Minimum-Energy Broadcasting in Multi-hop Wireless Networks Using a Single Broadcast Tree

Ioannis Papadimitriou and Leonidas Georgiadis

Abstract

In this paper we address the minimum-energy broadcast problem in multi-hop wireless networks, so that all broadcast requests initiated by different source nodes take place on the *same* broadcast tree. Our approach differs from the most commonly used one where the determination of the broadcast tree depends on the source node, thus resulting in different tree construction processes for different source nodes. Using a single broadcast tree simplifies considerably the tree maintenance problem and allows scaling to larger networks. We first show that, using the same broadcast tree, the total power consumed for broadcasting from a given source node is at most twice the total power consumed for broadcasting from any other source node. We next develop a polynomial-time approximation algorithm for the construction of a single broadcast tree. The performance analysis of the algorithm indicates that the total power consumed for broadcasting from any source node is within $2H(n-1)$ from the optimal, where n is the number of nodes in the network and $H(n)$ is the harmonic function. This approximation ratio is close to the best achievable bound in polynomial time. We also provide a useful relation between the minimum-energy broadcast problem and the minimum spanning tree, which shows that a minimum spanning tree may be a good candidate in sparsely connected networks. The performance of our algorithm is also evaluated numerically with simulations.

Index Terms

Wireless Networks, Minimum-Energy Broadcast, Spanning Trees, Approximation Algorithms, Performance Analysis

I. Introduction

The field of infrastructureless wireless multi-hop networks has attracted significant attention by many researchers in the recent years because of its large number of new and exciting applications. However, the technical challenges that arise pose many new problems and issues that have to be addressed when designing a network in this field [1], [2]. Such an issue is the efficient management of the available energy resources.

A preliminary version of this work appeared in the *Proceedings of WiOpt'04: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, University of Cambridge, UK, March 2004.

I. Papadimitriou and L. Georgiadis are with Electrical and Computer Engineering Department, Aristotle University of Thessaloniki, GREECE. E-mails: ipapad@egnatia.ee.auth.gr , leonid@eng.auth.gr .

I. Papadimitriou was fully supported for this work by the Public Benefit Foundation "ALEXANDER S. ONASSIS", Athens, GREECE.

One important distinction as to how energy consumption must be taken into account is whether energy is viewed as an expensive (but renewable) commodity or as a finite (and nonrenewable) resource [3].

In this paper we focus on the problem of energy-efficient broadcasting in wireless networks where omnidirectional antennas are used and there is flexibility of power adjustment. As indicated in one of the pioneer works by Wieselthier *et al.* in [4], broadcasting in a wireless environment where omnidirectional antennas are used, must take into account the fact that a node's transmission can reach multiple neighbors at the same time. Hence, the power needed to reach a node's set of neighbors is the *maximum* of the powers needed to reach each of the neighbors separately. Given a specific source node that initiates a broadcast request, the problem of determining a set of retransmitting nodes and their corresponding transmission powers, such that the sum of consumed node powers is minimized, is known as the *minimum-energy broadcast* problem.

Although the problem of minimum-energy broadcasting has been studied extensively in the literature (see Section II for references to prior work), most of previous approaches provide a solution for it which depends on the source node that initiates the broadcast request. That is, every time a node needs to broadcast some information to all other nodes in the network, the algorithm for the broadcast tree construction is executed for the specific source node. In general, for different source nodes, the trees that minimize the total power consumption are different (see Section III-B for an example). Hence, in general, each node in the network has to keep track of n broadcast trees, one for each of the possible source nodes (n is the number of nodes in the network). This requires large memory space and/or processing capabilities on behalf of the nodes in the network, a demand that cannot always be met.

The above situation will be greatly simplified if one can define a *single* broadcast tree, on which broadcasting initiated by *any* source node will take place in a predetermined manner. Hence, in our setup we are interested in selecting a unique broadcast tree that keeps the total power consumption as small as possible for any broadcast request. In this manner, a node needs to store only a small set of links that belong in the selected tree and the processing of broadcast information is minimal. More specifically, whenever a node receives a broadcast message for the first time in one of its tree links, it forwards it with appropriate power to all its neighbors in the tree except the one from which the message was received. Note that this is exactly how a Connected Dominating Set (CDS) would work in case we did not have the flexibility of power adjustment (see for example [5], [6], [7]). In this case, a single CDS is determined for the network and each node needs only to know whether it belongs to this set or not.

When nodes initiate broadcast requests at the same time, it may seem that the use of a single tree results in more collisions compared to the approach of using different trees for different source nodes. However, this is not necessarily the case. Indeed, the use of omnidirectional antennas implies that independently of the approach used (single or multiple broadcast trees) a node's transmission will interfere with its neighbors' transmissions or receptions. Hence, whether the node retransmission is always intended to particular destinations (in case of single broadcast tree) or to different destinations (according to the source node in case of multiple broadcast trees), all nodes in the neighborhood will be affected and the lower level issue of collision resolution does not create a bias towards one of the methodologies. Network instances and particular broadcast scenarios can be created where one approach is better or worse than the other.

There are two open issues with our approach that have to be answered. First, if all broadcast requests take place on the same tree, then this may result in widely varying total powers for different source nodes. Second, even if a tree is found without having the drawback of resulting in widely varying total powers for different source nodes, then, for a specific source node the resulting total power consumption may be far away from the optimal. We address both issues in Section III-B and provide satisfactory answers to them in Section IV. More precisely, we first show that if the same tree is used for broadcasting by all nodes, then the total power consumed for broadcasting from a given source node is at most twice the total power consumed for broadcasting from any other source node. Next, we develop a polynomial-time approximation algorithm for the construction of a single broadcast tree. The performance analysis of the algorithm indicates that the total power consumed for broadcasting initiated by any source node is at most $2H(n-1)$ times the optimal ($H(n)$ is the harmonic function), which is close to the best achievable approximation factor in polynomial time. This bound is better than any other we are aware of, even for the case of different broadcast trees for every possible source node. Moreover, it is valid for general networks, with arbitrary weights on links between nodes, which do not rely on unit disk graphs and geometric properties of the Euclidean space. This is a more realistic model since, for example, the power needed for communication between two pairs of nodes with equal distances between the nodes of each pair, may not be the same due to noise or other signal propagation phenomena. We also show that the performance of the *minimum spanning tree* [8] is within Δ times the optimal, where Δ is the maximum node degree in the network. Hence, a minimum spanning tree can also be used for broadcasting by all nodes in sparse networks.

Numerical results for various networks with different sizes are presented in Section V. The main performance metric of interest is the total broadcast power consumption for different source nodes. It is shown that our algorithm provides fairly satisfactory performance for networks represented by unit disk graphs. However, we note that our algorithm outperforms significantly other algorithms for some interesting instances of general networks and, therefore, it presents a good compromise between simplicity and achieved performance.

The rest of the paper is organized as follows. In Section II we give some references to prior work related to the minimum-energy broadcast problem. Section III provides formal definitions and formulation of the problem. In Section IV we develop a polynomial-time approximation algorithm and prove that it achieves a satisfactory approximation ratio regarding the metric of total power consumption. We also provide a useful for sparse networks relation between the minimum-energy broadcast problem and the minimum spanning tree. Numerical results are presented in Section V. Finally, Section VI summarizes the conclusions of our work and presents some interesting issues for further study. All proofs of the lemmas provided in the paper are given in the Appendix.

II. Related Work

The minimum-energy broadcast problem in wireless networks has received significant attention over the last few years. The work by Wieselthier *et al.* [4] exploits the "node-based" nature of wireless communications and introduces the notion of "wireless multicast advantage". One of the most notable contributions of the work in [4] is the Broadcast Incremental Power (BIP) algorithm. BIP constructs a broadcast tree starting from the source node and adding to the tree one node at every iteration. The selection of which uncovered node will be added to the tree is based on the *minimum additional cost* criterion. Numerical results demonstrate the advantages of BIP over conventional link-based schemes, but a performance analysis of the algorithm is not provided. In [9] the general combinatorial optimization problem, called Minimum Energy Consumption Broadcast Subgraph (MECBS), is considered. It is proved that MECBS is not approximable within a sub-logarithmic factor (unless NP has slightly superpolynomial time algorithms) and a polynomial-time approximation algorithm is provided for special cases in the Euclidean space.

The NP-completeness of the minimum-energy broadcast problem is also proved in [10], [11], [12], [13]. In [10], [11], [13] various heuristic algorithms are proposed and their performance is compared numerically to that of BIP. Analytical performance results are not presented. The approximation algorithm developed in [12] and the analytical results that are provided depend on the number of adjustable power levels at each node. By exploring geometric structures of Euclidean MSTs, analytical results are also provided in [14] for BIP and other algorithms. Three different integer programming (IP) models that can be solved by any standard IP technique are proposed in [15] for the minimum-energy broadcast problem. The problem of minimum-energy broadcasting is also addressed in [16], a survey where an overview is presented of the recent progress in applying computational geometry techniques to solve some questions in wireless networks.

