REPORTS IN INFORMATICS

ISSN 0333-3590

An Integer Programming Approach for Performance Evaluation of Minimum-Energy Broadcasting and Multicasting in Wireless Ad Hoc Networks

Di Yuan, Joanna Bauer and Dag Haugland

REPORT NO 288 December 2004

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/ps/2004-288.ps

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

An Integer Programming Approach for Performance Evaluation of Minimum-Energy Broadcasting and Multicasting in Wireless Ad Hoc Networks

Di Yuan[*], Joanna Bauer[†] and Dag Haugland[‡]

28th December 2004

Abstract

We present an integer programming approach for the purpose of performance evaluation of heuristic algorithms for minimum-energy broadcasting and multicasting in wireless ad hoc networks. The approach consists of an integer programming model and a Lagrangian relaxation scheme. The former permits optimal solution for small networks, and the latter allows us to compute a good approximation of the optimum for large-sized networks. We present extensive numerical experiments of our approach, and evaluate the performance of the most known heuristic algorithm, referred to as the multicast incremental power (MIP) algorithm, for minimum-energy broadcasting and multicasting in networks of various sizes.

Keywords: ad hoc networks, broadcasting, multicasting, integer programming

1 Introduction

Many applications of wireless networks require that networking units are energy-aware. This is particularly true for wireless ad hoc networks, in which nodes often use battery as the only energy source. Energy efficiency is therefore a crucial issue in design of such systems.

In this paper we study the problem of minimum-energy routing of broadcast and multicast sessions in ad hoc networks. Multicast refers to the process of delivering messages from a source node to a set of destination nodes. Broadcast is a special case of multicast where the set of destination nodes comprise all nodes other than the source.

In a wireless environment, broadcasting is an inherent characteristic of signal propagation. As long as omni-directional antennas are concerned, a transmission power corresponds to a range of coverage in all directions. Thus, it is sufficient to transmit once in order to deliver a message to all units within the range. In [27], this property is referred to as the "wireless multicast advantage". As a consequence of this property, the power needed to reach a set of receiving units is not the sum, but the maximum of the power for reaching any of these units.

The total power consumption of a multicast session is the sum of the transmission power used by the source and the nodes involved in message forwarding.

[*]Email: diyua@itn.liu.se, Department of Science and Technology, Linköping University, SE-601 74 Norrköping, Sweden

[†]Email: Joanna.Bauer@ii.uib.no, Department of Informatics, University of Bergen, PB. 7800, N-5020 Bergen, Norway

[‡]Email: Dag.Haugland@ii.uib.no, Department of Informatics, University of Bergen, PB. 7800, N-5020 Bergen, Norway

Minimum-energy multicasting refers to the optimization problem of assigning node transmission power with the objective of minimizing the total power used for multicast routing. In the subsequent text, we use MEMP to denote this optimization problem. The problem of minimum-energy broadcasting is treated as a special case of MEMP. Because the objective function of MEMP originates from nodes rather than from links, the structure of MEMP is quite distinct from that of the classical minimum spanning tree (MST) problem. This distinction has the consequence that MEMP is NP-hard [4, 11].

Wieselthier et al. [27, 28] presented several heuristic algorithms for MEMP. Among these, one is referred to as the multicast incremental power (MIP) algorithm. To the best of our knowledge, this is the most known algorithm for MEMP. Indeed, performance evaluation of other heuristics has used results of MIP as benchmark. However, in the absence of a procedure that is capable of finding the optimum of MEMP, or, alternatively, computing a sharp (lower) bound to the optimum, it is not possible to assess the performance of any heuristic in terms of optimality.

In this paper we propose an integer programming approach for performance evaluation of algorithms for MEMP. We present an integer programming model that permits exact solution of MEMP for small-sized networks. To deal with the computational complexity of MEMP for large-scale networks, we propose an effective Lagrangian relaxation scheme that yields a good approximation of the optimum. This gives us a computationally efficient approach for assessing the numerical performance of any heuristic algorithm for MEMP. In our numerical experiments, we apply our approach to evaluate the performance of MIP (and indirectly other heuristics) in terms of optimality for a large number of network instances of various sizes.

The remainder of the paper is organized as follows. In Section 2 we present the network model. Section 3 is devoted to a review of related work, in particular the MIP algorithm. The integer programming model and the Lagrangian relaxation scheme are discussed in Sections 4 and 5, respectively. We present results of performance evaluation in Section 6. In Section 7 we draw some conclusions and outline some possible extensions of the current work.

