Energy-efficient Broadcasting in All-wireless Networks [*]

Mario Čagalj[†]   Jean-Pierre Hubaux
Laboratory for Computer Communications and Applications (LCA)
Swiss Federal Institute of Technology Lausanne (EPFL)
EPFL-I&C-LCA, CH-1015 Lausanne, Switzerland
mario.cagalj@epfl.ch, jean-pierre.hubaux@epfl.ch

Christian C. Enz
Swiss Center for Electronics and Microtechnology (CSEM)
CH-2007 Neuchtel Switzerland
christian.enz@csem.ch

April 17, 2003

Abstract

In all-wireless networks, minimizing energy consumption is crucial as in most cases the nodes are battery-operated. We focus on the problem of power-optimal broadcast, for which it is well known that the broadcast nature of radio transmissions can be exploited to optimize energy consumption. This problem appears to be difficult to solve [31]. We provide a formal proof of NP-completeness for the general case and give an NP-completeness result for the geometric case; in the former, the network topology is represented by a generic graph with arbitrary weights, whereas in the latter a Euclidean distance is considered. For the general case, we show that it cannot be approximated better than $O(\log N)$, where N is the total number of nodes. We then describe an approximation algorithm that achieves the $O(\log N)$ approximation ratio. We also describe a new heuristic, Embedded Wireless Multicast Advantage. We show that it compares well with other proposals and we explain how it can be distributed.

Keywords: wireless ad hoc networks, minimum-energy networks, energy efficiency, approximation algorithms, complexity theory

1   Introduction

In recent years, all-wireless networks have attracted significant attention due to their potential applications in civil and military domains [21, 22, 4, 11]. An all-wireless network consists of numerous devices (nodes) that are equipped with processing, memory and wireless communication capabilities, and are linked via short-range ad hoc radio connections. This kind of network has no pre-installed infrastructure, but communication between the network nodes is supported by multi-hop transmissions. In these networks, each node has a limited energy resource (battery) and operates unattended; consequently, energy efficiency is an important design consideration [26, 32].

In this paper, we focus on the source-initiated broadcasting of data in static all-wireless networks. Data are distributed from a source node to each node in a network. Our main objective is to construct a *minimum-energy broadcast tree* rooted at the source node. Nodes belonging to a broadcast tree can be divided into two categories: *relay* nodes and *leaf* nodes. The relay nodes are those that relay data by forwarding it to other nodes (relaying or leaf), and leaf nodes only receive data. Each node can transmit at different power levels and thus reach a different number of neighboring nodes. Given the source node r, we want to find a set of relaying nodes and their respective transmission levels so that all nodes in the network receive a message sent by r, whereby the total energy expenditure for this task is minimized. We call this broadcasting problem the *minimum-energy broadcast* problem.

We base our work on the so called *node-based* multicast model [31]. In this model there is a trade-off between reaching more nodes in a single hop thus using more energy and reaching fewer nodes using less energy. This trade-off is made possible by the *broadcast nature* of the wireless channel.

In Section 2, we overview related work concerning the minimum-energy broadcast problem. In Section 3, we discuss the system model used. In Section 4, we prove that the minimum-energy broadcast problem is NP-complete and show that it cannot be approximated better than $O(\log N)$ for a general graph, where N is the number of nodes in a network; we also give the NP-completeness result for the geometric version of the minimum-energy broadcast problem. Then, in Section 5, we present $O(\log N)$-approximation algorithms for the general graph version and a heuristic algorithm that is easy to distribute. Performance evaluation results are presented in Section 6. Finally in Section 7 we conclude.

2   Related work

Minimizing the energy consumption of all-wireless networks has received significant attention over the last few years [1, 24, 9, 14, 33, 15, 25, 12]. We were inspired by the exciting results related to the problem of minimum-energy broadcasting in all-wireless networks [26, 30, 5, 13, 19, 20], and in particular by the work of Wieselthier et al. [31, 32]. In this work they introduce the node-based multicast model for wireless networks upon which they have built several broadcast and multicast heuristics. One of the most notable contributions of their work is the Broadcast Incremental Power (BIP) algorithm. The main objective of BIP is to construct a minimum-energy broadcast tree rooted at the source node. It constructs the tree by first determining the node that the source can reach with a minimum expenditure of power. BIP constructs a tree that initially contains a single node; it then determines which uncovered node can be added to the tree at a *minimum additional* cost. At each iteration of BIP, the nodes that have already covered some node can further increase their transmission power to reach some other yet uncovered node. BIP is similar to Prim's algorithm [3] for the formation of minimum spanning trees, but with the difference that weights, with BIP, are dynamically updated at each step.

In [30] Wan et al. provide the first analytical results for the minimum energy broadcast problem. By exploring geometric structures of an Euclidean minimum spanning tree (MST), they prove that the approximation ratio of MST is between 6 and 12, and that the approximation ratio of BIP is between $\frac{13}{6}$ and 12. They also found that for some instances, BIP fails to use the broadcast nature of the wireless channel. This happens because BIP adds only one node at each iteration, the one that can be added at a minimum additional cost. Thus BIP, although centralized, does not use all the available information about the network. As a result, it may construct a broadcast tree that coincides with the shortest path tree of a network graph, where the broadcast nature of the media is completely ignored. A possible approach to alleviate this problem is to add to the tree more than one node at each iteration,

and not necessarily at a minimum additional cost. But, in this case, there must be another criterion for the selection of nodes. Another difficulty with BIP is that distributing it is not obvious and according to the authors of BIP and of [30], the development of distributed algorithms is the major challenge for a minimum energy broadcast problem. However, Wan et al. [30] and Wieselthier et al. [31] do not really address this challenge. In Section 5.2 we present a heuristic algorithm that achieves the same approximation ration as BIP for the geometric case, and yet is easy to distribute.

Li et al., in another closely related work [13], also recognize the weaknesses of BIP and propose another centralized heuristic to tackle the broadcasting problem. However, they do not consider the issue of developing a distributed algorithm for a minimum energy broadcast. Li et al. [13] also sketch a proof of the NP-hardness of a general version of the minimum energy broadcast.

A proof of NP-hardness of the minimum energy broadcast problem in metric space has been proposed by Eğecioğlu et al. [5]. However, in their interpretation of the minimum energy broadcast problem, they restrict the transmission ranges of the nodes to a set of *integers*, which captures very few instances of the problem in metric space.

In [16] Liang provide a proof of NP-completeness of the minimum-energy broadcast problem, as well as an approximation algorithm for the problem in general setting, which delivers a feasible solution of cost $O(\log^3 n)$ times the optimum. The time complexity of the proposed algorithm is $O(kn^2 \log n)$, where k is the total number of power levels at each node. However, [16] does not provide an answer to the question: "Is there an approximation algorithm for the problem with a constant performance ratio?". We do provide an answer in Section 4 (Theorem 2).

Very recently, it was brought to our attention that more researchers are also studying the problem of minimum-energy broadcasting in all-wireless networks [2, 6, 17].

3   System model

We first provide a model of wireless communications. Then using it as a basis, we develop a graph model that will be used to assess the complexity of the minimum-energy broadcast problem and to develop an approximation algorithm.

In our model of a wireless network, nodes are stationary. In this paper, we assume a large availability of bandwidth resources, i.e. communication channels. We do so because we focus only on minimum energy broadcast communication and do not consider issues like contention for the channel or lack of bandwidth resources. We also assume that nodes in a network are equipped with omnidirectional antennas. Thus, due to the broadcast nature of wireless channels, all nodes that fall in the transmission range of a transmitting node can receive its transmission. This property of wireless media is called *Wireless Multicast Advantage*, which we refer to as WMA [31].