The first logarithmic approximation algorithm for the MECBS problem is presented in [17], where an interesting reduction to the Node-Weighted Connected Dominating Set problem is used. The proposed algorithm achieves a $10.8 \ln n$ approximation ratio for symmetric instances of MECBS, which is worse than ours. Moreover, the approach followed in [17] depends on the specific source node. An improved approximation ratio, which is closer to ours (but still, slightly worse), has been independently announced recently in [18]. The proposed algorithm improves the approximation ratio from $10.8 \ln n$ of [17] to $2 + 2\ln(n-1)$. However, this is also an approach for the broadcast problem in wireless networks which depends on the source node; on the other hand, the algorithm in [18] is applicable to networks with asymmetric power requirements.

The problem of constructing energy-efficient broadcast and multicast trees in an energy-, bandwidth-, and transceiver-limited wireless network is addressed in [19]. Similarities and differences between energy-limited and energy-efficient communications are described and the impact of these overlapping (and sometimes conflicting) considerations on network operation is illustrated. An approach to the problem of energy-aware broadcasting with emphasis on individual node power consumption is proposed in [20]. The lexicographic optimization criterion is introduced and the objective is to minimize lexicographically the consumed node powers or maximize lexicographically the residual node energies. We leave the issue of addressing our model in an energy-, and resource-limited environment as a subject for further study.

III. Definitions and Problem Description

Consider a connected undirected graph $G = (N, L)$, where N is the set of nodes and L is the set of undirected links. For a node $i \in N$ we denote by $L^G(i)$ the set of links adjacent to i. A node j such that link (i, j) belongs to L is called a one-hop neighbor of i or simply a neighbor of i. We denote by $N^G(i)$ the set of neighbors of node i. An undirected tree $T = (N, L^T)$ spanning G (spanning tree for short) is a connected acyclic subgraph that spans all the nodes. It follows from the definition of spanning tree that the number of links in T is $|L^T| = |N| - 1$.

A. Model for Wireless Broadcasting

We model the wireless network as a connected undirected graph $G = (N, L)$. N is the set of nodes in the network. If node j can successfully receive information transmitted by node i, and vice versa, then link (i, j) belongs to the set L of links in G. The power needed for a successful transmission over link $l = (i, j)$ is denoted by $c_l > 0$ and is also referred to as the link cost. Each node is equipped with an omnidirectional antenna. Hence:

Property 1: *If node i transmits with power p_i, it can reach any neighbor node j for which $c_{(i,j)} \leq p_i$.*

Note that in addition to the above power requirement, energy is also expended for transmission (encoding, modulation, etc.) and reception (demodulation,decoding, etc.) processing operations as indicated in [19]. For the analysis in the next sections, we assume that the energy consumption quantities for transmission and reception processing operations are small and thus can be neglected, an approach followed by many previous works referenced in Section II. However, we note that the incorporation of the aforementioned quantities into our model is not a major concern if the network nodes consume similar power for transmission (as well as reception) processing. More specifically, the fact that all nodes in the network (except the source node) receive the information in a broadcast process, results in adding a fixed quantity in the overall energy consumption. This does not affect the minimum-energy broadcast problem, since the total energy consumed for reception is always the same. Regarding the energy requirement for transmission processing operations, this quantity can be incorporated into the cost of each link without affecting our approach in the following sections.

Suppose that a source node s needs to broadcast some information to all other network nodes. In this case, we have to determine a set of retransmitting nodes and their corresponding transmission powers, so that eventually all nodes receive the information. A way to achieve this, which will be useful in the sequel, is as follows. Let T be a spanning tree of G. We define an s-rooted directed spanning tree $T_s = (N, L^{T_s})$ induced by T, with the following interpretation:

1) Node s uses all links in $L^T(s)$ as its outgoing links. We define $L^{T_s}_{out}(s) = L^T(s)$ and node s transmits with power $p^{T_s}_s = \max_{l \in L^{T_s}_{out}(s)}\{c_l\}$. Note that there is no link incoming to node s in the set L^{T_s}, that is, $L^{T_s}_{in}(s) = \varnothing$.

2) Any node i of T receiving the information in one of its tree links in $L^T(i)$, say link l, uses the set $L^T(i) - \{l\}$ as its outgoing links. Therefore, we define $L^{T_s}_{in}(i) = \{l\}$, $L^{T_s}_{out}(i) = L^T(i) - \{l\}$, and node i retransmits with power $p^{T_s}_i = \max_{l \in L^{T_s}_{out}(i)}\{c_l\}$. We refer to $p^{T_s}_i$ as the power induced on node i by tree T_s. If $L^{T_s}_{out}(i) = \varnothing$, then i is called a *leaf* node of T_s and $p^{T_s}_i = 0$.

Fig. 1 shows an example of the above definition. The directed spanning trees T_A, T_D, rooted at nodes

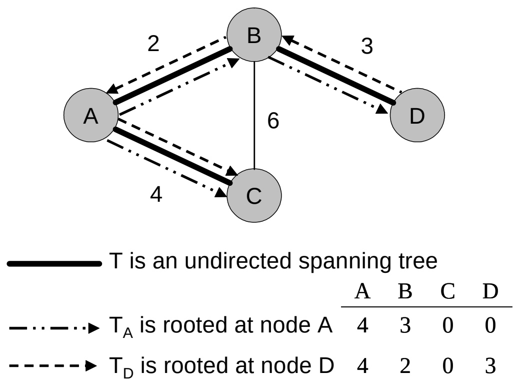

Fig. 1. Example of wireless broadcasting

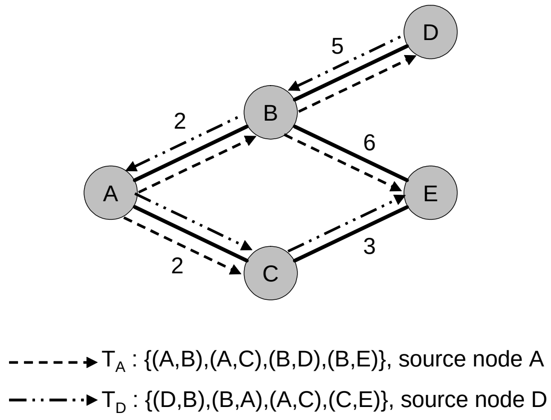

Fig. 2. Different minimum-energy broadcast trees for different source nodes

A and D respectively, are induced by the (undirected) spanning tree T with links {(A,B),(A,C),(B,D)}. Consider for example the tree T_A. The outgoing links of source node A in this case are {(A,B),(A,C)} and the outgoing link of node B is (B,D). Hence, the powers induced on nodes A and B by tree T_A are $p_A^{T_A} = 4$ and $p_B^{T_A} = 3$, respectively. Note that nodes C and D have no outgoing links in T_A and, therefore, $p_C^{T_A} = p_D^{T_A} = 0$. In a similar way, we have for tree T_D that $p_A^{T_D} = 4$, $p_B^{T_D} = 2$, $p_C^{T_D} = 0$, and $p_D^{T_D} = 3$.

B. The Minimum-Energy Broadcast Problem

Given an s-rooted directed spanning tree T_s, the total power consumed for broadcasting from source node s is $P(T_s) = \sum_{i \in N} p_i^{T_s}$. As discussed earlier, in general, for different source nodes the trees that minimize the sum of consumed node powers are different. Hence, each network node has to keep track of $|N|$ broadcast trees, one for every possible source node.

Consider for example the network in Fig. 2. The optimal (minimum-energy) broadcast trees for source nodes A and D are the trees T_A and T_D, respectively. The total power consumption for these trees is

$$P(T_A) = p_A^{T_A} + p_B^{T_A} = 2 + 6 = 8 \quad \text{and} \quad P(T_D) = p_D^{T_D} + p_B^{T_D} + p_A^{T_D} + p_C^{T_D} = 5 + 2 + 2 + 3 = 12.$$

Note that the underlying (undirected) spanning trees of T_A and T_D are different. If, for example, the source node D uses the underlying tree of T_A for broadcasting, then the sum of consumed node powers will be 13, larger than the optimal value 12 obtained from tree T_D.

The situation will be greatly simplified if one can define a single spanning tree T, on which broadcasting initiated by any source node will take place in a manner similar to the one described above. In this manner, a node i needs to store only the set $L^T(i)$ and processing of broadcast information is minimal. Hence, in our setup we are interested in selecting a unique spanning tree that keeps the total power consumption as small as possible for any source node.

There are two open issues with our approach that have to be answered; if all broadcasts (initiated by any source node) take place on the same tree, then:

Issue 1: Certain broadcasts may need much more total power than others, depending on the source node.

Issue 2: If one attempts to find a tree for which the total powers consumed for broadcasting initiated by different source nodes are approximately the same, then, given a certain source node, the resulting total power may be far away from the optimal.

