the edges that eventually end up being in E_1 are of the size γW_{MST}. It can be shown (proof omitted) that in the optimal solution the average node power P_{OPT} must be at least $W_{MST}/2$. Thus $P_{OPT} \geq W_{MST}/2$. We consider the solution where both the networks $T_1(\gamma)$ and $T_2(\gamma)$ are used with frequency one half each ($f_1 = f_2 = 1/2$).

By construction, the average power of a node v, for which the algorithm sets both its powers to $P_{MST}(v)$, is at most γW_{MST}, since all MST edges incident on node v must be from the set E_1. It can be shown (proof omitted) that for any other node v there are two possibilities in the solution output by the algorithm. Either its average power is at most γW_{MST}, or there are nodes u and w such that $e = (v, w) \in E$ and $B(u) = v$ and both the edges $e'' = (u, w)$ and $e' = (v, u)$ are in the MST. In addition an edge e^+ from the set E_2 is incident on node u in the MST. In this case the average power of node v is at most $(P(e) + P(e'))/2$. We now consider this case.

Note that since $B(u) = v$ then for any MST edge e^u incident on node u we have $P(e') + P(e^u) \leq w(e^u) \leq W_{MST}$. Recall that $w(e^u)$ is the weight of edge e^u in the MST. In particular this applies to $e^u = e''$ and to $e^u = e^+$. So it follows that $P(e') + P(e'') \leq W_{MST}$. Also since $e^+ \in E_2$ it follows that $P(e') \leq (1 - \gamma)W_{MST}$. Let $l(e)$ denote the length of edge e. Then by triangle inequality we have $l(e = (v, w)) \leq l(e' = (v, u)) + l(e'' = (u, w))$. Since $P(e)$ is proportional to $l(e)^\alpha$, we have

$$P(e) \leq (P(e')^{\frac{1}{\alpha}} + P(e'')^{\frac{1}{\alpha}})^\alpha.$$

Thus, the average power of node v is at most

$$\frac{(P(e')^{\frac{1}{\alpha}} + P(e'')^{\frac{1}{\alpha}})^\alpha + P(e')}{2}.$$

Since $P(e') \leq (1 - \gamma)W_{MST}$ and $P(e') + P(e'') \leq W_{MST}$, it follows (proof omitted) that the average power of node v is maximized when $P(e') = (1 - \gamma)W_{MST}$ and $P(e'') = \gamma W_{MST}$ for $0.5 \leq \gamma \leq 1$. Thus in this case the average power of node v is at most

$$\frac{W_{MST}}{2}((\gamma^{\frac{1}{\alpha}} + (1 - \gamma)^{\frac{1}{\alpha}})^\alpha + (1 - \gamma)).$$

¿From the above it follows (proof omitted) that for $\alpha = 2$ and 3 the best choices for γ are 0.88 and 0.983 resulting in solutions with approximation factors 1.76 and 1.966 respectively. ■

IV. Localized Construction of Two Power Solution

We now present a localized algorithm for construction of the two power solution. We prove the algorithm yields strongly connected topologies. We also prove the algorithm is a 2-approximation of any optimal algorithm that uses only localized communication.

The algorithm proceeds in two phases, creating a connected topology (may be a general graph rather than a tree) based on one set of node powers in each phase. In the first phase, each node broadcasts a node discovery message with its location information periodically. The nodes use their maximum transmission power (P_{max}) to transmit these messages. Since we assume all nodes have the same maximum transmission power, a node can transmit to all nodes it received a node discovery message from recently. This way a node is able to identify all its neighbors and their locations (set of neighbors of node u is denoted by $N(u)$). Based on the neighborhood information (neighbor locations), each node u constructs a *visibility graph*, $G_u = (V_u, E_u)$, $V_u = \{u\} \cup N(u)$ and $E_u = \{e = (u_1, u_2) | u_1, u_2 \in V_u, e \in E\}$. Note that node u is able to compute the set E_u since it knows the location of each of its neighbors in V_u. Thus, it can determine if there is an edge (in E) between any two of its neighbors. Each node u calculates the power it needs to communicate with its nearest neighbor, $P_u^{min} = \min\{P(e) | e = (u, v), v \in N(u)\}$. Then, each node u broadcasts P_u^{min}. Each node u then assigns weight $w(e)$ to each edge $e = (u_1, u_2) \in E_u$ as $\max\{P_{u_1}^{min}, P_{u_2}^{min}\} + P(e)$. For a given set of edge weights $w(e)$ $e \in E$ the algorithm considers edge $e_i = (u_i, v_i)$ to be heavier than edge $e_j = (u_j, v_j)$ if either $w(e_i) > w(e_j)$ or $u_i > u_j$ and $w(e_i) = w(e_j)$ or $v_i > v_j$ and $w(e_i) = w(e_j)$, $u_i = u_j$. The edge weights assigned to each edge are the same in all visibility graphs it is a part of. The edge weight is less than twice the minimum average power any two power solution will use at the end-points of the edge, if it picks the edge in the topology corresponding to either power assignment. We state this in the following lemma:

Lemma 4.1: Any two power solution with topologies $G_1^o = (V, E_1^o), G_2^o = (V, E_2^o)$ uses maximum average node power of at least $\max\{0.5w(e) = 0.5\{\max\{P_u^{min}, P_v^{min}\} + P(e))|e = (u, v) \in E_1^o \cup E_2^o\}$.