2 The Network Model

In our network model, we use a set of nodes N to represent the networking units in a wireless ad hoc network. The network itself is modeled using a graph $G = (N, A)$, where A is the set of (directed) possible communication links, defined as $A = \{(i, j), i, j \in N, i \neq j\}$. Let p_{ij} denote the (minimum) power required to transmit a message from node i to node j. Typically, p_{ij} is a function of the distance between i and j. Denoting this distance by d_{ij}, a widely-used formula for calculating p_{ij} is

$$p_{ij} = \kappa d_{ij}^{\alpha}. \tag{1}$$

Here, κ is a positive constant, and α is an environment-dependent parameter, whose value is between 2 and 4. (The power increases therefore at least quadratically in distance.) Since the value of κ is irrelevant to the problem, it is henceforth assumed that $\kappa = 1$. Although our integer programming approach is not dependent on the formula used for calculating the transmission power, we will use (1) in our numerical experiments as it has been used by many authors (e.g., [1, 3, 4, 7, 8, 9, 10, 11, 20, 21, 25, 27, 28]).

A multicast session is characterized by a source node s and a set of destination nodes D. Broadcasting is the special case of multicasting when $D = N \setminus \{s\}$. A destination node in the set D receives messages from the source node s either directly, or via some other nodes. We assume that any node is able to forward messages for other nodes. Connectivity between nodes is determined by transmission power. If

node i transmits at a power of, say, P_i, then it can send a message to all nodes j for which $p_{ij} \leq P_i$ in a single transmission. Therefore, the power necessary to reach a set of nodes equals the maximum power needed to reach any of them. Problem MEMP amounts to finding the transmission power of the nodes to enable message routing from s to all the nodes in D, with the objective of minimizing the total power[1].

Because all nodes in D must be connected to s (directly or indirectly), any optimal solution to MEMP can be characterized by a tree rooted at s. For this reason, all heuristic algorithms for MEMP construct a solution with a tree structure. We use the term multicast tree to refer to a tree representing a feasible solution of MEMP. Note that nodes other than those in the set D can be part of the tree, either for providing connectivity, or for reducing the power consumption, or both. As another observation, it is clear that, if our objective function would be the sum of the transmission power to establish links, the problem is the well-known Steiner tree problem, and, in addition, its special case of broadcasting reduces to MST.

3 Related Work

Wieselthier et al. [27, 28] were among the first who studied MEMP. They derived several heuristics for solving the broadcast version of MEMP. Among them, one became known as the broadcast incremental power (BIP) algorithm. The BIP algorithm involves two phases: tree construction and power reduction. The first phase, tree construction, is an adaptation of the Prim's algorithm. The result of this phase is a broadcast tree. Starting from the source, the algorithm builds up a tree by adding the node requiring a minimum amount of incremental power in every iteration. A formal description of this phase is given below, where we use N_T to denote the set of nodes that are included in the broadcast tree constructed by BIP. The tree itself is represented by the sets $C(i)$ and $F(i)$, $i \in N$, where $C(i)$ is the set of child nodes of i, and $F(i)$ is the set of upstream nodes of i (i.e. nodes on the path between s and i).

1. Set $P_i = 0$ and $C(i) = \emptyset\ \forall i \in N$, and set $F(s) = \{s\}$ and $N_T = \{s\}$.
2. Find a pair of nodes, $i^* \in N_T$, and $j^* \in N \backslash N_T$, such that $\max\{p_{i^*j^*} - P_{i^*}, 0\} \leq \max\{p_{ij} - P_i, 0\}, \forall i \in N_T, j \in N \setminus N_T$.
3. Set $P_{i^*} = \max\{p_{i^*j^*}, P_{i^*}\}$ and $N_T = N_T \cup \{j^*\}$.
4. Set $C(i^*) = C(i^*) \cup \{j^*\}$ and $F(j^*) = F(i^*) \cup \{j^*\}$.
5. If $N_T = N$, terminate. Otherwise go to Step 2.

The running time required by this first phase is $O(|N|^2)$.

In the second phase of the BIP algorithm, a procedure called sweep is used to improve the total power by removing some unnecessary transmissions. The sweep procedure examines, for any pair of nodes i and j, whether the current transmission range of i covers some child nodes of j. If this is the case, then it may be possible for j to reduce its transmission power. The subset of $C(j)$ that may choose i as the new parent node is $C(i,j) = \{k \in C(j) : P_i \geq p_{ik}\} \setminus F(i)$. If the node (or the nodes) defining the current transmission power of j belongs to $C(i,j)$, then node j can reduce its power to $\max_{k \in C(j) \setminus C(i,j)} p_{jk}$ without breaking network connectivity. This is illustrated in Figure 1. The power-saving operation is applied iteratively to the nodes in the sweep procedure. Typically, one application of the sweep procedure

[1] Multiple multicast sessions may take place simultaneously in the network. Generally speaking, a node may use different power levels for the multicast sessions in which it participates [27].