In the section that follows we will first show that the first concern (widely varying total powers) is not a major problem. More precisely, we will prove that, given a spanning tree T, the total power consumed for broadcasting based on T from a source node s is at most twice the total power consumed for broadcasting from any other source node. Next, we will propose an algorithm for the construction of a spanning tree T, which has the desirable property that the resulting total power consumption for any source node is close to the computationally feasible factor from the optimal.

For the development and analysis of the algorithm presented below, we need the following general definition of tree cost. This is a purely technical definition and it has no physical interpretation.

Definition 1: *Let T be a spanning tree of G. We define $\mathcal{A}$ to be a link assignment to nodes in G, which associates with each node i a set of links $\mathcal{A}(i) \subseteq L^T(i)$, such that $\mathcal{A}(i) \cap \mathcal{A}(j) = \varnothing$, whenever $i \neq j$, and $\cup_{i \in N} \mathcal{A}(i) = L^T$. Under link assignment $\mathcal{A}$, we define the "power" of node $i \in N$ as $p_i^{\mathcal{A}} = \max_{l \in \mathcal{A}(i)} \{c_l\}$ and the cost of tree T as $P^{\mathcal{A}}(T) = \sum_{i \in N} p_i^{\mathcal{A}}$.*

Note that the broadcasting initiated by a given source node s using tree T corresponds to a particular link assignment $\mathcal{A}_s$, such that $\mathcal{A}_s(s) = L^T(s)$ and for each node $i \in N$, $i \neq s$, $\mathcal{A}_s(i) = L^T(i) - \{l\}$, where l is the link of T over which the broadcast information arrives at node i. That is, the s-rooted directed spanning tree T_s (see Section III-A for the original definition) can also be defined by tree T and link assignment $\mathcal{A}_s$.

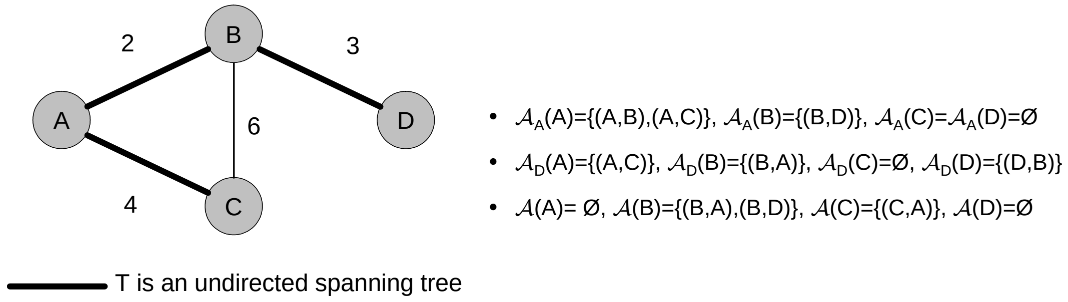

Fig. 3. Various link assignments for a given spanning tree T

Hence, we have that $P(T_s) = P^{\mathcal{A}_s}(T)$.

Fig. 3 shows an example of various link assignments for a given spanning tree T. Link assignments $\mathcal{A}_A$ and $\mathcal{A}_D$ correspond to broadcasting from source nodes A and D, respectively, using tree T. Therefore, it holds $P^{\mathcal{A}_A}(T) = P(T_A) = 7$ and $P^{\mathcal{A}_D}(T) = P(T_D) = 9$. In contrast to link assignments $\mathcal{A}_A$ and $\mathcal{A}_D$, $\mathcal{A}$ is an example of link assignment which does not correspond to any broadcasting process. However, according to Definition 1, the cost of tree T under assignment $\mathcal{A}$ is defined as $P^{\mathcal{A}}(T) = p_B^{\mathcal{A}} + p_C^{\mathcal{A}} = 3 + 4 = 7$.

IV. Broadcasting using a Single Broadcast Tree

A. Addressing Issue 1

In order to show that using the same tree for all broadcasts does not result in widely varying total powers for different source nodes, we first provide a useful lemma. The lemma that follows indicates that, given a spanning tree T, the cost of T under a link assignment that corresponds to broadcasting from a certain source node is at most twice the cost of T under any other link assignment.

Lemma 1: *Let T be a spanning tree of G. If $\mathcal{A}_s$ is a link assignment that corresponds to broadcasting from a given source node s using tree T and $\mathcal{A}$ is any other link assignment, then $P^{\mathcal{A}_s}(T) \leq 2P^{\mathcal{A}}(T)$.*

Consider now a source node $s' \neq s$. Since the cost of T under assignment $\mathcal{A}_s$, $P^{\mathcal{A}_s}(T)$, is at most twice the cost of T under any other link assignment, it follows that $P^{\mathcal{A}_s}(T)$ is also at most twice the cost of T under assignment $\mathcal{A}_{s'}$, which corresponds to broadcasting from source node s' using tree T. Hence, we have the following corollary:

Corollary 1: *If the same spanning tree T is used for broadcasting by all nodes, then the total power consumed for broadcasting from source node s is at most twice the total power consumed for broadcasting*

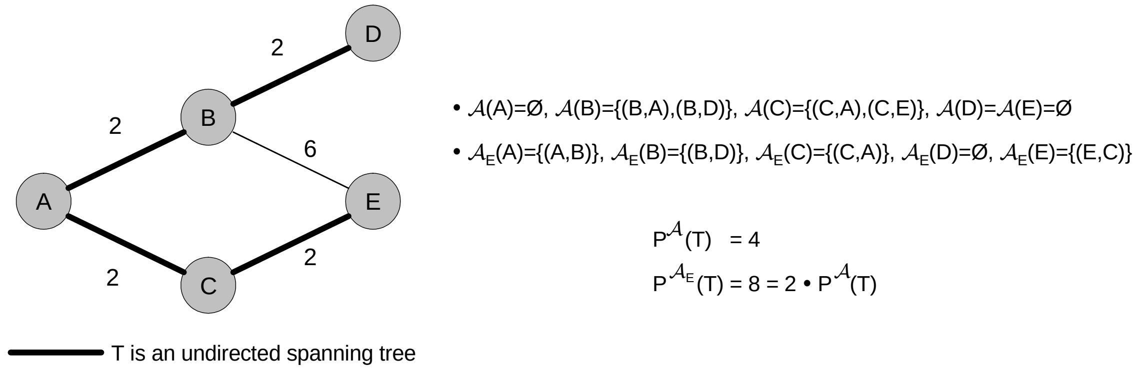

Fig. 4. A tight example for the result proven in Lemma 1

from any other source node. That is, for any two nodes s, s', $P(T_s) = P^{\mathcal{A}_s}(T) \leq 2P^{\mathcal{A}_{s'}}(T) = 2P(T_{s'})$.

Note that Lemma 1 is stronger than Corollary 1; it states that the right part of the inequality may concern *any* link assignment, not only assignments that correspond to broadcasting from a given source node.

A tight example for the result proven in Lemma 1 is provided in Fig. 4. Link assignment $\mathcal{A}_E$ corresponds to broadcasting from source node E using tree T, while $\mathcal{A}$ is a link assignment which does not correspond to any broadcasting process. For these two assignments, it holds

$$P^{\mathcal{A}}(T) = p_B^{\mathcal{A}} + p_C^{\mathcal{A}} = 2 + 2 = 4 \quad \text{and} \quad P^{\mathcal{A}_E}(T) = p_E^{\mathcal{A}_E} + p_C^{\mathcal{A}_E} + p_A^{\mathcal{A}_E} + p_B^{\mathcal{A}_E} = 2 + 2 + 2 + 2 = 8.$$

Therefore, $P^{\mathcal{A}_E}(T) = 2P^{\mathcal{A}}(T)$ and this example shows that the upper bound in Lemma 1 is a tight one. A similar example can also be constructed in case where the link assignment $\mathcal{A}$ corresponds to broadcasting from a particular source node. Consider three nodes A, B, C in tandem with link costs $c_{(A,B)} = c_{(B,C)} = 1$. Broadcasting from source node A needs power 2 (A to B and B to C), while broadcasting from source node B directly to nodes A and C needs power 1.

B. Addressing Issue 2

We now address the second issue, that is, the problem of selecting a tree such that the resulting total power consumed for broadcasting is not far away from the optimal for any source node. A problem closely related to the one of interest (see below for an explanation of this relation) is the following:

Problem 1: *Find a spanning tree T^* and a link assignment $\mathcal{A}^*$ such that, for any spanning tree T of G and any link assignment $\mathcal{A}$, it holds*

$$P^{\mathcal{A}^*}(T^*) \leq P^{\mathcal{A}}(T). \tag{1}$$

Note that if we use the tree T^* for broadcasting from source node s, we have according to Lemma 1 and inequality (1) that

$$P(T_s^*) = P^{\mathcal{A}_s^*}(T^*) \leq 2P^{\mathcal{A}^*}(T^*) \leq 2P^{\mathcal{A}}(T). \tag{2}$$

Consider now an optimal (minimum-energy) s-rooted directed spanning tree. As mentioned earlier (right after Definition 1), this tree can be defined by an undirected spanning tree and a particular link assignment. Since the total power consumed for broadcasting from source node s using tree T^*, $P(T_s^*)$, is at most twice the cost of T under assignment $\mathcal{A}$ (as indicated in (2)), where T is any spanning tree and $\mathcal{A}$ is any link assignment, it follows that $P(T_s^*)$ is also at most twice the optimal value. Therefore, T^* has the desirable property that the resulting total power consumption is not far away from the optimal for any source node.