Proof: W.l.o.g. let the fractions f_1, f_2 associated with these two powers (topologies) satisfy $f_1 \geq f_2$. Thus, since $f_1 + f_2 = 1$, we have $f_1 \geq 1/2$. Since G_2^o is connected, the edge connecting u (v) to other nodes in G_2^o uses at least power P_u^{min} (P_v^{min}). Consider an edge $e = (u, v) \in E_1^o$. Note that the power of node u and v in G_1^o is at least $P(e)$, since $e \in E_1^o$. Thus, the average power of u (v) is at least $f_2 P_u^{min} + f_1 P(e)$ ($f_2 P_v^{min} + f_1 P(e)$), which (since $f_1 \geq 1/2$ and since $P(e) \geq \max\{P_u^{min}, P_v^{min}\}$) is at least $0.5(P_u^{min} + P(e))$ ($0.5(P_v^{min} + P(e))$). Thus, the two power solution uses maximum average power of at least $0.5w(e)$, for all edges $e \in E_1^o \cup E_2^o$. ∎

Next, the nodes follow the distributed MST based topology construction algorithm (LMST) proposed in [7]. Each node runs the Kruskal's algorithm [2] on $G_u = (V_u, E_u)$ to construct an MST in its visibility graph. Kruskal's algorithm adds edges in increasing order of weight to merge disconnected components of nodes, until all nodes are connected. Then, for each edge (u, v) selected by node u on its MST for G_u, it informs node v that it has selected edge (u, v) in its MST. An edge is formed between two nodes u, v in the final topology if the edge (u, v) is present in the MSTs corresponding to both G_u and G_v. That is, if the edge (u, v) is not present in either one of the MSTs of G_u or G_v, the edge (u, v) is not formed in the topology. This is known as the G^- topology in [7]. The algorithm was proved to be min-max optimal in terms of the edge powers among the class of local algorithms (algorithms that do not use global communication) in [7]. That is, the maximum edge weight in the topology, with edge weight as the power needed to form that edge, is minimum for the topology constructed by this algorithm, among the class of all algorithms that only use local information. It can be shown that the algorithm is min-max optimal for our edge weight function as well. This is so because in our algorithm, the edges are added in an increasing order of weight in the MST of each visibility graph, and an edge (u, v) present in MST of G_u is not included in the final topology only if a path of smaller weight exists in the MST of G_v. Thus, the resultant topology is min-max optimal among the class of localized algorithms. It can be shown that our algorithm also yields a connected global topology if the network $G = (V, E)$ is connected. We use this topology as the first topology (call it G_1). The first topology provides a lower bound on an optimal localized two power solution, as stated in the following lemma:

Lemma 4.2: Any two power solution using localized communication uses maximum average node power of at least half the maximum edge weight $\max\{w(e)|e \in G_1\}$.

Proof: G_1 is min-max optimal among all algorithms that find a connected topology using localized communication. Thus, the maximum edge weight according to the weight function $w(e)$ among edges of any two power solution is at least $\max\{w(e)|e \in G_1\}$. Using Lemma 4.1, the maximum average power used in the two power solution is lower bounded by $0.5 \max\{w(e)|e \in G_1\}$, thus proving the result. ∎

Let the maximum power used in the first topology at node u be denoted by P_u^1. One possibility is to set the second node power P_u^2 also to P_u^1. This is equivalent to using the first topology twice (call it LMST duplication algorithm). In this case both the topologies will be connected, and the maximum average node power for every node u will be the same as the maximum node power P_u^1 in the first topology. Since the power of an edge is less than its edge weight $w(e)$, the algorithm would be a 2-approximation of an optimal localized two power assignment solution (using Lemma 4.2).

In the following we propose an alternative algorithm (not just LMST duplication) for assigning the second power P_u^2 (Phase II), so that the resulting two topologies are each strongly connected. This algorithm performs much better than

the LMST duplication algorithm in practice and at least as good as the LMST duplication algorithm in theory. Thus, our overall algorithm is also a 2-approximation of an optimal localized two power assignment algorithm but with better performance in practice.