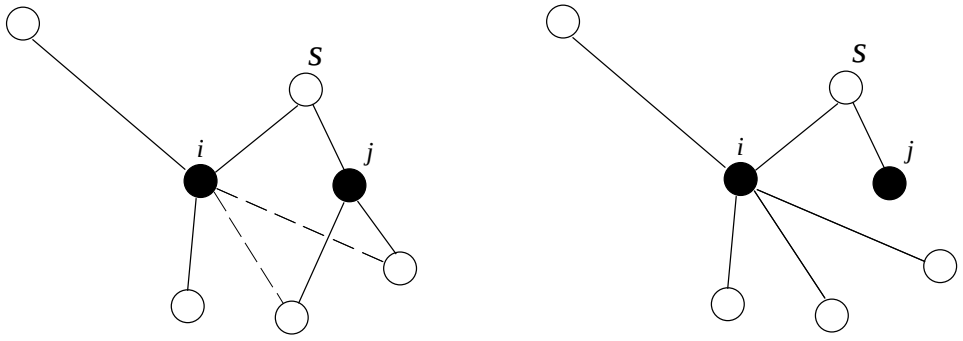

Figure 1: An example of the sweep operation.

yields significant power improvement, whereas only small additional gains can be obtained by applying the procedure more than once.

If the sweep operation that contributes to the largest power reduction is chosen first, the procedure can be expressed in algorithmic terms as follows:

1. $\delta^* = 0$
2. **repeat**
3. **for** $\forall i \in N$
4. **for** $\forall j \in N \setminus \{i\}$
5. Find $C(i,j) = \{k \in C(j) : P_i \geq p_{ik}\} \setminus F(i)$
6. $\delta = P_j - \max\{p_{ik} : k \in C(j) \setminus C(i,j)\}$
7. **if** $\delta > \delta^*$
8. $(i^*, j^*, \delta^*) = (i, j, \delta)$
9. **if** $\delta^* > 0$
10. **for** $\forall k$ in subtrees with root in $C(i^*, j^*)$
11. Let $F(k) = (F(k) \setminus F(j^*)) \cup F(i^*)$
12. $C(i^*) = C(i^*) \cup C(i^*, j^*)$
13. $C(j^*) = C(j^*) \setminus C(i^*, j^*)$
14. $P_{j^*} = P_{j^*} - \delta^*$
15. **until** $\delta^* = 0$

Clearly, one iteration of the sweep procedure is accomplished in $O(|N|^3)$ time. An analysis of the number of iterations is beyond the scope of this work, but a trivial bound is $O(|N|^2)$ iterations. Experiments show however that this bound is very conservative.

Remark 1. In [27, 28], the original sweep procedure is illustrated by examples in which node j is always one hop from i. It is however straightforward to generalize the procedure for any pair of nodes. We use therefore the generalized version of sweep in our numerical experiments.

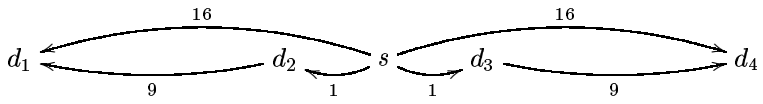

Figure 2: Case that BIP does not solve to optimality.

Remark 2. For arbitrary values of p_{ij} (i.e., formula (1) does not have to apply), the BIP algorithm finds the optimal solution for every instance consisting of at most three nodes. On the other hand, BIP can already be fooled by an instance of only four nodes, even if we assume (1) and Euclidian distances. Such an instance is, for example, given by a sender with coordinates $(0, 0)$, three destination nodes at $(0, \frac{1}{2})$, $(\frac{\sqrt{7}}{4})$ and $(1, 0)$, and $\alpha = 2$.
Figure 2 shows a case with five nodes where BIP does not give the exact solution: Given are a sender s with coordinates $(0, 0)$, four destination nodes with coordinates $d_1 = (-4, 0)$, $d_2 = (-1, 0)$, $d_3 = (1, 0)$ and $d_4 = (4, 0)$, and $\alpha = 2$. The sketch shows the nodes and the relevant arcs labeled with p_{ij}.