Hence, we are led to the problem of determining T^* and $\mathcal{A}^*$. Unfortunately, this problem is NP-complete. The proof is based on a modification of the argument used to prove the NP-completeness of the Minimum Broadcast Cover (MBC) problem presented in [13] and is omitted due to lack of space[1]. The main idea is to reduce the weighted version of the *Set Cover* (SC) problem [21] to an instance of Problem 1 and show that the SC decision problem is satisfiable if and only if the decision problem of Problem 1 is satisfiable. The argument in the proof also shows that the reduction from SC to an instance of Problem 1 preserves approximation ratios. Hence, based on the corresponding result for Set Cover [22], Problem 1 is not approximable within $(1-\epsilon)\log(n-1)$, for any $0 < \epsilon < 1$, unless NP$\subseteq$DTIME($n^{O(\log\log n)}$), where DTIME(t) is the class of problems for which there is a deterministic algorithm running in time $O(t)$.

The discussion above suggests that instead of finding a tree and an assignment that solve Problem 1, we should attempt to find a good approximation. We present in Fig. 5 an approximation algorithm to construct a single spanning tree T which, as will be shown, has the desirable property in a worst case sense. At each iteration, the algorithm maintains a forest of trees in the network, such that each node i in G belongs to a forest tree $T_F = (N^{T_F}, L^{T_F})$ with link assignment $\mathcal{A}_F$. The node "power" and the cost of a forest tree are defined as described in Definition 1. Initially, each node constitutes a forest tree with no links assigned

[1]The complete proof is available upon request for anyone who is interested in a more detailed explanation.

Algorithm 1:

Initially each node i in G constitutes a forest tree T_F with link assignment $\mathcal{A}_F$ such that $N^{T_F} = \{i\}$, $L^{T_F} = \mathcal{A}_F(i) = \varnothing$, hence, $p_i^{\mathcal{A}_F} = 0$.

1. For each node i in G which belongs to a forest tree T_F with link assignment $\mathcal{A}_F$ **do**

Let $L'(i) = \{(i,j) \in L^G(i) : j \notin N^{T_F}\}$ (set of links adjacent to i and terminating outside the tree T_F).

For each link $l \in L'(i)$ **do**

i. Let $\mathbb{T}_i(l)$ be the set of distinct trees (other than T_F) that can be reached by node i when power c_l is used. Let also $B_i(l)$ be the set of links used by node i to reach these trees in the set $\mathbb{T}_i(l)$ (see also **Note 1**).

ii. Define $a_i(l) = (c_l - p_i^{\mathcal{A}_F})/|\mathbb{T}_i(l)|$.

end do

end do

2. Among all nodes $i \in N$, $L'(i) \neq \varnothing$, identify a node and a link with minimum value $a_i(l)$, say node $i_{\min}$ and link $l_{\min} \in L'(i_{\min})$. Let $a_{\min}$ be that value (note that, because of step 4 below, there exists at least one node i for which $L'(i) \neq \varnothing$, hence, $a_{\min}$ is well defined). That is, $a_{\min} = \min_{i \in N: L'(i) \neq \varnothing, l \in L'(i)} \{a_i(l)\}$. Let also $T_{F_{\min}}$ be the tree to which node $i_{\min}$ belongs and define $\mathbb{T} = \{T_{F_{\min}}\} \cup \mathbb{T}_{i_{\min}}(l_{\min})$.

3. Join node $i_{\min}$ with the trees in the set $\mathbb{T}_{i_{\min}}(l_{\min})$ using the set of links $B_{i_{\min}}(l_{\min})$. Hence, the trees in $\mathbb{T}$ are merged to a new tree T'_F with link assignment $\mathcal{A}'_F$ as follows: $N^{T'_F} = \cup_{T_F \in \mathbb{T}} N^{T_F}$, $L^{T'_F} = \left(\cup_{T_F \in \mathbb{T}} L^{T_F}\right) \cup B_{i_{\min}}(l_{\min})$, $\mathcal{A}'_F(i_{\min}) = \mathcal{A}_F(i_{\min}) \cup B_{i_{\min}}(l_{\min})$.

4. If there is a single forest tree T **then Return** T and its corresponding assignment $\mathcal{A}$. **Else**, **go to** step 1.

Fig. 5. Approximation algorithm for the construction of a single broadcast tree

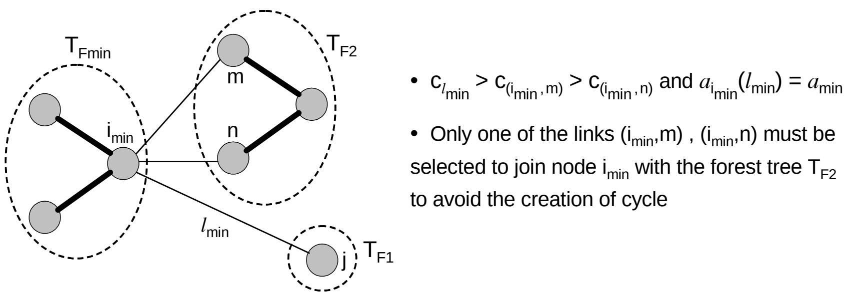

Fig. 6. Example of constructing a single broadcast tree using Algorithm 1

to it. At each iteration of the algorithm, the forest is expanded by joining trees through nodes so that the "incremental power consumed per joined tree", as defined in the algorithm, is minimal. This is achieved as follows. For every node i in the network, we examine its adjacent links that terminate outside the tree to which node i belongs. For such a link l, we define $\mathbb{T}_i(l)$ to be the set of distinct trees that can be reached by node i when power c_l is used (a tree is "reached" by node i, if i can reach at least one of its nodes). We also define $a_i(l)$ to be the ratio of the "additional power" required by node i to reach the trees in $\mathbb{T}_i(l)$ to the number of these trees. We select a node and a link for which the quantity $a_i(l)$ is minimal. If $i_{\min}$ is the selected node, then we join $i_{\min}$ with the appropriate trees. The set of links used for this union are assigned to node $i_{\min}$. The algorithm terminates when the forest consists of a single spanning tree T.

Note 1: When the set of links $B_i(l)$ is determined at step 1 of Algorithm 1, if node i can reach a forest tree through multiple links, then only one link is chosen to avoid the creation of cycles. Consider for example the network shown in Fig. 6. Node $i_{\min}$ is selected to be joined with forest trees T_{F1} and T_{F2}. Since $i_{\min}$ can be joined with T_{F2} through links $(i_{\min}, m)$ and $(i_{\min}, n)$, only one of these links must be chosen to avoid the creation of cycle with links already selected at previous iterations of the algorithm. Various selection criteria (e.g., choosing a link of minimal cost) can be used. In any case, we note that the worst case performance analysis of the algorithm is not affected by the criterion used to select a link. Moreover, the simulations that we performed showed only slight differences on the total power consumption for different selection criteria.

Note 2: Algorithm 1 can also be applied in the general case, where the graph G is not connected. In this case, the algorithm constructs a spanning tree for every component of the graph, which can be used for energy-efficient broadcasting inside the component. However, the condition for the termination of the algorithm at step 4 must be different. In the general case, the algorithm stops when there is no node i having at least one adjacent link that terminates outside the tree to which node i belongs, that is, when $L'(i) = \varnothing$ for every node $i \in N$.

Our algorithm uses the notion of "minimal incremental power consumed per joined tree". A similar notion has been used before for related problems. For example, finding the subset of "minimum weight per uncovered element" is the main idea of the well-known approximation algorithm for the weighted version of Set Cover problem [21]. A cost function similar to ours is also defined in [11], where the proposed algorithm constructs a clustering on the nodes using a function which represents the average cost induced per unmarked node. The node that (globally) has the most cost efficient range increase becomes a clusterhead and the nodes reached by the clusterhead, after its range increase, are marked. After a clustering has been found, they proceed in a second phase where they use a well-known algorithm for constructing directed minimum spanning trees [23] to join the clusters together. The algorithm in [11] computes a different broadcast tree for every possible source node and its worst case performance has not been established.

1) Performance Analysis of Algorithm 1: Let $\overline{T}$, $\overline{\mathcal{A}}$ be the tree and the corresponding link assignment returned by Algorithm 1. The next lemma shows that the cost of $\overline{T}$ under assignment $\overline{\mathcal{A}}$ has an approximation ratio $H(n-1)$ with respect to the cost of tree T^* under assignment $\mathcal{A}^*$ that solve Problem 1, where $n = |N|$ is the number of nodes in the network and $H(n)$ is the harmonic function $H(n) = \sum_{k=1}^{n} \frac{1}{k}$.