In this model, each node can choose to transmit at different power levels that do not exceed some maximum value $p_{\max}$. Let P denote the set of power levels at which a node can transmit. When a node i transmits at some power level $p \in P$, we assign it a weight equal to p, which we call a *node power*. The connectivity of the network depends on the transmission power. Node i is said to be *connected* to node j if node j falls in the transmission range of node i. This link is then assigned a *link cost* c_{ij} that is equal to the minimum power that is necessary to sustain link (i, j).

Next we define a graph model for wireless networks, which captures important properties of wireless media (including the wireless multicast advantage). An all-wireless network can be modeled by a directed graph $G = (V, E)$, where V represents the finite set of nodes and E the set of communication links between the nodes. Each edge (arc) $(i, j) \in E$ has link cost $c_{ij} \in \mathbb{R}_+$ assigned to it, and each node $i \in V$ is assigned a *variable node power* p_i^v. The variable node power takes a value from the

set P defined above. Initially, the variable node power assigned to a node is equal to zero and is set to value $p \in P$ if the node transmits at p. Let V_i denote the set of *neighbors* of node i. Node j is said to be a *neighbor* of node i if node j falls in the maximum transmission range of node i, which is determined by $p_{\max}$. All nodes $j \in V_i$ that satisfy $c_{ij} \leq p_i^v$ are said to be *covered* by node i. Thus, if node i transmits at power $p_{\max}$, all the nodes of V_i will be covered.

Now that we have the model, we study in detail the intrinsic complexity of the minimum-energy broadcast problem in the following section.

4   Complexity issues

In this section, we give an in-depth analysis of the complexity of the minimum-energy broadcast problem. Let us first briefly recall a few concepts from complexity theory [8]. The problems polynomially solvable by *deterministic* algorithms belong to the P class. Whereas, all the problems solvable by *nondeterministic* algorithms belong to the NP class. It can easily be shown that P $\subseteq$ NP. Also, there is widespread belief that P $\neq$ NP. The theory of complexity is focused on *decision problems*, i.e., problems that have either *yes* or *no* as an answer. Notice that each optimization problem can be easily stated as the corresponding decision problem. Informally, a decision problem Π is said to be NP-complete if $\Pi \in$ NP and for all other problems $\Pi^{'} \in$ NP, there exists a polynomial transformation from $\Pi^{'}$ to Π (we write $\Pi^{'} \propto \Pi$) [8]. There are two important properties of the NP-complete class. If any NP-complete problem could be solved in polynomial time, then all problems in NP could also be solved in polynomial time. If any problem in NP is intractable[1], then so are all NP-complete problems. Presently, there is a large collection of problems considered to be intractable.

In this section, we consider the problem of minimum-energy broadcast in two different graph models, specifically a general graph and a graph in Euclidean metric space. In general graphs, links are arbitrarily distributed and have weights arbitrarily chosen from the set P. This graph model is well suited for modeling wireless networks in indoor environments. Whereas, for graphs in Euclidean metric space, the existence and the weight of the link between two nodes depends exclusively on the distance between the nodes and their transmission levels. This graph model fits well for outdoor scenarios.

4.1   General graph version

In the following, we show that a general graph version of the minimum-energy broadcast problem is intractable, that is, it belongs to the NP-complete class. Because of its similarity to the well-known *Set Cover* problem [10] that aims at finding the minimum cost cover for a given set of nodes, we call it the *Minimum Broadcast Cover* and refer to it as MBC. We convert MBC into a decision problem in the following way:

MINIMUM BROADCAST COVER (MBC)

INSTANCE: A directed graph $G = (V, E)$, a set P consisting of all power levels at which a node can transmit, edge costs $c_{ij} : E(G) \to \mathbb{R}_+$, a source node $r \in V$, an assignment operation $p_i^v : V(G) \to P$ and some constant $B \in \mathbb{R}_+$.

QUESTION: Is there a node power assignment vector $A = [p_1^v \ p_2^v \ \ldots \ p_{|V|}^v]$ such that it induces the directed graph $G^{'} = (V, E^{'})$, where $E^{'} = \{(i, j) \in E : c_{ij} \leq p_i^v\}$, in which there is a path from r to any node of V (all nodes are covered), and such that $\sum_{i \in V} p_i^v \leq B$?

---

[1] We refer to a problem as intractable if no polynomial time algorithm can possibly solve it.

Notice that the above question is the equivalent of asking if there is a broadcast tree rooted at r with total cost B or less, and such that all nodes in V are included in the tree (covered).

We prove NP-completeness of MBC for a general graph by showing that a special case of MBC is NP-complete. In order to obtain this special case of MBC, we define the following restriction to be placed on the instances of MBC: All the links between any node i and its neighbors $j \in V_i$ have the same cost c. Consequently, the node i either does not transmit or it transmits with $p_i^v = c$. We call this special case SINGLE POWER MBC. We prove NP-completeness of the SINGLE POWER MBC problem by reduction from the SET COVER (SC) problem, which is well known to be NP-complete [8].

SET COVER (SC)

INSTANCE: A set I of m elements to be covered and a collection of sets $S_j \in I$, $j \in J = \{1, ..., n\}$. Weights w_j for each $j \in J$, and a constant $B \in \mathbb{R}^+$.

QUESTION: Is there a subcollection of sets C that form a cover, i.e., $\cup_{j \in C} S_j = I$ and such that $\sum_{j \in C} w_j \leq B$?

First we describe the construction of a graph G that represents any instance of the set cover problem. The graph G has a vertex set $I \cup \{v_1, v_2, ..., v_n\}$, that is, G consists of elements of I and set vertices v_j representing sets $S_j \in I$, $j \in J = \{1, ..., n\}$. There is an edge between an element $e \in I$ and a set node v_i if the set S_i contains the element. Each set node v_i is assigned the weight w_i of the set S_i the node represents. All other nodes and all edges are not weighted, that is, they have a weight of zero. Thus, $G = (V, E)$ is a bipartite graph, as is illustrated in Figure 1 (the graph on left).

The transformation from SC to SINGLE POWER MBC consists first in adding a source (root) node r to G and making it adjacent to all the set nodes v_j. Note that we use undirected edges here to emphasize that the links between the source r and nodes v_j are bidirectional. We proceed by assigning a zero weight to every edge the root node r shares with the set nodes v_j. Then, the edges between v_j and elements $e \in I$ are directed in order to capture the fact that no element $e \in I$ is ever selected into the cover set C. Finally, the directed edges the node v_j shares with elements $e \in I$ are assigned the weight w_j. The resulting graph, which we denote with $G_b = (V_b, E_b)$, is illustrated in Figure 1 (the graph on right). It is easy to see that the transformation can be done in polynomial time.
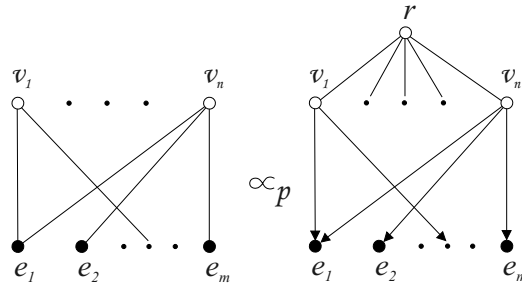

Figure 1: The reduction of SET COVER (left) to MINIMUM BROADCAST COVER (right)

Next, we prove the following theorem.

***Theorem* 1** *SINGLE POWER MBC is NP-complete.*

Proof:
The proof consists first in showing that SINGLE POWER MBC belongs to the NP class and then in showing that the above polynomial transformation (Figure 1) reduces SC to SINGLE POWER MBC.