In the first step, the BIP algorithm will decide to send from s to d_2 or d_3, thus reaching both of them requiring power 1. The incremental power needed to send from s to d_1 or d_4 is 15. Hence p_{21} and p_{34} equal 9, so BIP adds the arcs (d_2, d_1) and (d_3, d_4) in the third and fourth iteration, ending up with a total power requirement of 19. The optimal solution would be to send directly from s to all nodes requiring power 16.

The MIP algorithm is the above BIP algorithm with one additional phase to obtain a multicast tree for cases where $|D| < |N| - 1$. In this additional phase, tree links that do not lead to any destination are pruned from the broadcast tree. Node power levels are then updated (reduced) again as a result of this pruning.

Since MIP was presented, research on minimum-energy broadcasting and multicasting has been very active. Wan et al. [25, 26] examined the approximation ratios of the heuristics proposed by Wieselthier et al. They showed that the approximation ratio of the BIP algorithm is between 6 and 12. The authors of [4, 15] presented NP-completeness results for the minimum-energy broadcasting problem with arbitrary positive power as input. (The NP-completeness of MEMP follows then immediately.) They also provided a new heuristic and compared its performance to that of MIP. A proof of the NP-completeness result for the case of Euclidean metric space was provided in [11]. Most of research work devoted to algorithm design of MEMP have focused on heuristics [4, 5, 6, 8, 9, 10, 18, 19, 20, 21]. Some of these algorithms can be used for solving MEMP, whereas others are developed for the case of broadcasting. A majority of the algorithms are based on centralized computing, which requires some global knowledge of the network. Some authors, however, presented distributed, localized algorithms and protocols [4, 5]. A second type of solution approaches for MEMP are algorithms having performance guarantee in terms of approximation ratio [3, 15]. For more details of algorithms and communication protocols proposed for MEMP, we refer to the references mentioned in this paragraph as well as surveys of minimum-energy broadcasting and some related topics [14, 17, 23].

Another standpoint of studying MEMP is mathematical programming techniques. The authors of [7] proposed three integer programming models. The first formulation is a hop-indexed model similar to that for constrained MST developed by Gouveia (e.g., [12]). The second formulation is based on subtour elimination constraints. In the third formulation, connectivity is ensured by flow conservation con-

straints. The authors of [1] derived an integer programming model of set-covering type for MEMP. However, in view of the number of heuristic algorithms proposed for MEMP, mathematical programming models have been studied in a much less extent for this problem.

4 Integer Programming Models

Similar to the flow formulation presented by Das et al. in [7], our integer programming models use network flows to represent the communication paths. Therefore, we use the formulation by Das et al. as the starting point of our discussion. The flow formulation in [7] uses the follows sets of variables.

$P_i =$ The transmission power of node i,

$x_{ij} =$ The (total) network flow on link (i,j),

$y_{ij} = \begin{cases} 1 & \text{if link } (i,j) \text{ is used (included in the multicast tree),} \\ 0 & \text{otherwise.} \end{cases}$

The flow formulation in [7] is as follows.

$$\text{[F1]} \quad P^* = \min \sum_{i \in N} P_i \tag{2}$$

$$\text{s. t.} \quad \sum_{j:(s,j)\in A} x_{sj} = |D|, \tag{3}$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = -1 \quad \forall i \in D, \tag{4}$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \quad \forall i \in N \setminus D \setminus \{s\}, \tag{5}$$

$$x_{ij} \le |D| y_{ij} \quad \forall (i,j) \in A, \tag{6}$$

$$p_{ij} y_{ij} \le P_i \quad \forall (i,j) \in A, \tag{7}$$

$$x_{ij} \ge 0 \quad \forall (i,j) \in A, \tag{8}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in A. \tag{9}$$

The idea behind formulation F1 is that every node in the set D receives one unit of flow from the source node s. The total flow originated from s is thus $|D|$. These conditions are ensured by the flow conservation constraints (3), (4) and (5). Clearly, x_{ij} lies between 0 and $|D|$. Sending any positive amount of flow from i to j (i.e., $x_{ij} > 0$) means that link (i,j) is part of the multicast tree, that is, y_{ij} equals to 1. This relation is stated in (6). Finally, a constraint of (7) ensures that the transmission power of node i will be set to reach all the nodes in the transmission range chosen for i.