Lemma 2: *It holds* $P^{\overline{\mathcal{A}}}(\overline{T}) \leq H(n-1)P^{\mathcal{A}^*}(T^*)$.

Combining Lemmas 1, 2, and inequality (1), it follows that if we use the tree $\overline{T}$ for broadcasting from a given source node s, then we have for the total power consumption that

$$P(\overline{T}_s) = P^{\overline{\mathcal{A}}_s}(\overline{T}) \leq 2P^{\overline{\mathcal{A}}}(\overline{T}) \leq 2H(n-1)P^{\mathcal{A}^*}(T^*) \leq 2H(n-1)P^{\mathcal{A}}(T), \tag{3}$$

where T is any spanning tree and $\mathcal{A}$ is any link assignment of T. Since the optimal (minimum-energy) s-rooted directed spanning tree can be defined by an undirected spanning tree and a particular link assignment, and $P(\overline{T}_s)$ is at most $2H(n-1)$ times the cost of T under assignment $\mathcal{A}$ (as indicated in (3)), it follows that $P(\overline{T}_s)$ is also at most $2H(n-1)$ times the optimal value. Hence, we have the following corollary:

Corollary 2: *For any source node s, the total power consumed for broadcasting using tree $\overline{T}$ has an approximation ratio $2H(n-1)$ with respect to the optimal power.*

2) Complexity Analysis of Algorithm 1: For the complexity analysis that follows, we assume that the links adjacent to a node $i \in N$ are sorted in non-decreasing order of their costs. This can be made during initialization in $O(|L| \log |L|) = O(|L| \log |N|)$ time for all nodes in G.

Let us now provide the complexity for one iteration of the algorithm. For every node $i \in N$, such that $L'(i) \neq \varnothing$, step 1 requires the determination of the sets $\mathbb{T}_i(l)$, $B_i(l)$, and the computation of the quantities $a_i(l)$ for each $l \in L'(i)$. Step 2 requires the identification of node $i_{\min}$ and link $l_{\min}$. Since the adjacent links of a node $i \in N$ are sorted, we examine them in non-decreasing order of their costs. By defining appropriate variables, we can achieve an efficient implementation for steps 1 and 2, which requires the examination of the adjacent links of each node only once. Hence, steps 1-2 take time $O(\sum_{i \in N} |L^G(i)|) = O(|L|)$. In step 3, node $i_{\min}$ is joined with the trees in the set $\mathbb{T}_{i_{\min}}(l_{\min})$ using the set of links $B_{i_{\min}}(l_{\min})$. Recall that $\mathbb{T} = \{T_{F_{\min}}\} \cup \mathbb{T}_{i_{\min}}(l_{\min})$, where $T_{F_{\min}}$ is the tree to which node $i_{\min}$ belongs, and that the trees in $\mathbb{T}$ are merged to a new tree T'_F. We need $O(|N|)$ time to inform each node in the trees in $\mathbb{T}$ that it now belongs to the new tree T'_F, and $O(|L|)$ time to assign the links of the set $B_{i_{\min}}(l_{\min})$ to node $i_{\min}$. Therefore, step 3 takes time $O(|N|) + O(|L|) = O(|L|)$.

Hence, one iteration of steps 1-4 of the algorithm requires $O(|L|)$ time and, since at most $|N|$ such iterations may occur, the worst case running time of Algorithm 1 is $O(|L||N|)$. Note that, in practice, the running time of the algorithm may be much smaller than that of the worst case, since more than two forest trees may be merged to a new tree at every iteration. Moreover, code optimization can also be made, so that only relevant links are checked in steps 1-2.

C. Broadcasting using a Minimum Spanning Tree

A minimum spanning tree of G is a spanning tree whose sum of link costs is minimal. The problem of finding an MST has been studied extensively in the literature and polynomial-time centralized and distributed algorithms exist for its solution (see for example [8], [24]). In this subsection, we provide a simple relation between the minimum-energy broadcast problem and the minimum spanning tree, which shows that an MST can also be used for broadcasting by all nodes in sparse networks.

Let $\widehat{T}$ be an MST, that is, a spanning tree that minimizes the cost $C(T) = \sum_{l \in L^T} c_l$. Given a source node s, $\widehat{T}_s$ is the s-rooted directed spanning tree induced by $\widehat{T}$. Let also T_s be any s-rooted directed spanning tree. The following lemma shows that if we use the tree $\widehat{T}_s$ for broadcasting from source node s, then the total power consumption is at most Δ times the total power consumed when the tree T_s is used, where Δ is the maximum node degree in the network.

Lemma 3: *It holds* $P(\widehat{T}_s) \leq \Delta P(T_s)$.

Since $P(\widehat{T}_s)$ is at most Δ times the power $P(T_s)$, where T_s is any s-rooted directed spanning tree, it follows that $P(\widehat{T}_s)$ is also at most Δ times the total power consumed when an optimal (minimum-energy) s-rooted directed spanning tree is used. Hence, we have the following corollary:

Corollary 3: *For any source node s, the total power consumed for broadcasting using a minimum spanning tree, is at most Δ times the optimal power, where Δ is the maximum node degree in the network.*

We note that the proof of Lemma 3 can be used to show that the above relation between the minimum-energy broadcast problem and the minimum spanning tree is also valid when G is a strongly connected *directed* graph. In this case, Δ is the maximum node outdegree in the network and the minimum spanning tree depends on the source node.

D. Issues of Distributed Implementation

Algorithm 1 assumes knowledge of network topology. Hence, it can be used in networks with infrequent topological changes and low mobility [3]. In general, Algorithm 1 can be applied in network environments where at least partial information of network topology is proactively maintained at each node, as in Optimized Link State Routing (OLSR) protocol [25]. Regarding its distributed implementation, we note that our algorithm has similarities with Kruskal's algorithm for determining a minimum spanning tree in a connected undirected graph [8], which can also be implemented in a distributed fashion [24]. Kruskal's algorithm builds a minimum spanning tree by adding one link at a time. At every iteration of the algorithm, a forest of trees is maintained, as in Algorithm 1, and a link of minimal cost is selected to join two forest trees, so that no cycle is created with previously selected links. The main difference between our algorithm and the distributed algorithm for MSTs in [24] is the manner by which the forest trees are joined, which in our case is more complicated. The issue of detailed distributed implementation and analysis of our algorithm is beyond the scope of the current work and it is a subject for further study.

V. Numerical Results

In this section we compare numerically the performance of the following three algorithms for various networks with different sizes: 1) Broadcast Incremental Power algorithm [4] ("BIP" algorithm for short), 2) our Algorithm 1 which constructs a single broadcast tree ("SBT" algorithm), and 3) the algorithm for determining a minimum spanning tree ("MST" algorithm). We choose BIP as the main algorithm for comparison, because it was one of the first algorithms that exploit the node-based nature of wireless networks and it was used by many researchers to evaluate numerically the performance of other heuristic algorithms for the minimum-energy broadcast problem. We note again that BIP determines a different broadcast tree

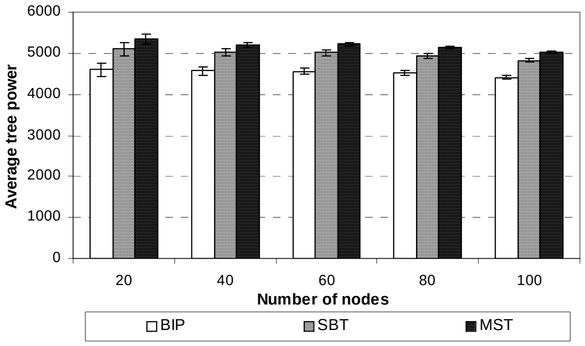

Fig. 7. Average tree power (over all possible source nodes) for various network sizes; $a = 2$, complete networks.

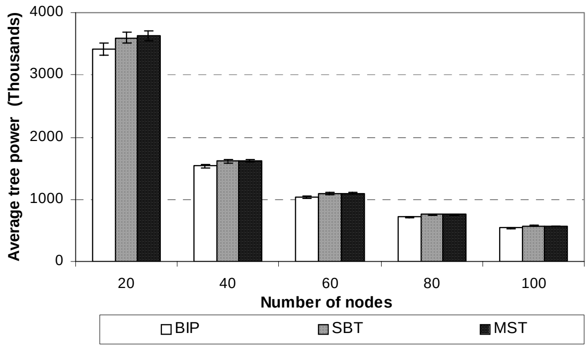

Fig. 8. Average tree power (over all possible source nodes) for various network sizes; $a = 4$, complete networks.

for every possible source node, while SBT algorithm constructs a single tree used by all nodes for broadcasting. Moreover, BIP improves its performance by using what is called the "sweep" operation, which detects redundant transmissions as well as transmissions whose power can be reduced.