Let the maximum power used in the first topology at node u be P_u^1. In the proposed algorithm (Phase II), each node u broadcasts P_u^1, using power P_{max}. Using the received information, each node u calculates the maximum power (P_{max}^{1u}) among the nodes in G_u. That is, $P_{max}^{1u} = \max_{v \in V_u} P_v^1$. Then, each node u broadcasts P_{max}^{1u} using power P_{max}. On receiving these powers each node u computes the maximum of these received powers as $P_{max}^{2u} = \max_{v \in V_u} P_v^{1u}$. Next any pair of nodes u_1, u_2 with an edge e between them in E calculate the maximum power (P_e^1) used in the first topology among all visibility graphs they are a part of. That is, $P_e^1 = \max\{P_{max}^{2u_1}, P_{max}^{2u_2}\}$. Then, u_1 and u_2 broadcast this maximum power P_e^1 for each such edge e, using power P_{max}. Next, each node u assigns the weight of each edge $e = (u_1, u_2)$ in E_u as $w'(e) = \max\{P_{u_1}^1, P_{u_2}^1\} + P(e)$, representing the edges view of the maximum average powers if this edge is included in the second power topology.

For a given set of edge weights $w'(e)$ $e \in E$ the algorithm considers edge $e_i = (u_i, v_i)$ to be heavier than edge $e_j = (u_j, v_j)$ if either $w'(e_i) > w'(e_j)$ or $u_i > u_j$ and $w'(e_i) = w'(e_j)$ or $v_i > v_j$ and $w'(e_i) = w'(e_j)$, $u_i = u_j$. Then, each node u removes an edge e from its visibility graph G_u if $w'(e) > 2P_e^1$. The LMST algorithm is then executed on these visibility graphs, and the G^- topology is constructed. This topology is used as the second topology (call it G_2). Then, each node u calculates its two powers P_u^1 and P_u^2 as the maximum power among edges connected to it (the node u) in the first and second topologies respectively. Each node then sets the two fractions $f_1 = f_2 = 1/2$. Algorithm 1 describes the algorithm for constructing the two topologies. We also explain the algorithm with the help of an example shown in Figure 1. Figure 1(a) shows the input topology, with power needed for each edge marked against it. We assume all nodes can see the full graph in their visibility graphs. In Phase I, the edge weights are calculated by adding to the edge power the maximum over the two end-points of their minimum neighbor distances (power to reach the neighbor). Figure 1(b) shows the edge weights, and Figure 1(c) shows the locally constructed topology (using LMST), which is used as the first topology. Nodes A and B use a power of two, while nodes C and D use a power of one. In Phase II, these powers are used to update the edge weights (by adding maximum over the end-points to the edge powers), and a topology is constructed using LMST on these edge-weights. Figure 1(d) shows the edge weights, and Figure 1(e) shows the second constructed topology. Thus, the average power used over the two topologies is 1.5 for all nodes. The LMST duplication algorithm (that duplicates the first topology/node powers) would result in the maximum node power average being 2 (for nodes A and B).

We now prove that the algorithm yields strongly connected topologies. Lemma 4.3 states the result:

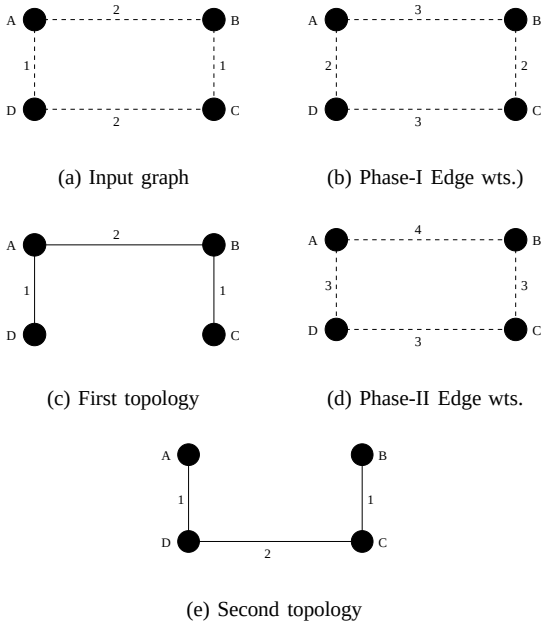

Fig. 1. Example execution of the localized algorithm

Lemma 4.3: The topologies of the two-power solution $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are strongly connected if the graph $G = (V, E)$ is strongly connected on the set of nodes V.

Proof: The first topology G_1 is computed using the LMST algorithm, with edge weights $w(e)$ as defined before. The topology has been proved to be strongly connected for any set of unique edge weights [7], if G is strongly connected. Thus, G_1 is strongly connected if G is strongly connected.