It is easy to see that SINGLE POWER MBC belongs to the NP class since a nondeterministic algorithm needs only to guess a set of transmitting nodes ($p_i^v > 0$) and to check in polynomial time whether there is a path from the source node r to any node in a final solution and whether the cost of the final solution is $\leq$ B.

We continue the proof by showing that given the minimum broadcast cover C_b of G_b with cost $cost(C_b)$, the set $C_b - \{r\}$ always corresponds to the minimum set cover C of G of the same cost ($cost(C) = cost(C_b)$), and vice versa. Let C denote the minimum set cover of G. Let $cost(C) = \sum_{j \in C} w_j$ denote the cost of this cover. It is easy to see that all nodes of G_b can also be covered with total cost $cost(C)$. This can be achieved by having the source node r cover all the set nodes v_j, $j \in J = \{1, ..., n\}$ at zero cost and then by selecting among the covered nodes those corresponding to the nodes of G that satisfy $v_j \in C$ as new transmitting nodes, which we refer to as $C_b - \{r\}$. Hence the minimum broadcast cover of G_b is C_b with total cost $cost(C_b) = cost(C)$.

Conversely, suppose that we have the minimum broadcast cover C_b of G_b with total cost $cost(C_b)$. Thus the minimum set cover C of G must be $C = C_b - \{r\}$, i.e., $cost(C) = cost(C_b)$. We prove this by contradiction. Let us first assume that $cost(C) < cost(C_b)$ (hence $C \neq C_b - \{r\}$). In this case, with the same reasoning as before, G_b can be covered by $C_b' = C + \{r\}$ that satisfies $cost(C_b') < cost(C_b)$. This, however, contradicts the preceding assumption that C_b is the minimum broadcast cover of G_b. On the other hand, let us assume that $cost(C_b) < cost(C)$ (hence $C \neq C_b - \{r\}$). Since C_b covers all the elements $e \in I$, we can obtain a set cover C' for this instance as follows: $C' = C_b - \{r\}$. Now we have $cost(C') = cost(C_b) < cost(C)$, which contradicts the optimality of C and concludes the proof.

$\square$

Since the SINGLE POWER MBC problem is a special case of the MBC problem, and MBC belongs to the NP class, which can be shown along the similar lines as for THE SINGLE POWER MBC problem, we have the following corollary:

Corollary 1 *MINIMUM BROADCAST COVER (MBC) is NP-complete.*

Another important implication of Theorem 1 is the following theorem. Let N denote the total number of nodes in an instance of MBC.

Theorem 2 *There exists a constant $c > 0$ such that MINIMUM BROADCAST COVER (MBC) cannot be approximate better than $c \log N$, if $P \neq NP$.*

Proof: To prove this we recall that there exists a constant $c' > 0$ such that no polynomial-time approximation algorithm for SC achieves an approximation ratio smaller than $c' \log n$ if $P \neq NP$, where n is the total number of elements in an instance of SC [29]. We showed above how any instance of the SC problem can be transformed to the corresponding instance of MBC. Now, assume that we have an approximation algorithm for MBC with the performance guarantee better than $c' \log (N-1)$. By applying this algorithm to the instance of MBC obtained from the SC instance, we would get a solution with a cost lower than $c' \log(n + 1 - 1) \cdot OPT = c' \log n \cdot OPT$. Since this solution is also feasible to the instance of SC, this would mean that we can approximate SC better than $c' \log n$, which contradicts the fact that SC is hard to approximate better than $c' \log n$. We obtain the theorem by noting that $c' \log(N - 1) = c \log N$, where $0 < c \leq c'$ for $N > 2$. $\square$

Fortunately, Theorem 2 does not hold for all instances of the minimum-energy broadcast problem. By exploring the geometric structure of the minimum-energy broadcast problem, Wan et al. were able to show that the Euclidean minimum spanning tree approximates the minimum-energy broadcast

problem within a factor of 12 [30]. But, whether the geometric instances of the minimum-energy broadcast problem can be solved in polynomial time was left as an open question. We provide an answer in the next subsection.

4.2 Geometric version

In this section, we show that the minimum-energy broadcast problem in two-dimensional Euclidean metric space is intractable. In metric space, the distance between points (nodes) obeys triangle inequality, that is, $d_{ij} \leq d_{ik} + d_{kj}$, where d_{xy} is the Euclidean distance between nodes x and y. We have seen that given the graph version of the minimum-energy broadcast problem we can have arbitrary costs of links between nodes. This is because we did not have to worry about the distances between nodes and all links have been imposed by a given graph. On the contrary, in metric space, links and their respective costs are dictated by the distances between nodes and their transmission energies. The cost c_{ij} between two nodes i and j is given as

$$c_{ij} = k d_{ij}^{\alpha} \tag{1}$$

where $k \in \mathbb{R}^+$ is constant depending on the environment and α is a propagation loss exponent that takes values between 2 and 5 [23].

We refer to this instance of the minimum-energy broadcast problem as the *Geometric Minimum Broadcast Cover* (GMBC) problem. The decision problem related to GMBC can be formulated as follows:

GEOMETRIC MINIMUM BROADCAST COVER (GMBC)

INSTANCE: A set of nodes V in the plane, a set P consisting of all power levels at which a node can transmit, a constant $k \in \mathbb{R}_+$, costs of edges $c_{ij} = k d_{ij}^{\alpha}$ where d_{ij} is the Euclidean distance between i and j, a real constant $\alpha \in [2..4]$, a source node $r \in V$, an assignment operation $p_i^v : V(G) \rightarrow P$ and some constant $B \in \mathbb{R}_+$.

QUESTION: Is there a node power assignment vector $A = [p_1^v \ p_2^v \ \dots \ p_{|V|}^v]$ such that it induces the directed graph $G = (V, E)$, with an edge (arc) directed form node i to node j if and only if $c_{ij} \leq p_i^v$, in which there is a path from r to any node of V (all nodes are covered), and such that $\sum_{i \in V} p_i^v \leq B$? Given the above formal definition of the geometric version of the minimum-energy broadcast problem, we have the following theorem:

***Theorem* 3** *GEOMETRIC MINIMUM BROADCAST COVER (GMBC) is NP-complete.*

The proof of the theorem is in [28]. We proved NP-completeness of GMBC is done by reduction from the PLANAR 3-SAT problem, which is known to be NP-complete [18].

In the following section, we devise approximation algorithms that enable us to find good solutions to the minimum-energy broadcast problem.

5 Proposed algorithms

In this section, we first present an approximation algorithm that achieves $O(\log N)$ approximation ratio for any instance of MBC. Then, we elaborate on the algorithm EWMA, designed deliberately for the geometric version of the minimum energy broadcast problem, and we explain how to convert it to a distributed algorithm.

5.1 $O(\log N)$-approximation algorithm

The MBC problem can be seen as a special case of the *hitting set problem*. The hitting set problem is defined as follows [10]: Given subsets $S_1, \ldots, S_p$ of a ground set E and given a nonnegative cost c_e for every element $e \in E$, find a minimum-cost subset $A \subseteq E$ such that $A \cap S_i \neq \emptyset$ for every $i = 1, \ldots, p$ (i.e. A hits every S_i).