The figures that follow represent the averages of the results obtained from 100 randomly generated network instances for each network size considered. We generate random networks with a specified number of nodes (20,40,...,100) as follows. We fix a rectangular grid of 100×100 points. A number of these points is randomly selected with uniform probability to represent the network nodes. The power needed for successful transmission over link (i, j) depends on the distance $d_{(i,j)}$ between the two nodes and it is given by $c_{(i,j)} = d^a_{(i,j)}$, where a is the propagation loss exponent.

The main performance metric of interest is the total power consumed for broadcasting initiated by different source nodes. In order to quantify this metric, we introduce a closely related measure which provides the average total power consumption for broadcasting initiated by any source node. That is, for a given network size, $|N|$, and for each individual network instance, we define the *average tree power* of algorithm X as $P^X = \frac{\sum_{s \in N} P(T^X_s)}{|N|}$, where T^X_s is the s-rooted directed spanning tree returned by algorithm X and $P(T^X_s)$ is the total power consumed for broadcasting from source node s. BIP algorithm constructs a different tree T^{BIP}_s for every possible source node $s \in N$, while the trees T^{SBT}_s are induced by the unique broadcast tree returned by algorithm SBT as described in Section III-A; a single tree is also returned by algorithm MST.

Fig. 7 shows the average tree power of the algorithms for various network sizes, when the propagation loss exponent a is 2 and there is no constraint on the maximum transmission power; that is, we assume that each node can successfully transmit information to all other network nodes. The symbols $\overline{\perp}$ on top of each bar represent the standard deviation of tree powers $\{P(T_s^X)\}_{seN}$. We observe that SBT algorithm provides fairly satisfactory performance, comparable to that of BIP, for all network sizes considered. The average tree power of SBT is 10.9% higher than that of BIP for $|N| = 20$, while this percentage falls to 9.1% for $|N| = 100$. The corresponding percentages for MST algorithm are 16.4% and 14%. The decrease in standard deviations for all algorithms as the network size increases, is due to the fact that the trees returned by the algorithms use shorter links (links with smaller costs) as the number of nodes increases, since the density of the nodes within the same geographical area increases as well; hence, the variations in tree powers for different source nodes are smaller for larger networks. Fig. 8 provides similar results for $a = 4$. In this case, the average tree power of SBT is 5.2% higher than that of BIP for $|N| = 20$, and 6.2% for $|N| = 100$. The corresponding percentages for MST algorithm are 6.2% and 5.9%. We observe that the difference in performance of the algorithms decreases as the propagation loss exponent increases. This observation conforms to results of previous works [13]. The main reason for this behavior is that the "penalty" of using longer links increases for larger values of a. Hence, the use of such links is avoided by all algorithms and the trees returned by BIP and SBT converge to MST when a increases.

The results presented thus far, show that our SBT algorithm performs fairly well for networks represented by unit disk graphs. We will now provide some interesting instances of general networks for which SBT outperforms significantly the other two algorithms. The simulations that follow attempt to model the following physical environment. Assume that the nodes are deployed on a terrain where there are various obstacles that may prohibit direct communication of certain nodes. Assume also that some of the nodes are located high up (on top of hills, buildings, etc.) so that the communication channel between these nodes and the rest of network nodes is less hostile, having smaller attenuation factor. We would like to evaluate the performance of the algorithms in such an environment.

The experiments performed in this case are the following. We set $a = 2$ and assume that the grid of 100×100 points is on the xy-plane of the 3-dimensional space. Nodes are placed on the grid as before

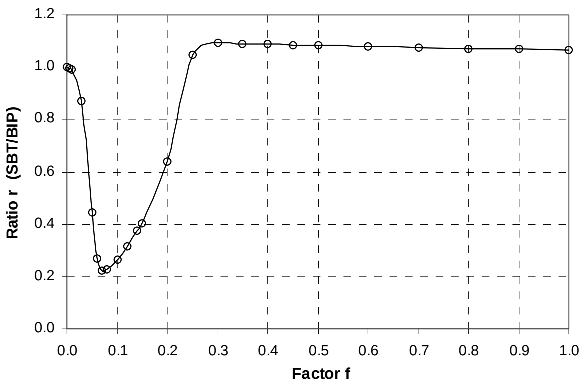

Fig. 9. Ratio of avg. tree power of SBT to that of BIP for different values of factor f; $a = 2$, 100-node sparse networks + 1 "special" node.

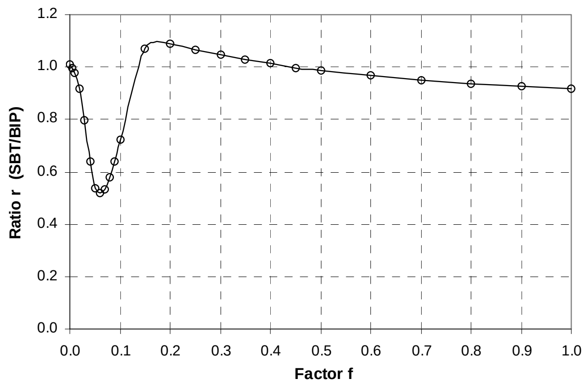

Fig. 10. Ratio of avg. tree power of SBT to that of BIP for different values of factor f; $a = 2$, 100-node sparse networks + 4 "special" nodes.

but, for each network instance, we include only links whose power is less than $c_{\max}$, defined as the smallest value that guarantees network connectivity. Hence, a link l between two nodes on the grid belongs to the set L of network links if $c_l \leq c_{\max}$. This constraint results in sparsely connected networks on the grid (however, we note that the results presented below are not sensitive to the choice of $c_{\max}$; similar behavior is observed even if $c_{\max}$ is chosen infinite, i.e., when the grid network is densely connected). After the network on the grid is constructed, we add a "special" node in the middle of the grid and in height $h = 50$, that is, the coordinates of this node are $(50, 50, 50)$. We assume that the constraint on the maximum transmission power does not hold for this additional node; hence, there is a link between this node and every other node on the grid. The power of such a link is $f \cdot d^2$, where f is a factor such that $0 < f \leq 1$, and d is the distance in the 3-dimensional space between the "special" node and a node on the grid. An alternative network example is to split the grid to four quarters and add 4 "special" nodes with coordinates $(25, 25, 50)$, $(25, 75, 50)$, $(75, 25, 50)$, $(75, 75, 50)$. Each one of these nodes is able to communicate only with nodes on the corresponding quarter of the grid.

Figures 9 and 10 present the ratio r of average tree power of SBT to that of BIP for different values of factor f, when 1 or 4 "special" nodes, respectively, are added to the 100-node sparsely connected networks. We observe that in both cases there is a range of values of f for which SBT significantly outperforms BIP. In Fig. 9, the best performance of SBT is achieved for $f = 0.07$ (ratio $r = 0.225$), while in Fig. 10 the corresponding values are $f = 0.06$ and $r = 0.517$. The behavior of the curves in these figures is explained

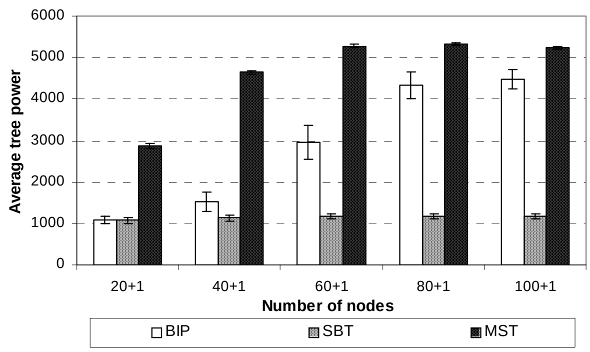

Fig. 11. Average tree power (over all possible source nodes) for various network sizes; $a = 2$, 1 "special" node added to the sparse networks, factor $f = 0.1$.

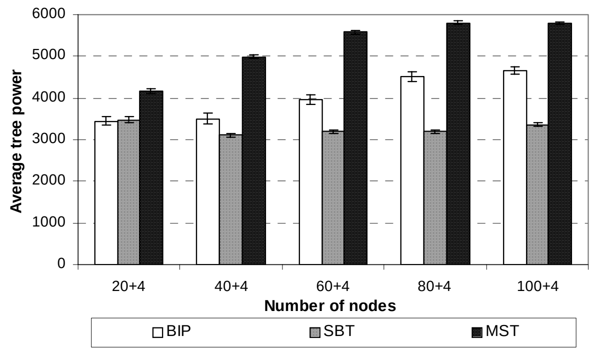

Fig. 12. Average tree power (over all possible source nodes) for various network sizes; $a = 2$, 4 "special" nodes added to the sparse networks, factor $f = 0.1$.

as follows. When f is very small (0.001 to 0.02) the costs of links between the "special" nodes and nodes on the grid are also very small, compared to the costs of links between nodes on the grid only; hence, both algorithms select the former links and construct almost identical trees (r is close to 1). The algorithms also behave almost identically (r is also close to 1) when f is large (larger than 0.25 in Fig. 9 and 0.15 in Fig. 10); in this case, both algorithms avoid using links of the "special" nodes, since their costs are larger than the costs of links between nodes on the grid. Among these two cases (very small or large values of f) there is a range of values for which, although it is more cost efficient to use links of the "special" nodes, BIP algorithm does not succeed in selecting these links. On the other hand, SBT algorithm exploits the cost function "incremental power consumed per joined tree" and succeeds in selecting the links of "special" nodes. Although SBT outperforms significantly BIP for a range of values of f in both figures, the difference in performance is greater in Fig. 9. This is due to the fact that in this case there is only 1 "special" node which communicates with all nodes on the grid, while in Fig. 10 there are 4 additional nodes and each one of them communicates only with nodes in its corresponding quarter. Hence, the gain that SBT achieves is higher in the first case, since the cost of the constructed trees is much smaller.