In Phase II, the edge weights are changed to $w'(e)$, and they are still unique. The input graph G is modified by pruning some edges locally. The pruning condition for each edge is the same for all visibility graphs G_u containing the edge. Therefore, an edge is either pruned from all visibility graphs it is a part of, or is not pruned from any of them. Thus, effectively, the edges are pruned from the global graph G to yield a graph $G' = (V, E')$. The pruning condition for an edge e is that the edge weight $w'(e)$ should not exceed twice the maximum node power in topology G_1 among nodes in the visibility graphs $G_u = (V_u, E_u)$ such that $e \in E_u$. For every edge $e = (u, v) \in E_1$, $w'(e) \leq 2\max\{P_u^1, P_v^1\}$ since $P(e) \leq \max\{P_u^1, P_v^1\}$, and thus it will not be pruned. Thus, all the edges in E_1 will be in E'. Since $G_1 = (V, E_1)$ is connected if G is connected and $E_1 \subseteq E'$, $G' = (V, E')$ is connected if G is connected. The algorithm of Phase II finally constructs a topology (G_2) using the LMST algorithm on G' with weights $w'(e)$. Since G' is connected if G is connected, G_2 is connected as well if G is connected ■

We now prove that the algorithm is a 2-approximation of an optimal localized two power assignment algorithm. Lemma 4.4 states the result.

Lemma 4.4: The two power assignment solution con-

Algorithm 1 Localized construction of two power solution

1: Each node broadcasts its location information.
2: Each node u constructs a visibility graph $G_u = (V_u, E_u)$ on its set of neighbors $N(u)$. Here, $V_u = \{u\} \cup N(u)$ and $E_u = \{e = (u_1, u_2) | u_1, u_2 \in V_u, e \in E\}$.
3: Each node u broadcasts the required power to reach its nearest neighbor, P_u^{min}.
4: Each node assigns weight $w(e) = \max\{P_{u_1}^{min}, P_{u_2}^{min}\} + P(e)$ to each edge $e = (u_1, v_1) \in G_u$.
5: Next, each node executes the LMST algorithm to compute the first topology G_1. Denote the power used at each node u by P_u^1.
6: Each node broadcasts P_u^1.
7: Each node u calculates $P_{max}^{1u} = \max_{v \in V_u} P_v^1$, and broadcasts it.
8: The end-points of every edge $e \in E$, u_1, u_2, calculate P_e^1 based on the maximum received power, and broadcast it.
9: Each node u constructs a new visibility graph $G'_u = (V_u, E'_u)$. Here, $V_u = \{u\} \cup N(u)$, and $E'_u = \{e = (u_1, u_2) | u_1, u_2 \in V_u, w'(e) < 2P_e^1\}$, $w'(e) = \max\{P_{u_1}^1, P_{u_1}^1\} + P(e)$.
10: Each node u executes the LMST algorithm, using visibility graph G'_u and weights $w'(e)$. The resulting topology is assigned as the second topology G_2.
11: Each node u uses the first power P_u^1 as the transmission power among edges adjacent to it in G_1, and the second power P_u^2 as the transmission power among edges adjacent to it in G_2.
12: Each node sets $f_1 = f_2 = 1/2$.

structed by our algorithm has the maximum average power of any node less than twice the maximum average power in a solution constructed by an optimal localized algorithm.

Proof: We prove this by proving that the maximum average power at a node cannot be greater than the maximum average for the LMST topology duplication algorithm. The result follows by Lemma 4.2. Note that the edge weight $w(e)$ in Phase I is at least $P(e)$. Thus the maximum node power used in Phase I ($\max_u P_u^1$) is at most $\max\{w(e) | e \in G_1\}$. By Lemma 4.2 this is at most twice the maximum node power used by any locally optimal solution. The edge pruning condition in Phase II is to remove each edge e for which its weight $w'(e)$ exceeds twice the maximum power used in the first tree over all the nodes in all visibility graphs G_u that contained e in Phase I. Since the maximum power used in G_1 is at least as much as the maximum over these visibility graphs, there is no edge in the network that has a weight more than twice the maximum power used in G_1. Thus, the maximum weight in the resulting topology cannot exceed the objective value attained by duplicating the first topology. Thus, the solution is a 2-approximation of an optimal two power solution constructed using localized communication. Another thing to note is that in the second phase, LMST is executed with edge weights representing the sum of powers at each