The numbers of variables and constraints in formulation F1 are both of $O(|N|^2)$. The main drawback of this formulation, however, is that its linear programming (LP) relaxation tends to be very weak. In other words, the formulation does not admit efficient computation of a sharp lower bound. As a consequence, applying a standard solver to this formulation results in a huge branch-and-bound enumeration tree. The computational complexity of finding the optimal solution grows very fast with respect to network size.

We propose several enhancements that lead to a formulation that allows us to, whenever the problem cannot be solved to optimality due to the size, compute a sharp lower bound to optimum. The first enhancement is to represent the flow in a disaggregate fashion. We introduce the following binary variables to distinguish between flows designated to different destination nodes.

$x_{ij}^d =$ The flow from s to destination d on link (i,j).

The corresponding flow conservation equations are

$$\sum_{j:(i,j)\in A} x_{ij}^d - \sum_{j:(j,i)\in A} x_{ji}^d = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = d \\ 0 & \text{otherwise.} \end{cases} \qquad \forall i \in N, \tag{10}$$

The purpose of using disaggregate flow is to improve the representation of the relationship between the flow variables and the set of variables that we introduce in the second enhancement. This second enhancement involves the way of describing the node power levels. We know that in an optimal solution, the transmission power $P_i = \max_{j\in N, j\neq i} p_{ij} y_{ij}$. This observation enables a more explicit representation of power levels than that used by (7). We introduce the following new set of binary variables, which will be used instead of the y-variables.

$$z_{ij} = \begin{cases} 1 & \text{if the power level of } i \text{ equals } p_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

We no longer need the P-variables to represent the total power. Instead, the objective function can be written as

$$\min \sum_{(i,j)\in A} p_{ij} z_{ij}. \tag{11}$$

Clearly, at most one of the possible power levels will be used by any node, resulting in the following set of inequalities.

$$\sum_{j:(i,j)\in A} z_{ij} \leq 1, \forall i \in N. \tag{12}$$

To send a unit of flow on link (i,j), the power level of node i must be at least p_{ij}. This gives us the following relationship between the flow variables and the z-variables.

$$x_{ij}^d \leq \sum_{k\in N: k\neq i, P_{ik}\geq p_{ij}} z_{ik} \quad \forall (i,j) \in A, \forall d \in D. \tag{13}$$

Unlike (6), constraints (13) do not contain any coefficient that is greater than 1. This significantly strengthens the LP.

The third enhancement is an extended version of (13). Observe that, because at most one unit of flow will leave any node to any destination, the left-hand-side of (13) can be extended to include all flow variables for which the corresponding link power is less than or equal to p_{ij}, that is,

$$\sum_{k\in N: k\neq i, p_{ik}\leq p_{ij}} x_{ik}^d \leq \sum_{k\in N: k\neq i, p_{ik}\geq p_{ij}} z_{ik} \quad \forall (i,j) \in A, \forall d \in D. \tag{14}$$

Putting the pieces together, our integer programming formulation can be summarized as follows.

$$\begin{aligned} \text{[F2]} \quad & P^* = \min \sum_{(i,j)\in A} p_{ij} z_{ij} \\ \text{s. t.} \quad & (10), (12), (14), \\ & x_{ij}^d \geq 0 \quad \forall (i,j) \in A, \forall d \in D, & (15) \\ & z_{ij} \in \{0,1\} \quad \forall (i,j) \in A. & (16) \end{aligned}$$

Both the number of variables and the number of constraints of formulation F2 are of $O(|N|^3)$. This is a magnitude higher than that of F1. However, this

increase in the complexity of formulation does pay off in the sense that, because the lower bound of the LP relaxation of F2 is magnitudes higher than that of F1, F2 scales much better than F1 when using a standard solver to solve MEMP exactly. We would like to strength the fact that the three enhancements are all crucial for obtaining this tight formulation of MEMP.

5 A Lagrangian Relaxation Scheme

For small networks, we can find the optimum of MEMP by applying a standard optimization software to any of the two models discussed earlier. For large-scale networks, finding the optimum is very difficult because of the problem complexity. For the purpose of performance evaluation of heuristics, an alternative is to use a good lower bound of optimum. Such a bound can be obtained from the LP relaxation of formulation F2. However, for networks with many nodes, the solution of the LP relaxation is also very time-consuming. Recall that the numbers of variables and constraints of F2 are both of order $O(|N|^3)$. For a network of, say, 100 nodes, the size of the resulting LP relaxation of F2 is out of reach of an state-of-the-art linear programming solver if we require a reasonable amount of computing time (e.g., a few hours). To overcome this difficulty, we propose a Lagrangian relaxation scheme for F2 with the aim of efficiently approximating the LP bound.