The fact that SBT achieves higher gain when there is 1 rather than 4 "special" nodes can also be observed in Figures 11 and 12, which both provide the average tree power of the algorithms for various network sizes when the factor f is 0.1. Fig. 11 corresponds to network instances with 1 "special" node added, while Fig. 12 presents the results when there are 4 additional nodes. We can see for example that when there are 40

nodes on the grid, the average tree power of BIP is 35.3% higher than that of SBT in Fig. 11, while this percentage falls to 12.8% in Fig. 12. The corresponding percentages when there are 80 nodes on the grid are 270.7% and 41.4%. Hence, it is concluded again, for the reason explained earlier, that the gain achieved by SBT is higher when there is 1 rather than 4 "special" nodes added to the network. Regarding the MST algorithm, it performs considerably worse in both figures for all the network sizes considered.

VI. Conclusions - Issues for Further Study

In this paper we addressed the minimum-energy broadcast problem in wireless networks, so that all broadcast requests initiated by different source nodes take place on the same broadcast tree. The main contribution is that we do not have to determine a different broadcast tree every time a source node initiates a broadcast request. Moreover, the provided results are valid for general networks and do not rely on unit disk graph models and geometric properties of the Euclidean space. We first showed that using the same broadcast tree does not result in widely varying total powers for different sources. We next developed a polynomial-time approximation algorithm to construct a single broadcast tree and analyzed its performance. We also provided a useful relation between the minimum-energy broadcast problem and the minimum spanning tree, and evaluated numerically with simulations the performance of our algorithm.

There are some interesting issues for further study that arise from our work. In this paper, we considered general undirected graphs to model the wireless network; however, the development of an appropriate algorithm in case of directed networks with asymmetric power requirements (different costs between two opposite directed links) remains an open problem. The distributed implementation of our algorithm could also be desirable in network environments with high mobility and frequent topological changes. Another important issue is the construction of a unique multicast tree, in case where only a subset of the nodes in the network need to communicate in an energy-efficient way. A trivial solution in this case would be to employ the broadcasting algorithm presented in this paper and then prune the resulting tree, so that only the multicast group nodes are located at the leaves of the tree. However, it might be possible to provide better solutions by looking directly at the multicast problem. In any case, we note that the adaptation of the proof of Lemma

1 for the multicast problem is straightforward and, hence, Corollary 1 also holds for the multicasting case. Finally, addressing our model in an energy-, and resource-limited environment where the maximization of network lifetime is the primary objective, is also an interesting subject which requires additional parameters, such as the initial battery energy of each node, to be considered and incorporated into the model.

REFERENCES

[1] I.F. Akyildiz *et al.*, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, March 2002, pp. 393-422.

[2] A.J. Goldsmith and S.B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wirel. Commun.*, vol. 9, no. 4, Aug. 2002, pp. 8-27.

[3] A. Ephremides, "Ad hoc networks: not an ad hoc field anymore," *Wirel. Commun. Mob. Comput.*, vol. 2, no. 5, Aug. 2002, pp. 441-448.

[4] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *MONET*, vol. 7, no. 6, Dec. 2002, pp. 481-492. A preliminary version of this paper appeared *in Proc. IEEE INFOCOM*, March 2000, pp. 585-594.

[5] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, Apr. 1998, pp. 374-387.

[6] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "Distributed heuristics for connected dominating sets in wireless ad hoc networks," *J. of Commun. and Networks*, vol. 4, no. 1, March 2002, pp. 22-29.

[7] J. Wu, B. Wu, and I. Stojmenovic, "Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets," *Wirel. Commun. Mob. Comput.*, vol. 3, no. 4, June. 2003, pp. 425-438.

[8] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993.

[9] A.E.F. Clementi *et al.*, "On the complexity of computing minimum energy consumption broadcast subgraphs," *in Proc. Symp. on Theor. Aspects of Comp. Science*, Feb. 2001, pp. 121-131.

[10] F. Li and I. Nikolaidis, "On minimum-energy broadcasting in all-wireless networks," *in Proc. IEEE Local Computer Networks*, Nov. 2001, pp. 193-202.

[11] A. Ahluwalia, E. Modiano, and L. Shu, "On the complexity and distributed construction of energy-efficient broadcast trees in static ad hoc wireless networks," *in Proc. Conf. on Inform. Sciences and Systems*, March 2002.

[12] W. Liang, "Constructing minimum-energy broadcast trees in wireless ad hoc networks," *in Proc. ACM MOBIHOC*, June 2002, pp. 112-122.

[13] M. Čagalj, J.-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," *in Proc. ACM MOBICOM*, Sep. 2002.

[14] P.-J. Wan *et al.*, "Minimum-energy broadcasting in static ad hoc wireless networks," *Wirel. Networks*, vol. 8, no. 6, Nov. 2002, pp. 607-617.

[15] A.K. Das *et al.*, "Minimum power broadcast trees for wireless networks: integer programming formulations," *in Proc. IEEE INFOCOM*, March-Apr. 2003.

[16] X.-Y. Li, "Algorithmic, geometric and graph issues in wireless networks," *Wirel. Commun. Mob. Comput.*, vol. 3, no. 2, March. 2003, pp. 119-140.

[17] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos, "A logarithmic approximation algorithm for the minimum energy consumption broadcast subgraph problem," *Inform. Proc. Letters*, vol. 86, no. 3, May 2003, pp. 149-154.

[18] G. Călinescu *et al.*, "Network lifetime and power assignment in ad hoc wireless networks," *in Proc. European Symp. on Algorithms*, Sep. 2003, pp. 114-126.

[19] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "Resource management in energy-limited, bandwidth-limited, transceiver-limited wireless networks for session-based multicasting," *Computer Networks*, vol. 39, no. 2, June 2002, pp. 113-131.

[20] I. Papadimitriou and L. Georgiadis, "Energy-aware broadcasting in wireless networks," *in Proc. WiOpt*, March 2003, pp. 267-277.

[21] V.V. Vazirani, *Approximation Algorithms*, Springer-Verlag, 2001.

[22] U. Feige, "A threshold of $\ln n$ for approximating set cover," *J. of ACM*, vol. 45, no. 4, July 1998, pp. 634-652.

[23] P.A. Humblet, "A distributed algorithm for minimum weight directed spanning trees," *IEEE Trans. Commun.*, vol. 31, no. 6, June 1983, pp. 756-762.

[24] R.G. Gallager, P.A. Humblet, and P.M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Trans. on Progr. Lang. and Systems*, vol. 5, no. 1, Jan. 1983, pp. 66-77.

[25] T. Clausen and P. Jacquet, "Optimized link state routing protocol," *IETF Internet Draft*, draft-ietf-manet-olsr-11.txt, July 2003 (Work in progress).