In our case, we are given a connected graph $G = (V, E)$ with positive edge costs and a special root node r. The sets to hit are all r directed cuts, i.e. the sets of edges of the form $\delta^-(S) = \{(i, j) \in E : i \notin S, j \in S\}$ where $S \subseteq V - \{r\}$. Informally, for any subset of nodes $S \subseteq V - \{r\}$, we should have at least one transmitter $i \notin S$ that covers at least one node $j \in S$. It is easy to see that, if this is fulfilled for all $S \subseteq V - \{r\}$, we obtain a feasible solution for MBC. Consequently, any set $S \subseteq V - \{r\}$ that has no edge incoming to it is said to be violated. For simplicity, we will say that S is not hit while meaning that $\delta^-(S)$ is not hit. The number of violated sets can be in theory as large as $2^{|V|-1}$, i.e. exponential in the total number of nodes. In order to drastically reduce this number, we apply the technique described in [10], where, instead of considering all possible violated sets, we take into account only *minimal violated sets*. Any violated S is said to be a minimal violated set if there exists no violated S' with $S' \subset S$. The rule we use to calculate minimal violated sets is defined by the following:

Definition 1 *The minimal violated set is a* strongly connected *component (i.e. collection of nodes)* $S \subseteq V - \{r\}$ *that contains no directed edge incoming to it.*

We next describe an approximation algorithm (Algorithm 1)that achieves an $O(\log N)$ approximation ratio for MBC. Let C denote the set comprising pairs (i, k) where $i \in V$ is *transmitter* and where $k \in P$ its respective *transmission power level*. The algorithm iteratively selects the most cost-effective pair (i, k) and puts it into the set C and updates correspondingly the collection of minimal violated sets $\mathcal{V}$, until $\mathcal{V}$ is empty (i.e. C is a feasible solution).

Algorithm 1 $O(\log N)$-approximation algorithm

---

1   $C \leftarrow \emptyset; t = 0$

2   While C is not feasible

3     $t \leftarrow t + 1; \quad \mathcal{V}(t) \leftarrow Violation(C)$

4     $(i, k)_t = \arg\min_{(i,k)} \frac{c_i^k(t) - \underline{c}_i(t)}{|S(t)|}$

5     $C \leftarrow (i, k)_t; \quad \underline{c}_i(t) = c_i^k(t)$

6   For all $S \in S(t)$

7       $\text{price}(S) = \frac{c_i^k(t) - \underline{c}_i(t)}{|S(t)|}$

8 For $j \leftarrow t$ downto 1

9   if $C - \{(i, k)_j\}$ is feasible $C \leftarrow C - \{(i, k)_j\}$

At the beginning of the algorithm, the set C is empty (and thus not feasible). Let *Violation(C)* be an oracle that calculates the minimal violated sets of the graph G for a given C; the oracle does this by following Definition 1. The set $\mathcal{V}(t)$ holds all the minimal violated sets returned by the oracle at the beginning of each iteration t (line 3). Note that at the very beginning, the number of minimal violated sets is $N - 1$ (i.e. all $i \in V - \{r\}$). The algorithm selects the most cost-effective pair (i, k) (line 4);

here, c_i^k denotes the cost assigned to the node i that transmits at the power level k and $\underline{c}_i$ represents the cost induced by any previous selection of the node i into the set C. Thus, we allow a node to be selected more than one time in the final solution, which does not mean that the node actually transmits two times or more. This uniqueness is ensured by the delete step (lines 8 and 9). The set $S(t) \subseteq \mathcal{V}(t)$ is defined as follows $S(t) = \{S \in \mathcal{V}(t) : \delta^-(S) \text{ is hit by } (i,k)_t\}$. Informally, $S(t)$ comprises the minimal violated sets that are newly hit at the iteration t. The sets $S \in S(t)$ are then assigned the price (line 7), which will be used in the proof of Theorem 4.

In the rest of this section we evaluate the performance guarantee of the algorithm. Let A_t denote the event that a new minimal violated set is induced in iteration t. We first prove the following lemma:

Lemma 1

$$|\mathcal{V}(t+1)| = \begin{cases} |\mathcal{V}(t)| - |S(t)| + 1, & \text{if } A_t \\ |\mathcal{V}(t)| - |S(t)|, & \text{otherwise.} \end{cases}$$

Proof: Let us consider the iteration t with $|\mathcal{V}(t)|$ minimal violated sets. Let $(i,k)_t$ be selected into the set C at this iteration, that is, $(i,k)_t$ hits at least one minimal violated set $S \in \mathcal{V}(t)$. Then we have the following two possibilities: either by this transmission i produces no new minimal violated set (strongly connected component) or it produces one or more. Clearly, in the first case we have $|\mathcal{V}(t+1)| = |\mathcal{V}(t)| - |S(t)|$.

In the second case, the node i is included in any strongly connected component (minimal violated set) that it has newly produced. Consequently, there must exist a directed path from every node of such components to the node i, and vice versa, from the node i to any node of these components. This in turn means that these components belong to the same strongly connected component (i.e. the same minimal violated set). Therefore, i induces, at most, one new minimal violated set, in which case $|\mathcal{V}(t+1)| = |\mathcal{V}(t)| - |S(t)| + 1$. □

Let us introduce the following indicator variable:

$$I(t) = \begin{cases} 1, & \text{if in t a new violated set is induced} \\ 0, & \text{otherwise.} \end{cases}$$

Let m denote the total number of iterations of our algorithm, and l the total number of minimal violated sets during the course of the algorithm. Then by using Lemma 1 and observing that $|\mathcal{V}(1)| = N - 1$ and $|\mathcal{V}(m+1)| = 0$, we obtain: $l = \sum_{t=1}^{m} |S(t)| = \sum_{t=1}^{m} I(t) + N - 1$.

We next evaluate the bound on the total number of newly generated minimal violated sets.

Lemma 2

$$\sum_{t=1}^{m} I(t) \leq N - 2$$

Proof: By Definition 1, every newly created minimal violated set is a strongly connected component. Therefore, $\sum_{t=1}^{m} I(t)$ is, at most, the number of newly generated strongly connected components. At the very beginning, the number of eligible nodes (components) for the creation of newly strongly connected components is $N - 1$. Since each time a new strongly connected component is created at least two eligible components are merged, the number of eligible components is decreased by at least 1. Therefore, the total number of newly created strongly connected components is at most $N - 2$, which concludes the proof. □

By applying Lemma 2 to the expression for the total number of minimal violated sets l, we obtain: $l = \sum_{t=1}^{m} |S(t)| \leq 2N - 3$. We can use this inequality to obtain the upper bound on the total number

of iterations m. Having $|S(t)| = 1$ in every iteration t, we obtain: $m \leq 2N - 3$. Since the violation oracle can be implemented to run in polynomial time[2], our algorithm is polynomial in the total number of nodes N.

Let OPT denote the total cost of the optimal solution. We next prove the following lemma, which is similar to Lemma 2.3 in [27]:

Lemma 3 *For each $(i,k)_t$ selected into C, $\frac{c_i^k(t) - \underline{c}_i(t)}{|S(t)|} \leq \frac{OPT}{|\mathcal{V}(t)|}$.*

Proof: In any iteration t, transmitters from the optimal solution can cover the sets from $\mathcal{V}(t)$ at a cost of at most OPT. Consequently, the cost-effectiveness of any of these transmitters is at most $\frac{OPT}{|\mathcal{V}(t)|}$. Therefore, by selecting the most cost-effective $(i,k)_t$ at the iteration t (i.e. $(i,k)_t = \arg\min_{(i,k)} \frac{c_i^k(t) - \underline{c}_i(t)}{|S(t)|}$), we must have $\frac{c_i^k(t) - \underline{c}_i(t)}{|S(t)|} \leq \frac{OPT}{|\mathcal{V}(t)|}$. $\square$

Finally, we prove the following theorem on the performance guarantee of our approximation algorithm.