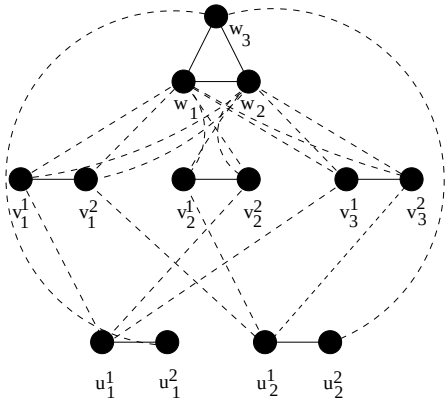

Fig. 2. An example reduction from 3-SAT.

node. Since LMST is min-max optimal in terms of the input edge weights, the solution is much better than duplicating the first LMST topology (that used different edge weights) in practice. ■

V. Hardness Results

Lemma 5.1: The two power assignment problem is NP-hard and hard to approximate to a factor strictly better than 2 when we do not assume any property of radio propagation. Moreover for this problem, unless P = NP, no polynomial time algorithm can be strictly better (in the worst case metric) than an optimal algorithm for the single power assignment problem.

Proof: We show a polynomial time reduction from 3-SAT which is known to be NP-complete [4]. The 3-SAT problem is to determine if a satisfying truth assignment exists for the given 3CNF formula. A 3CNF formula is defined by variables $x_1, x_2, \ldots x_n$ and clauses $C_1, C_2, \ldots C_m$ each a disjunction of exactly 3 literals over the set of variables. Given a 3-SAT instance, the reduction proceeds as follows.

For each variable x_i we create two literal nodes v_i^1 and v_i^2 and an edge (v_i^1, v_i^2). The former corresponds to the literal x_i and represents x_i being true in the satisfying assignment. The latter corresponds to the literal $\overline{x_i}$ and represents x_i being false in the satisfying assignment. Likewise for each clause C_i we create two clause nodes u_i^1 and u_i^2 and an edge (u_i^1, u_i^2). There are three more nodes w_1, w_2, w_3 with 3 edges connecting all pairs of nodes (w_i, w_j). All the edges e described so far are low power edges with edge power $P(e) = \epsilon$ for some small value ϵ.

In addition to the low power edges described above there are some high power edges e described below. Each high power edge e has $P(e) = X$ for some X much larger than ϵ. The node u_i^1 for each clause C_i has 3 such edges incident on it. The other end-points of these edges are the literal nodes corresponding to the literals in the clause C_i. The nodes w_1 and w_2 each have one such edge to each of the literal nodes. The node w_3 has such an edge to each of the clause nodes u_i^2. This completes our description of the reduction. Figure 2 shows an example reduction from the 3-SAT problem $(x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$.

Let the 3-SAT instance have a satisfying assignment. Let S be the set of literal nodes for each variable in this satisfying

assignment such that S contains node v_i^1 if x_i is true in the satisfying assignment and contains node v_i^2 otherwise. We denote by $\overline{S}$ the set of other literal nodes. Consider the network T_1 formed by taking all the low power edges e (with power $P(e) = \epsilon$) and the high power edges (with power $P(e) = X$) connecting the nodes in S with w_1 and the high power edges connecting the nodes in S with the nodes u_i^1 for each clause C_i. Consider also the network T_2 formed by taking all the low power edges e and the high power edges connecting the nodes in $\overline{S}$ with w_2 and the high power edges connecting the node w_3 with the nodes u_i^2 for each clause C_i. It can be verified that the edges incident on any node v are all low powered ($P(e) = \epsilon$) in one of the networks T_1 or T_2. Thus the average power needed at any node v is at most $(X + \epsilon)/2$. In addition it can be verified that both T_1 and T_2 are strongly connected since all the clauses are satisfied and the clause nodes u_i^1 must be connected to at least one node in the set S. Thus, there is a feasible solution to the two power assignment problem instance with average node power at most $(X + \epsilon)/2$.

Now consider the case when there does not exist a satisfying assignment. Let T_1 and T_2 be a feasible solution to the two power assignment problem instance. Thus, both T_1 and T_2 are strongly connected. We establish that there exists a node v which is incident on high powered edges ($P(e) = X$) in both T_1 and T_2. Let no such node exists. Consider the network where node w_2 has no high powered edges incident on it. W.l.o.g. let this be network T_1. Note that in T_1 there must be a high powered edge from every clause node u_i^1 to some literal node. Let S be the set of literal nodes with an high powered edge to some clause node u_i^1 in T_1. Note that S must have both the literal nodes v_j^1 and v_j^2 corresponding to some variable x_j, because otherwise the literal nodes in S will correspond to a satisfying assignment, which by our assumption does not exist. But then v_j^1 and v_j^2 cannot be incident on any high powered edges in T_2 and hence must be disconnected from the rest of the network in T_2. Thus in this case T_2 cannot be strongly connected, a contradiction. Thus, we have established that there exists a node v that is incident on high powered edges ($P(e) = X$) in both T_1 and T_2. The average power of node v is therefore X in the solution T_1 and T_2.