APPENDIX

A. *Proof of Lemma 1*

Consider a node $i \in N$. Note that due to Definition 1 of node "power", if all links in $\mathcal{A}_s(i)$ are included in $\mathcal{A}(i)$, then $p_i^{\mathcal{A}_s} \leq p_i^{\mathcal{A}}$. The latter statement and the fact that $\cup_{i\in N}\mathcal{A}(i) = L^T$ imply that if $p_i^{\mathcal{A}_s} > p_i^{\mathcal{A}}$, then there is at least one link l' in the set $\mathcal{A}_s(i)$ with cost $c_{l'} = p_i^{\mathcal{A}_s}$, which is assigned to a neighbor node $j \in N^T(i)$ under link assignment $\mathcal{A}$. Since l' may not be the only link assigned to node j under assignment $\mathcal{A}$, using again the definition of node "power", we conclude that in this case it holds $p_i^{\mathcal{A}_s} = c_{l'} \leq p_j^{\mathcal{A}}$. Hence, in general we can write $p_i^{\mathcal{A}_s} \leq p_i^{\mathcal{A}} + p_{j_i}^{\mathcal{A}}$, where $p_{j_i}^{\mathcal{A}} = 0$ for a node i for which $\mathcal{A}_s(i) = \varnothing$, while for a node i for which $\mathcal{A}_s(i) \neq \varnothing$, $(i, j_i) \in \mathcal{A}_s(i)$ and j_i is the neighbor of i in T whose "power" is maximal under link assignment $\mathcal{A}$ among any other node j such that $(i, j) \in \mathcal{A}_s(i)$. Therefore,

$$\sum_{i\in N} p_i^{\mathcal{A}_s} \leq \sum_{i\in N} p_i^{\mathcal{A}} + \sum_{i\in N} p_{j_i}^{\mathcal{A}}. \tag{4}$$

Recall now that $\mathcal{A}_s$ corresponds to broadcasting from a given source node s using tree T. Since $(i, j_i) \in \mathcal{A}_s(i)$ for a node i for which $\mathcal{A}_s(i) \neq \varnothing$, the set of links $L^T(j_i) - \{(i, j_i)\}$ is assigned to node j_i under link assignment $\mathcal{A}_s$. That is, there can be no other node i' such that $(i', j_i) \in \mathcal{A}_s(i')$. Therefore, it holds $j_i \neq j_{i'}$ for any two nodes $i \neq i'$ for which $\mathcal{A}_s(i) \neq \varnothing$ and $\mathcal{A}_s(i') \neq \varnothing$. From the latter statement and the fact that

$p_{j_i}^{\mathcal{A}} = 0$ for a node i for which $\mathcal{A}_s(i) = \varnothing$, it is concluded that all terms in $\sum_{i \in N} p_{j_i}^{\mathcal{A}}$ are also included in $\sum_{i \in N} p_i^{\mathcal{A}}$ (zero terms do not contribute to a sum in any case). Hence, $\sum_{i \in N} p_{j_i}^{\mathcal{A}} \leq \sum_{i \in N} p_i^{\mathcal{A}}$ and inequality (4) gives $\sum_{i \in N} p_i^{\mathcal{A}_s} \leq 2 \sum_{i \in N} p_i^{\mathcal{A}} \Rightarrow P^{\mathcal{A}_s}(T) \leq 2P^{\mathcal{A}}(T)$. ■

B. Proof of Lemma 2

Let b_k be the number of forest trees at the beginning of k^{th} iteration. Therefore, we have $b_1 = n$. If the algorithm takes K iterations to complete, then we define $b_{K+1} = 1$. It follows that at k^{th} iteration, the number of links that join node $i_{\min}$ with the trees in the set $\mathbb{T}_{i_{\min}}(l_{\min})$ at step 3 of Algorithm 1 is $b_k - b_{k+1}$ (note that $b_k - b_{k+1}$ is equal to the cardinality of set $\mathbb{T}_{i_{\min}}(l_{\min})$). Let q_k be the extra power needed by node $i_{\min}$ to reach the $b_k - b_{k+1}$ forest trees at the k^{th} iteration. We will show that

$$q_k \leq \frac{b_k - b_{k+1}}{b_k - 1} P^{\mathcal{A}^*}(T^*). \tag{5}$$

The above inequality implies the lemma. To see this, sum (5) over all K iterations to obtain

$$\sum_{k=1}^{K} q_k \leq \sum_{k=1}^{K} \frac{b_k - b_{k+1}}{b_k - 1} \cdot P^{\mathcal{A}^*}(T^*). \tag{6}$$

Note that for a certain node $i \in N$, the "power" of i under assignment $\overline{\mathcal{A}}$ (see Definition 1), $p_i^{\overline{\mathcal{A}}}$, is equal to the sum over all K iterations of the powers q_k that correspond to that node i. That is,

$$p_i^{\overline{\mathcal{A}}} = \sum_{k=1}^{K} (q_k \cdot 1(\text{power } q_k \text{ corresponds to node } i)), \tag{7}$$

where the indicator function is included to denote whether each one of the powers q_k, $1 \leq k \leq K$, corresponds to node i. From the definition of the cost of tree $\overline{T}$ under assignment $\overline{\mathcal{A}}$ (see Definition 1) and equality (7), it follows that

$$P^{\overline{\mathcal{A}}}(\overline{T}) = \sum_{i \in N} \sum_{k=1}^{K} (q_k \cdot 1(\text{power } q_k \text{ corresponds to node } i)) = \sum_{k=1}^{K} q_k. \tag{8}$$

Observe also that

$$\frac{b_k - b_{k+1}}{b_k - 1} = \overbrace{\frac{1}{b_k - 1} + \frac{1}{b_k - 1} + ... + \frac{1}{b_k - 1}}^{b_k - b_{k+1} \text{ terms}} \leq \frac{1}{b_k - 1} + \frac{1}{b_k - 2} + ... + \frac{1}{b_{k+1}}.$$

Since $b_1 = n$ and $b_{K+1} = 1$, we have from the above inequality that

$$\sum_{k=1}^{K} \frac{b_k - b_{k+1}}{b_k - 1} \leq \sum_{k=1}^{n-1} \frac{1}{k} = H(n-1). \tag{9}$$

From (6), (8), and (9), the lemma is concluded. Let us now prove (5). The tree T^* is a spanning tree and, hence, it spans all nodes in G. This implies that it also joins the b_k forest trees at the beginning of k^{th} iteration with at least $b_k - 1$ links. Each of these links is assigned according to $\mathcal{A}^*$ to exactly one node. Let U be the set of nodes in T^* to which these links are assigned. For a node $i \in U$, let l' be the link with largest cost among the aforementioned links that have been assigned to it. Let also $n_i(l')$ be the number of distinct forest trees (other than the tree to which node i belongs) that can be reached by i when power $c_{l'}$ is used. Since T^* joins the b_k forest trees with at least $b_k - 1$ links, it holds

$$\sum_{i \in U} n_i(l') \geq b_k - 1. \tag{10}$$

By the definition of the quantities $a_i(l)$ at the k^{th} iteration of Algorithm 1, we have

$$\min_{l \in L'(i)} \{a_i(l)\} \leq \frac{c_{l'} - p_i^{\mathcal{A}_F}}{n_i(l')} \leq \frac{p_i^{\mathcal{A}^*}}{n_i(l')}, \tag{11}$$

where the second inequality in (11) is due to the fact that $p_i^{\mathcal{A}_F} \geq 0$ and that the link l' may not have the largest cost among all links that are eventually assigned to node i according to $\mathcal{A}^*$. From (10) and (11) it follows that

$$\sum_{i \in U} \frac{p_i^{\mathcal{A}^*}}{\min_{l \in L'(i)} \{a_i(l)\}} \geq b_k - 1. \tag{12}$$

Since $P^{\mathcal{A}^*}(T^*)$ is the sum of "powers" $p_i^{\mathcal{A}^*}$, $i \in N$, and $U \subseteq N$, it holds

$$P^{\mathcal{A}^*}(T^*) \geq \sum_{i \in U} p_i^{\mathcal{A}^*} \Rightarrow \frac{P^{\mathcal{A}^*}(T^*)}{a_{\min}} \geq \sum_{i \in U} \frac{p_i^{\mathcal{A}^*}}{a_{\min}}. \tag{13}$$

Since $a_{\min}$ is the minimum of quantities $a_i(l)$, $i \in N$ such that $L'(i) \neq \varnothing$, $l \in L'(i)$, it follows from (12) and (13) that

$$\frac{P^{\mathcal{A}^*}(T^*)}{a_{\min}} \geq b_k - 1. \tag{14}$$

By the definition of $a_{\min}$ at the k^{th} iteration, we have

$$a_{\min} = \frac{q_k}{b_k - b_{k+1}}. \tag{15}$$

Combining (14) and (15), inequality (5) is concluded. ■

C. *Proof of Lemma 3*

Note that for a node i in the tree T_s, it holds

$$\max_{l \in L_{out}^{T_s}(i)} \{c_l\} \leq \sum_{l \in L_{out}^{T_s}(i)} c_l \leq \Delta \max_{l \in L_{out}^{T_s}(i)} \{c_l\}. \tag{16}$$

Using the first of the above inequalities, we have

$$P(\widehat{T}_s) = \sum_{i \in N} p_i^{\widehat{T}_s} = \sum_{i \in N} \max_{l \in L_{out}^{\widehat{T}_s}(i)} \{c_l\} \leq \sum_{i \in N} \sum_{l \in L_{out}^{\widehat{T}_s}(i)} c_l = \sum_{l \in L^{\widehat{T}_s}} c_l = C(\widehat{T}_s). \tag{17}$$

Since $\widehat{T}_s$ is induced by $\widehat{T}$, which is an MST, it follows from (17) and the second inequality in (16) that

$$P(\widehat{T}_s) \leq C(T_s) = \sum_{l \in L^{T_s}} c_l = \sum_{i \in N} \sum_{l \in L_{out}^{T_s}(i)} c_l \leq \sum_{i \in N} \left(\Delta \max_{l \in L_{out}^{T_s}(i)} \{c_l\} \right) = \Delta \sum_{i \in N} p_i^{T_s} = \Delta P(T_s). \qquad \blacksquare$$