Theorem 4 *Algorithm 1 delivers a feasible solution of cost not larger than $c \log N \cdot OPT$, where $c = 2 \cdot \left(\frac{1}{\log e} + \frac{2}{\log N} \right) < 18$ for any $N \geq 2$. That is, Algorithm 1 is an $O(logN)$-approximation algorithm [3].*

Proof: Since the cost of each pair $(i,k)_t$ of the output C is evenly distributed among the newly hit minimal violated sets $S(t)$, $cost(C) = \sum_{j=1}^{l} price(S_j) = \sum_{t=1}^{m} \sum_{j=1}^{|S(t)|} price(S_j)$. Now we have:

$$cost(C) = \sum_{t=1}^{m} \sum_{j=1}^{|S(t)|} \frac{c_i^k(t) - \underline{c}_i(t)}{|S(t)|} \tag{2}$$

$$\leq \sum_{t=1}^{m} \sum_{j=1}^{|S(t)|} \frac{OPT}{|\mathcal{V}(t)|} \tag{3}$$

$$= 2 \cdot OPT \sum_{t=1}^{m} \sum_{j=1}^{|S(t)|} \frac{1}{2|\mathcal{V}(t)|} \tag{4}$$

$$\leq 2 \cdot OPT \sum_{t=1}^{m} \sum_{j=1}^{|S(t)|} \frac{1}{2|\mathcal{V}(t)| - j + 1} \tag{5}$$

$$< 2 \cdot OPT \sum_{i=1}^{2(N-1)} \frac{1}{i} \tag{6}$$

$$\leq 2 \cdot [\ln(N - 1) + \ln 2 + 1] \cdot OPT \tag{7}$$

$$< 2 \cdot \left(\frac{1}{\log e} + \frac{2}{\log N} \right) \cdot \log N \cdot OPT \tag{8}$$

where (3) follows from Lemma 3; (6) follows from Lemma 1, $\sum_{t=1}^{m} |S(t)| < 2(N-1)$ and $|\mathcal{V}(1)| = N - 1$; and (7) follows from the inequality $\sum_{j=1}^{n} \frac{1}{j} \leq \ln n + 1$. $\square$

In this subsection we developed the approximation algorithm for the general graph version (MBC). In the following two subsections, we first elaborate on a centralized heuristic algorithm deliberately

---

[2] For example STRONGLY CONNECTED COMPONENT ALGORITHM given in [29] runs in $O(N + |E|)$, where $|E| < N^2$ (Theorem 2.19).

[3] log designates base 10 logarithm.

designed for the geometric version (GMBC). Then, we explain how it can be converted to a distributed algorithm.

5.2 A heuristic based approach

Let us first present an informal description of the heuristic we propose. We first construct a feasible solution (an initial feasible broadcast tree). Then we improve this solution by exchanging some existing branches in the initial tree for new branches so that the total energy necessary to maintain the broadcast tree is reduced; and in doing this the feasibility of the obtained solution remains intact. We call the difference in the total energies of the trees before and after the branch exchange a *gain*. In our heuristic, the notion of gain is used as the criterion for the selection of transmitting nodes in the broadcast tree.

We use the link-based minimum spanning tree (MST) as the initial feasible solution. The main reason we choose MST is that it performs quite well, even as a final solution to our problem (which can be seen from the simulation results in Section 6).
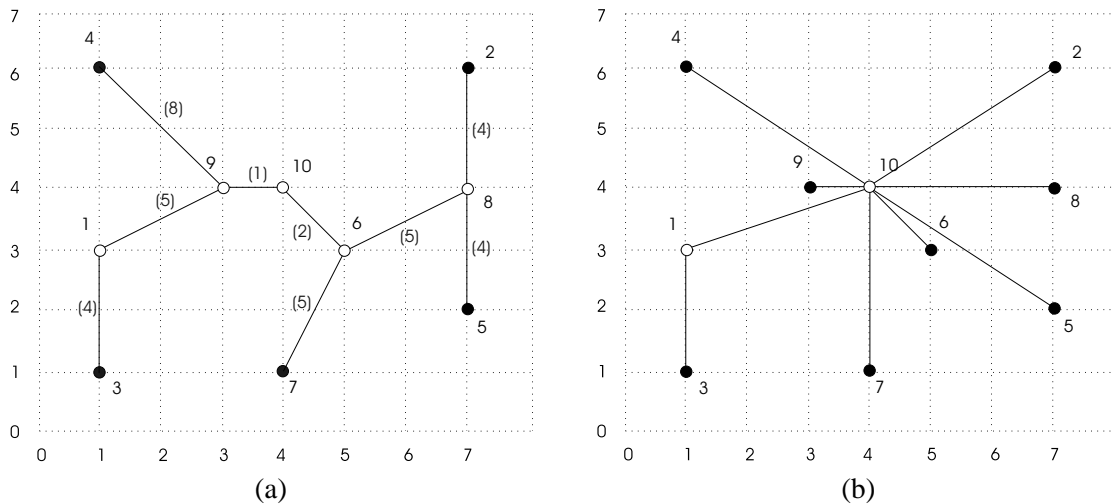

Figure 2: Example of the EWMA algorithm: (a) the initial MST ($e_{MST} = 23$) and (b) the broadcast tree obtained by EWMA ($e_{EWMA} = 17$)

We will now describe in detail our algorithm, which we call *Embedded Wireless Multicast Advantage* (EWMA). An example is provided in Figure 2. Let us first introduce some notations. Let C denote the set of covered nodes, F the set of transmitting nodes of the final broadcast tree, and E the set of *excluded* nodes. Node i is said to be an *excluded node* if is transmitting node in the initial solution but not in the final solution (i.e. $i \notin F$). Notice that the contents of the above sets change throughout the execution of the EWMA and that the sets do not hold any information about the MST. Initially, $C = \{r\}$, where r is the source node (node 10 in our example), and sets F and E are empty.

In this example, we assume a propagation loss exponent $\alpha = 2$. After the MST has been built in the *initialization* phase, we know which nodes in the MST are transmitting nodes and their respective transmission energies. In our example, the transmitting nodes are 10, 9, 6, 1, 8, and their transmission energies are 2, 8, 5, 4, and 4, respectively. The total energy of the MST is $e_{MST} = 23$. Notice here that we take into consideration the wireless multicast advantage in the evaluation of the cost of the MST. Notice also that $C = \{10\}$, and $F = E = \{\emptyset\}$. In the second phase, EWMA builds a broadcast

tree from nodes in the set $(C - F) - E$ by determining their respective gains. The gain of a node v is defined as the decrease in the total energy of the broadcast tree obtained by excluding some of the nodes from the set of transmitting nodes in MST, in exchange for the increase in node v's transmission energy. Notice that this increase of node v's transmission energy has to be sufficient for it to reach all the nodes that were previously covered by the nodes that were excluded. Consequently, the feasibility of the solution is preserved. At this stage of the algorithm, the set $(C - F) - E$ contains only the source node 10. Thus for example, in order to exclude node 8, the source node 10 has to increase its transmission energy by (see Figure 2):

$$\triangle e_{10}^8 \quad = \quad \max_{i \in \{2,5\}} \{e_{10,i}\} - e_{10} \ = 13 - 2 = 11$$

The gain (g_{10}^8) obtained in this case is:

$$g_{10}^8 \quad = \quad e_6 + e_8 + e_9 - \triangle e_{10}^8 \ = 5 + 4 + 8 - 11 = 6$$

where $e_i, i = \{6, 8, 9\}$, is the energy at which node i transmits in MST. Notice that, in addition to node 8, the nodes 6 and 9 can also be excluded.