Thus, we have shown that if the 3-SAT instance has a satisfying assignment then the two power assignment problem instance has a solution of power $(X + \epsilon)/2$. Otherwise even the best power solution to the two power assignment problem instance has a solution of power X. Since ϵ can be made arbitrarily small, this establishes that not only is the two power assignment problem NP-hard but cannot be approximated to a factor better than 2. Also there does not exist any algorithm for the two power assignment problem that can have strictly better performance than finding the one power solution using the optimal minimum spanning tree construction algorithm. ■

VI. Simulation Results and Discussion

We investigate the performance of the following algorithms: the centralized MST duplication (CMSTD), centralized approximation algorithm of Section III (CAA), localized LMST based topology duplication (LTD), and the localized algorithm proposed in Section IV (LA). To characterize the performance impact of various parameters, we vary the number of nodes, maximum transmission power (P_{max}), and the path loss factor.

To get a sense of how far our algorithms are from the optimal, we normalize our results with a lower bound we compute for each instance of the problem. The first lower bound is the maximum nearest neighbor distance (power required to reach the neighbor) over the nodes in the network. At least this power will be needed in each tree to connect the node which achieves this maximum. The second lower bound is half the maximum edge length (power) in the MST constructed using the Kruskal's algorithm. We take the lower bound to be the maximum of the two lower bounds. The first lower bound dominates when there is an isolated node far from all other nodes. Otherwise, the second lower bound dominates usually. We normalize the objective value of each algorithm by this lower bound.

A. Variation with Number of Nodes

For the first set of simulations, the network is a unit square. The path loss factor α is fixed at 2, and the number of nodes (N) is varied from 40 to 200. The maximum transmission power of each node is such that it can communicate with a node half unit distance away. The transmission range is chosen so that the global topology at maximum transmission power is connected (which is difficult to guarantee when nodes are placed randomly). The network area is divided in four grids, and one fourth of the nodes are placed in each grid. The location of the nodes are chosen uniformly randomly in the grid they are placed in. We generate 20 sets of locations for each value of N. For $N_2 > N_1$, the set of locations i for N_2 will have N_1 nodes at the same locations they were at in set i generated for N_1.

Figure 3 shows the average performance of the algorithms, relative to the lower bound on the optimal, over these 20 sets of locations. Results show that the centralized approximation algorithm (CAA) performs better than the MST duplication algorithm (CMSTD), as expected from the theoretical analysis of CAA (for $\alpha = 2$, CAA is a 1.76-approximation while CMSTD is a 2-approximation). The localized topology duplication algorithm (LTD) performs the worst among these, as it follows the same algorithm as CMSTD, but with local information at each node. The localized algorithm proposed in Section IV (LA) has a significant improvement over LTD (we proved that LA will always do at least as good as LTD). LA also outperforms CAA, since we lose some performance in CAA when each node connects to the MST neighbors of the nodes which chose it as their buddy. This leads to a few nodes having a higher average power in practice compared to LA. Although, as we will show later, if the maximum transmission range is small, then CAA outperforms LA by a significant amount due to non-availability of critical global information locally in LA. The performance of LA relative to the lower bound on optimal global solution improves as the number of nodes increases. This is because as the node density increases,

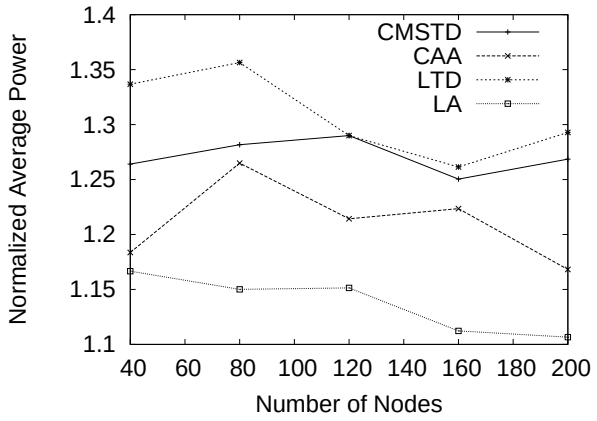

Fig. 3. Variation with number of nodes in network of unit length, $\alpha = 2$.