Likewise, $g_{10}^1 = 5$, $g_{10}^6 = -2$, and $g_{10}^9 = 6$. Having the gains for all nodes from $(C - F) - E$, our algorithm selects the node with the highest positive gain in the set F. Our algorithm then adds all the nodes that this node excludes to the set E. Thus the source node 10 is selected in the set F to transmit with energy that maximizes its gain, that is:

$$e_{10}^{'} \quad = \quad e_{10} + \arg\max_{\triangle e_{10}^i} \{g_{10}^i\}, \qquad g_{10}^i \geq 0$$

The source node 10 transmits with energy $e_{10}^{'} = e_{10} + \triangle e_{10}^8 = 2 + 11 = 13$ at which it can cover nodes 6, 8, 9 and all their *child* nodes in MST. Node j is said to be a *child node* of node i if node j is included in the broadcast tree by node i. Hence, at this stage we have $C = \{1, 2, 4, 5, 6, 7, 8, 9, 10\}$, $E = \{6, 8, 9\}$ and $F = \{10\}$. If none of the nodes from $(C - F) - E$ has a positive gain, EWMA selects among them the node that includes its child nodes in the MST at minimum cost (energy).

The above procedure is repeated until all nodes in the network are covered. In our example, there is still one node to be covered, namely node 3. Again, EWMA scans the set $(C - F) - E = \{1, 2, 4, 5, 7\}$ and at last selects node 1 to be the next forwarding node. When node 1 transmits with energy $e_1 = 4$, all nodes are covered ($C = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$) and the algorithm terminates. At the final stage we have $E = \{6, 8, 9\}$ and $F = \{1, 10\}$. The resulting tree, shown in Figure 2b, has a cost $e_{EWMA} = 17$. Notice that our algorithm always results in a broadcast tree with a total energy $\leq e_{MST}$, which is, in the case of Euclidean MST, at most $12e_{opt}$ [30].

In the next subsection we explain how to convert our centralized heuristic algorithm to a distributed algorithm.

5.3 Distributed implementation of EWMA

One of the major research challenges, with respect to the broadcasting problem, is the development of a distributed algorithm [31, 30]. In the following we describe our solution.

Let us first introduce the notations we will be using. Let node i transmit at power level $p \in P$. We denote the set of nodes that are covered by this transmission with V_i^p. Let node j be a neighbor of i, that is, $j \in V_i$. We denote with O_{ij}^p the set of nodes belonging to $V_i^p \cap V_j$ and call it the *overlapping* set. We assume that each node knows its two-hop neighborhood. So, once node j receives a message

from node i, it can learn which of the nodes from its neighbor set V_j have also received the message by calculating the overlapping set O_{ij}^p. The neighbors of node j that have not yet received the message are said to be *uncovered*, and we denote this set with U_j where $U_j = V_j - O_{ij}^p$. If node j is a forwarding node in the MST, then the set of yet uncovered children nodes of node j in the MST is denoted with U_j^{mst} where $U_j^{mst} = V_j^{mst} - O_{ij}^p$. Here, V_j^{mst} is the set comprising all the children nodes of j in the MST. Finally, we denote with e_j^{mst} the energy with which node j transmits in the MST.

Our distributed algorithm is divided into two phases. In the first phase, all nodes run a distributed algorithm proposed by Gallager et al. [7] to construct a *minimum-weight spanning tree*. The total number of messages required for a graph of $|V|$ nodes and $|E|$ edges is at most $5|V|\log_2|V| + 2|E|$, and the time until completion is $O(|V|\log|V|)$ [7]. Notice that Gallager et al. considered the link-based model, whereas we use the node-based multicast model, which captures the wireless multicast advantage property [31]. As a result, the total number of messages required in our model may be considerably lower. We require that at the end of the first phase, each node has information about the cost of its two-hop neighbors related to the MST built.
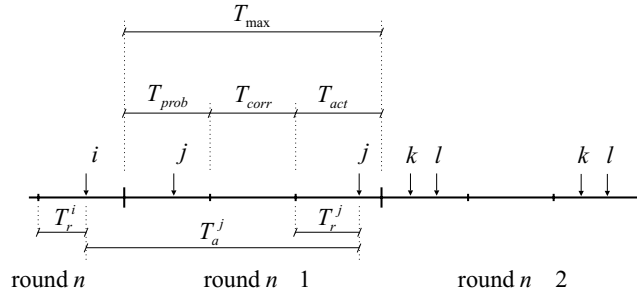

Figure 3: Synchronization of the second phase of distributed EWMA

In the second phase, the final broadcast tree is built up. The main difficulty in this distributed setting is the unavailability of information about which nodes have been covered, up to a certain moment. In order to cope with this problem, we apply two techniques. First, we organize this second phase in rounds. Second, we require that the identities of the nodes on the transmission chain from the source to a given node, along with their respective transmission powers are propagated along that chain to the node in question (source routing technique).

Each round of the second phase is $T_{\max}$ long. Rounds are additionally divided into three time periods, namely, a *probation* period (T_{prob}), a *correction* period (T_{corr}), and an *active* period (T_{act}), which are all known by network nodes (Figure 3). Let node i transmit at T_r^i time from the beginning of the active period of round n. Node j receives this message and begins the following update procedure. It calculates the overlapping set for the sender i and for other transmitters on this chain of transmitting nodes for which node j has neighbors in common (recall that this information is propagated along the chain). If node j is a forwarding node in the MST and it finds that the set of uncovered nodes U_j^{mst} is empty for the received message, it will not re-broadcast the message. Otherwise, (i.e. if U_j^{mst} is non-empty or j was a leaf node in the MST), it calculates the gains it can achieve by covering yet uncovered nodes (based on locally available information), and selects the maximum gain $g_{j\max}$. In the case $g_{j\max} > 0$, node j can contribute to the decrease of the total cost of the broadcast tree and its transmission energy increases as follows: $e_j = e_j + \arg\max_{\triangle e_j^l}\{g_j^l\}$, otherwise ($g_{j\max} \leq 0$) its transmission energy remains unchanged.

At this stage, node j waits for some time period T_a^j before possibly re-broadcasting the message. The

13

waiting period is given as follows:

$$T_a^j = T_{max} + T_r^j - T_r^i$$

where $T_r^j = \frac{\Delta_1}{g_{j\,max}}$ if $g_{j\,max} > 0$, and $T_r^j = \Delta_2 \cdot e_j$ if $g_{j\,max} \leq 0$ and $e_j > 0$. In the first case the waiting period T_a^j is reciprocal to the gain, in order to give priority to nodes with higher positive gains over nodes with lower positive gains. In the second case, the waiting period T_a^j is proportional to the transmission energy in order to give priority to nodes with lower transmission energies over nodes with higher transmission energies. Additionally, the nodes with positive gains are given priority to the nodes with low transmission energies (i.e. $\frac{\Delta_1}{g_{j\,max}} \ll \Delta_2 \cdot e_j$). This property is ensured by appropriately setting the constants Δ_1 and Δ_2.

Since node j calculates the gains based on only locally available information, in the calculation of the gains, node j can try to exclude already excluded nodes. In order to prevent this, node j transmits a probe message during the probation period T_{prob} of round $n + 1$. Note that by knowing T_{act} and T_r^i (which j received from i) node j actually knows when round $n + 1$ starts. The probe message carries the addresses of all the nodes by exclusion of which node j attains $g_{j\,max} > 0$, and it carries the starting time of the correction period. If some of these nodes have already been excluded, they will respond back to node j during the correction period. Node j will accordingly update its gain and the waiting period T_a^j by taking into account the already elapsed time of the waiting period. The duration of the probation and correction periods should be such that any potential forwarding node is given the chance to test its prospect of actually being the forwarding node.