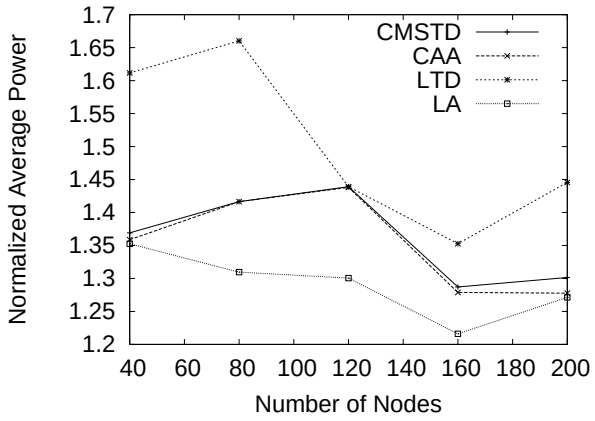

Fig. 4. Variation with number of nodes in network of unit length, $\alpha = 4$.

more edges are available for replacing long edges chosen in the first power topology. LA performs within 10-15% of the lower bound for the global optimal, which is very impressive. The worst case performance of LA is within twice the lower bound on the centralized optimal, while it is within 2.7 times the lower bound for other algorithms. Note that the optimal performance is worse than the lower bound for each instance, thus even though the algorithms are 2-approximations (CAA is even better) of the optimal, they can have a performance degradation worse than two with respect to the lower bound on some instances.

B. Performance at Different Path Loss Factor

We now change the path loss factor α to four, and perform the simulations on the same sets of node locations. The maximum transmission power is increased so that each node can reach a node half a unit distance away. This ensured the connectivity of the network at maximum transmission power for the node locations instances in our simulations. Figure 4 shows the average performance of the algorithms relative to the lower bound on the optimal. The algorithms do not perform as well (compared to the lower bounds) as for $\alpha = 2$. The degradation is due to the squaring of the cost function as α is increased from 2 to 4. In particular, the centralized approximation algorithm (CAA) does not give a significant performance gain over the centralized MST duplication algorithm (CMSTD) for $\alpha = 4$. The degradation of CAA's performance relative to the lower bound as well as CMSTD with α agrees with the analysis. The analysis proved the approximation ratio to be 1.76 for $\alpha = 2$, 1.966 for $\alpha = 3$, and would yield a factor closer to 2 for $\alpha = 4$ (CMSTD has an approximation ratio of 2). The performance of the algorithms with respect to each other is still the same, with LA performing the best, staying within 35% of the lower bound on the optimal.

C. Performance at High Maximum Transmission Range

We now double the maximum transmission range, for $\alpha = 2$ (thus increase P_{max} four times). This also leads to each node having almost a global view of the topology. The rest of the simulation set-up is the same as for the first set of simulations. Figure 5 shows the average performance of the algorithms relative to the lower bound on the optimal. Since the local algorithms have a global view of the network, CMSTD and LTD have the same performance. Due to the same reason, the performance of LA also improves compared to the first set of simulations. The performance of CMSTD and CAA remains the same.

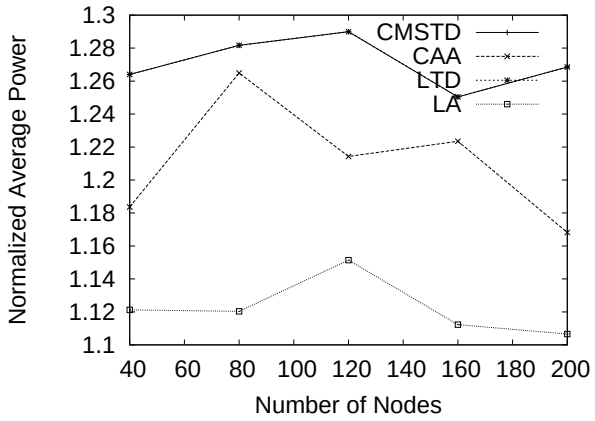

Fig. 5. Variation with number of nodes in network with high transmission range, $\alpha = 2$.

D. Variation with Maximum Transmission Range

We now fix the number of nodes, the path loss factor α, and vary the maximum transmission range from 0.3 to unit distance in a unit length network of 40 randomly distributed nodes. We simulate the node locations (divided into four grids) 20 times (if the network G is disconnected in some instance, we simulate another set of locations to replace that). Figure 6 shows the average performance of the algorithms relative to the lower bounds. Results show that the centralized algorithms outperform the localized algorithms by a significant amount at low maximum transmission powers. This is expected as the localized algorithms have less information about the network topology locally at low maximum transmission range. As the maximum range increases, the localized algorithm LA starts performing better than the centralized algorithms, and the