Finally, node j enters into the active period. Again, based on the knowledge of T_{prob} and T_{corr}, node j knows when the active period of round $n + 1$ starts. If during that period and before expiration of the waiting period T_a^j node j receives a duplicate message, it repeats the update procedure above, otherwise, upon expiration of T_a^j, it re-broadcasts the message with energy e_j, stores this value and marks itself as the forwarding node. In our example shown in Figure 3, node j decides to be the forwarding node and broadcasts a message at power e_j. By doing so, it initiates the update procedure at nodes k and l that repeat the whole process.

Next we show under which conditions the waiting period T_a^j expires solely during the active period of round $n + 1$. From Figure 3 we can see that this happens if T_a^j conforms to the following conditions:

$$T_a^j \geq T_{\max} - T_r^i$$
$$T_a^j \leq T_{\max} + T_{act} - T_r^i$$

From the first inequality and the definition of T_a^j we obtain that $T_r^j \geq 0$, which is always satisfied. Along the same lines, from the second inequality we obtain that $T_r^j \leq T_{act}$. Consequently, we define the active period as follows:

$$
\begin{aligned}
T_{act} &= \max_{j \in F}\{T_r^j\} \\
&= \max_{j \in F}\{\Delta_2 \cdot e_j\}
\end{aligned}
$$

where the second equality follows from the fact that $\frac{\Delta_1}{g_{j\,max}} \ll \Delta_2 \cdot e_j$. Now, since we already have decided on Δ_1 and Δ_2, we only have to find the cost of the most expensive edge in the MST. Note that this information can be obtained from the first phase of the algorithm. This, in addition to the appropriate selection of the periods T_{prob} and T_{corr}, ensures a synchronous execution of the second phase of the distributed algorithm.

The duration of the second phase is bounded by $|F| \cdot T_{\max}$, where F is the set of the forwarding nodes at the end of the second phase. Thus, at the end of the second phase, the broadcast tree is built (i.e. we have a set of forwarding nodes F and their respective transmission energies for a given source node). Any subsequent broadcast message originating from the source node can be disseminated along the tree in an asynchronous way (i.e. forwarding nodes may re-broadcast a message immediately upon receiving it).

6   Performance evaluation

We performed a simulation study to evaluate our centralized algorithm (EWMA) and its distributed version.

We compared the centralized version of our algorithm (EWMA) with BIP and MST algorithms. We performed simulations using networks of four different sizes: 10, 30, 50 and 100 nodes. The nodes in the networks are distributed according to a spatial Poisson distribution over the same deployment region. Thus, the higher the number of nodes, the higher the network density. The source node for each simulation is chosen randomly from the overall set of nodes. The maximum transmission range is chosen such that each node can reach all other nodes in the network. The transmission power used by a node in transmission (d^α) depends on the reached distance d, where the propagation loss exponent α is varied. Similarly to Wieselthier et al. in [31], we ran 100 simulations for each simulation setup consisting of a network of a specified size, a propagation loss exponent α, and an algorithm.
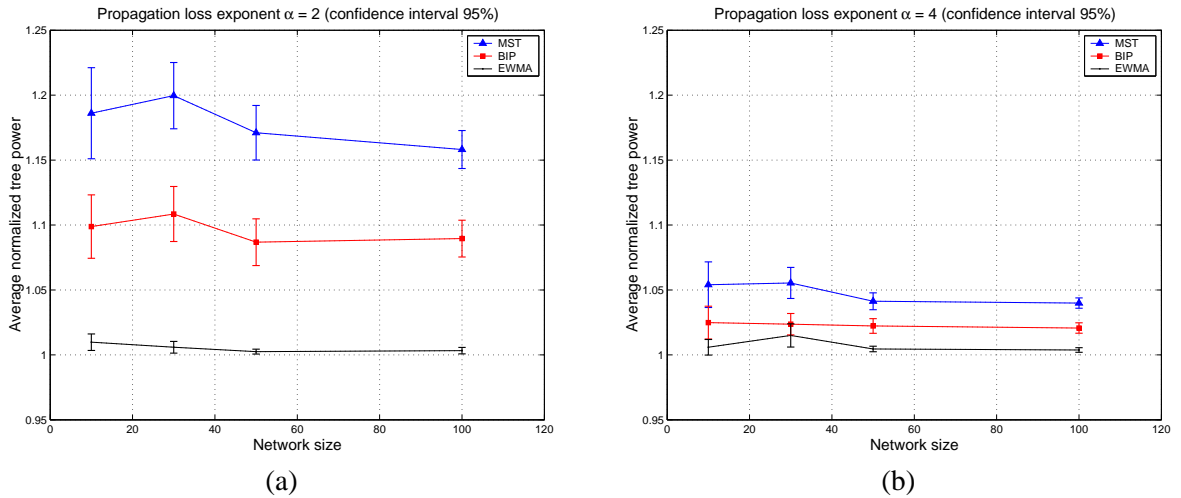

Figure 4: Normalized tree power comparison: (a) $\alpha = 2$ and (b) $\alpha = 4$

The performance metric used is the total power of the broadcast tree. Here we use the idea of the *normalized tree power* [31]. Let $p_i(m)$ denote the total power of the broadcast tree for a network instance m, generated by algorithm $i = \{EWMA, BIP, MST\}$. Let p_0 be the power of the lowest-power broadcast tree among the set of algorithms performed and all network instances (100 in our case). Then the normalized tree power associated with algorithm i and network instance m is defined as follows: $p_i'(m) = \frac{p_i(m)}{p_0}$.

Let us first consider the performance of the algorithms shown in Figure 4. We can see the average normalized tree power (shown on the vertical axis) achieved by the algorithms on networks of different
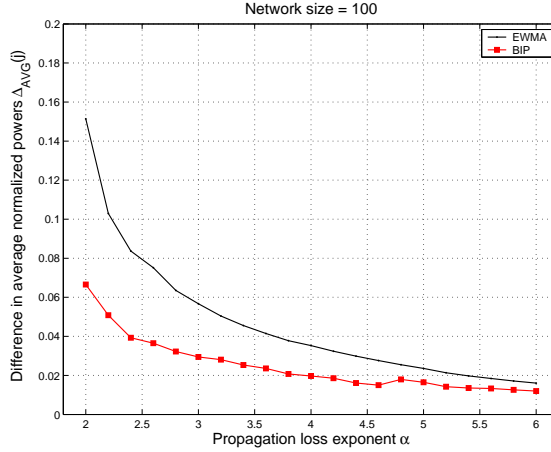
15

Figure 5: Normalized tree power as a function of the propagation loss exponent α

sizes (the horizontal axis) for (a) $\alpha = 2$, (b) $\alpha = 4$. To estimate the average power, we used an interval estimate with a confidence interval of 95%. The figure shows that the solutions for the broadcast tree obtained by EWMA have, on the average, lower costs than the solutions of BIP and MST. (This is also true for $\alpha = 3$, which is not shown in the figure). However, we notice that for the propagation loss exponent of $\alpha = 4$, the confidence intervals of the algorithms overlap for certain cases, which means that the solutions obtained by the algorithms are not significantly different. Thus the figure also reveals that the difference in performance decreases as the propagation loss exponent increases. This is better seen in Figure 5, where the difference in the average normalized tree powers between EWMA (BIP) and MST algorithms ($\Delta_{AVG}(j) = AVG(p'_{MST}(m)) - AVG(p'_j(m))$, $j = \{BIP, EWMA\}$) is shown for different values of the propagation loss exponent (the horizontal axis). Notice here that the larger the difference $\Delta_{AVG}(j)$, the lower the cost of the broadcast tree. The main reason for such behaviour is that by increasing the propagation loss exponent, the cost of using longer links increases as well. Consequently, EWMA and BIP select their transmitting nodes to transmit using lower power levels, which is typical for the transmitting nodes of MST. Hence, in a sense, EWMA and BIP's broadcast trees *converge* to the MST tree when α increases. This indicates that in scenarios where α takes higher values, MST performs quite well.