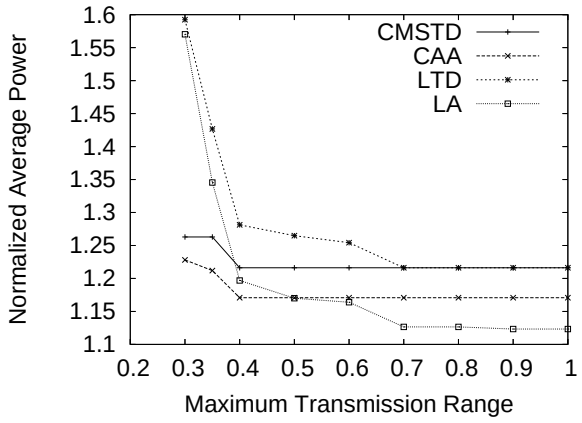

Fig. 6. Variation with maximum transmission range in unit length networks, $N = 40, \alpha = 2$.

localized LMST duplication algorithm (LTD) performs the same as centralized MST duplication (CMSTD) for maximum transmission ranges above 0.7. The performance of the centralized algorithms relative to the lower bounds is constant above a certain transmission range. This is because for each instance, the minimum weight topologies do not need edges larger than a certain weight, and thus increasing maximum transmission range (power) above a threshold does not change the solution. In fact, the performance becomes almost the same for the localized algorithm LA as well, due to the same reason.

E. Result Summary

Our localized algorithm LA typically excels when the transmission range is large in which case, it has more information to make better decision. With very small transmission range, the centralized algorithm CAA performs much better than LA. CAA performs much better than CMSTD for low α, as the analysis proved. However, as the analysis showed, the relative performance degrades with increasing α. The LA algorithm does surprisingly well when compared with the global lower bound for large maximum transmission power, and significantly outperforms the other localized algorithm, LTD, for all values of maximum transmission powers considered.

VII. Related Work

There is a vast literature on the use of power assignment to improve energy efficiency in multi-hop wireless networks. The most closely related to ours are topology control [9], [6], [7], [5] and network lifetime optimization.

Topology control algorithms [5] typically make use of localized algorithm and try to either minimize the sum of powers over all nodes, or minimize the maximum transmission power among all nodes. In the meantime, the algorithms try to preserve certain nice properties such as spanner and connectivity. In particular, the LMST [7] based approach tries to min-max transmission power using localized distributed algorithm. We extend the algorithm for multiple power assignment with the objective of "power balancing" among nodes.

The work [3] tries to maximize the lifetime of multicast sessions. This is achieved by a schedule of power assignments, i.e. different power is used in different time slots. Our problem is different from theirs in that, our power assignment is used by all applications, not just multicast. Our power assignment function sits between the routing layer and link layer. Thus, we are oblivious of the application traffic. To appreciate the difference, for a single power assignment, their problem is NP-hard while ours can be solved optimally in polynomial time. The work [1] seeks to maximize the network lifetime with a set of power assignments where each power is used at different time slots. They do not limit the number of powers used. In this paper, we consider a more practical setting where nodes alternate between two transmission powers.

VIII. Conclusion and future work

Power balancing among nodes in energy-constraint multi-hop wireless networks is crucial for longevity of the network. A solution that sits between network layer and link layer is oblivious to the application and can be used in a wider context. We present such an approach where nodes at the 2.5 layer alternate different power assignments in an attempt to minimize the energy expended dynamically. In particular, for unicast applications, each forwarding node can send the packet using a different transmission power independently using the available path to the destination with that power. For broadcast applications, the source will dictate which power should be used by intermediate nodes to maintain network connectivity. Our simulation results demonstrate the power of multiple power assignments. For future work, we would like to explore the system aspects of this problem and plan to implement our solution in a real sensor network test-bed.

IX. Aknowledgement

The problem came up during some conversations between Li (Erran) Li and Alon Efrat at Univ. of Arizona. We would like to thank Alon for initial stimulating discussions.

References

[1] Gruia Calinescu, Sanjiv Kapoor, Alexander Olshevsky, and Alexander Zelikovsky. Network lifetime and power assignment in ad hoc wireless networks. In *Proc. ESA*, pages 114–126, 2003.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[3] P. Floreen, P. Kaski, J. Kohonen, and P. Orponen. Lifetime maximization for multicasting in energy-constrained wireless networks. *IEEE Journal on Selected Areas in Communications*, 23(1):117–126, 2005.

[4] Michael R. Garey and David S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[5] L. Jia, R. Rajaraman, and C. Scheideler. On local algorithms for topology control and routing in ad hoc networks. In *Proc. SPAA*, pages 220–229, 2003.

[6] Li Li, Joseph Y. Halpern, Paramvir Bahl, Yi-Min Wang, and Roger Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *Proc. PODC*, pages 264–273, 2001.

[7] Ning Li, Jennifer Hou, and Lui Sha. Design and analysis of an MST-based topology control algorithm. In *Proc. IEEE INFOCOM*, pages 1702–1712, 2003.

[8] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 1996.

[9] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Communications*, 17(8):1333–1344, August 1999.