We also conducted a simulation study of the distributed algorithm presented in Section 5.3. The performance metric used here is the same as in the case of the centralized algorithm and is based again on the normalized tree power. However, here we do not consider the cost of building a broadcast tree, but only the cost of the final tree produced by the distributed algorithm. The performance of the distributed algorithm is compared to that of the centralized algorithms, and is shown in Figure 6. We can see that broadcast trees produced by distributed EWMA have, on the average, lower costs than those obtained by the centralized BIP and MST. Also, we can see that distributed EWMA performs almost as well as its centralized counterpart. Note that the results for the centralized algorithms differ between Figure 4(a) and Figure 6. This is because here we run another set of simulations for all the algorithms, and for each network the source node is chosen at random.

Based on our simulation results, we conclude that EWMA utilizes the wireless multicast advantage property at least as well as BIP. The main problem with BIP is that it is not easy to distribute. On the other hand, we showed here that EWMA can be easily distributed.
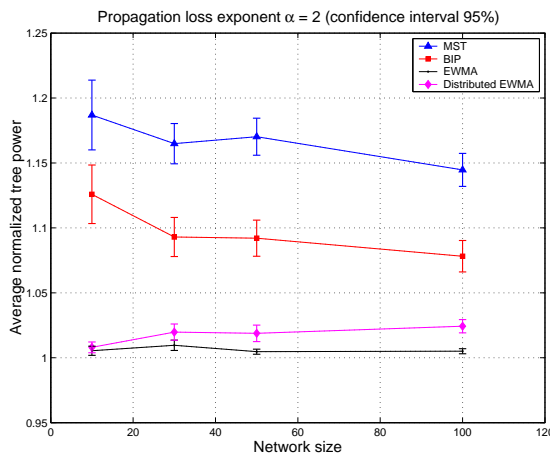
Figure 6: Normalized tree power comparison (distributed EWMA; $\alpha = 2$)

7 Conclusion

We provide novel contributions to several relevant aspects of power-efficient broadcasting in all-wireless networks. First, we studied the complexity of the problem: we discussed two configurations, represented each by a specific graph - a general graph and a graph in Euclidean space (geometric case). For both, we showed that the problem is NP-complete. Furthermore, we showed that the general version cannot be approximate better than $O(\log N)$.

Second, we elaborated an approximation algorithm for the general version that achieves approximation ratio of $18 \log N$. Then we elaborated a new algorithm called Embedded Wireless Multicast Advantage (EWMA) that compares well with the existing proposals. Finally, we explained how centralized EWMA can be converted to a distributed algorithm, which is almost as energy-efficient as its centralized counterpart.

In future work we intend to study how to cope with the mobility of the nodes and study the minimum-energy multicast problem.

References

[1] J. Chang and L. Tassiulas. "Energy Conserving Routing in Wireless Ad-hoc Networks". In *Proceedings of IEEE INFOCOM 2000*. ACM, 2000.

[2] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the Complexity of Computing Minimum Energy Consumption Broadcast Subgraphs. In *Proceedings of 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2001)*, 2001.

[3] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms. Second Edition*. The MIT Press, 2001.

[4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MOBICOM 1999)*, Seattle, Washington, August 1999.

[5] O. Eğecioğlu and T. Gonzalez. Minimum-energy Broadcast in Simple Graphs with Limited Node Power. In *Proceedings of IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2001)*, pages 334–338, Anaheim, CA, August 2001.

[6] A. Faragó and V. Syrotiuk. Algorithmic Problems in Power Controlled Ad Hoc Networks. In *Proceedings of the 14th International Conference on Parallel and Distributed Computing Systems (PDCS 2001)*, Dalas, Texas, August 2001.

[7] R. Gallager, P. Humblet, and P. Spira. A Distributed Algorithm for Minimum-Weight Spanning Trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, January 1983.

[8] M. Garey and D. Johnson. *Computers and intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 2000)*, Maui, Hawaii, January 2000.

[10] D. S. Hochbaum. *Approximation Algorithms for NP-hard problems*. PWS Publishing Company, 1997.

[11] J.-P. Hubaux, T. Gross, J.-Y. L. Boudec, and M. Vetterli. Towards self-organized mobile ad hoc networks: the Terminodes Project. *IEEE Communications Magazine*, 39(1), January 2001.

[12] E.-S. Jung and N. Vaidya. "An Energy Efficient MAC Protocol for Wireless LANs". In *IEEE INFOCOM 2002*, New York, USA, June 2002.

[13] F. Li and I. Nikolaidis. On Minimum-Energy Broadcasting in All-Wireless Networks. In *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, Tampa, Florida, November 2001.

[14] L. Li, V. Bahl, Y. Wang, and R. Wattenhofer. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In *Proceedings of IEEE INFOCOM 2001*, April 2001.

[15] Q. Li, J. Aslam, and D. Rus. Online Power-aware Routing in Wireless Ad-hoc Networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networks (MOBICOM 2001)*, Rome, Italy, July 2001.

[16] W. Liang. Constructing Minimum-Energy Broadcast Trees In Wireless Ad Hoc Networks. In *Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2002)*, Lausanne, Switzerland, June 2002.

[17] W. Liang. Constructing Minimum-Energy Broadcast Trees In Wireless Ad Hoc Networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002)*, Lausanne, Switzerland, June 2002.

[18] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, May 1982.

[19] K. Makki, N. Pissinou, and O. Frieder. "Efficient Solutions to Multicast Routing in Communication Networks". *ACM Mobile Networks and Applications (MONET)*, 1(2):221–232, 1996.

[20] M. Nagy and S. Singh. "Multicast Scheduling Algorithms in Mobile Networks". *Cluster Computing*, 1(2):177–185, 1998.

[21] G. Pottie and W. Kaiser. Wireless Integrated Network Sensors. *Communication of the ACM*, 43(5):51–58, May 2000.

[22] J. Rabaey, M. Ammer, J. da Silva Jr., D. Patel, and S. Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer Magazine*, 33(7):42–48, July 2000.

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

[24] V. Rodoplu and T. H. Meng. Minimum Energy Mobile Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 17(8), August 1999.

[25] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. B. Srivastava. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January-March 2002.

[26] S. Singh, C. Raghavendra, and J. Stepanek. Power-Aware Broadcasting in Mobile Ad Hoc Networks. In *Proceedings of IEEE PIMRC'99*, Osaka, Japan, September 1999.

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

[28] M. Čagalj, J.-P. Hubaux, and C. Enz. Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networks (MOBICOM 2002)*, Atlanta, Georgia, September 2002.

[29] B. K. . J. Vygen. *Combinatorial Optimization. Theory and Algorithms*. Springer-Verlag, 2000.

[30] P.-J. Wan, G. Calinescu, and O. F. X.-Y. Li. Minimum-energy broadcast routing in static ad hoc wireless networks. In *IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.

[31] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks. In *IEEE INFOCOM 2000*, pages 586–594, Tel Aviv, Israel, 2000.

[32] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Algorithms for Energy-Efficient Multicasting in Static Ad Hoc Wireless Networks. *ACM Mobile Networks and Applications (MONET)*, 6(3):251–263, June 2001.

[33] Y.Xu, J.Heidemann, and D. Estrin. Geography-informed Energy Conservation for Ad Hoc Routing. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networks (MOBICOM 2001)*, July 2001.