In the Lagrangian relaxation, the flow conservation constraints (10) are relaxed using Lagrangian multipliers $\lambda_i^d, i \in N, d \in D$. The resulting relaxation reads

$$L(\lambda) = \min \sum_{(i,j)\in A} p_{ij} z_{ij} + \sum_{d\in D}\sum_{i\in N} \lambda_i^d \Big(\sum_{j:(i,j)\in A} x_{ij}^d - \sum_{j:(j,i)\in A} x_{ji}^d \Big) + \sum_{d\in D} (\lambda_d^d - \lambda_s^d)$$

$$\text{s. t.} \qquad (12), (14), (15), \text{ and } (16).$$

Note that the second term of $L(\lambda)$ equals $\sum_{d\in D}\sum_{(i,j)\in A} c_{ij}^d x_{ij}^d$, where $c_{ij}^d = \lambda_i^d - \lambda_j^d$. For a set of fixed values of λ, it is clear that the relaxation decomposes into $|N|$ subproblems, one for every node. The subproblem for node i is as follows. (We omit the last, constant term of $L(\lambda)$ in the discussion of the subproblem.)

$$L_i(\lambda) = \min \sum_{j:(i,j)\in A} p_{ij} z_{ij} + \sum_{d\in D}\sum_{j:(i,j)\in A} c_{ij}^d x_{ij}^d \tag{17}$$

$$\text{s. t.} \qquad \sum_{j:(i,j)\in A} z_{ij} \le 1,$$

$$\sum_{k\in N: k\neq i, p_{ik}\le p_{ij}} x_{ik}^d \le \sum_{k\in N: k\neq i, p_{ik}\ge p_{ij}} z_{ik} \quad \forall j:(i,j)\in A, \forall d \in D,$$

$$x_{ij}^d \in \{0,1\} \quad \forall j:(i,j)\in A, \forall d \in D,$$

$$z_{ij} \in \{0,1\} \quad \forall j:(i,j)\in A.$$

We can use a simple, greedy procedure to solve the subproblem $L_i(\lambda)$. As at most one of the z-variables can be set to one, we can solve the subproblem by enumerating the set of possible power levels of node i, that is, setting $z_{ij} = 1$ for one j at a time and minimizing (17) in the flow variables. Due to the second constraint of the subproblem, at most one x-variable for any destination can be set to one. The same constraint enforces that, if $z_{ij} = 1$, then a variable x_{ik}^d can be set to one only if the power for reaching node k, p_{ik}, is less than or equal to p_{ij}. Among such variables, the one with the most negative value in the coefficient c_{ik}^d minimizes (17). Repeating this computation for all z-variables gives us the optimum of $L_i(\lambda)$. In mathematical

terms, for every j, we compute $\hat{p}_{ij} = p_{ij} + \sum_{d \in D} \sum_{k:k \neq i, p_{ik} \leq p_{ij}} \min\{0, c^d_{ik}\}$. Then, if $\min_{j \in N, j \neq i} \hat{p}_{ij} < 0$, the z-variable giving this minimum equals one, and the values of the flow variables follow immediately from our computations; otherwise it is optimal to set all the variables to zero.

For any assignment of the values to the multipliers, $\lambda = \bar{\lambda}$, solving the resulting relaxation yields a lower bound to the integer optimum P^*. To maximize the value of this bound (i.e., to solve the Lagrangian dual function), we apply a subgradient optimization procedure. Denoting an optimal flow solution to the Lagrangian relaxation by $\{\bar{x}^d_{ij}, (i,j) \in A, d \in D\}$, a subgradient of $L(\lambda)$ at $\bar{\lambda}$ is $\bar{\xi} = \{\bar{\xi}^d_i, i \in N, d \in D\}$, where

$$\bar{\xi}^d_i = \sum_{j:(i,j) \in A} \bar{x}^d_{ij} - \sum_{j:(j,i) \in A} \bar{x}^d_{ji} - b^d_i. \tag{18}$$

In (18), b^d_i is 1 if $i = s$, -1 if $i = d$, and 0 otherwise. The values of the multipliers are updated by taking a step t in the direction of the subgradient:

$$\bar{\lambda}' = \bar{\lambda} + t\bar{\xi}. \tag{19}$$

The step size t is computed using the following formula [24], in which $\bar{P}$ is an upper bound to the integer optimum P^*, and γ is a parameter whose value is between 0 and 2.

$$t = \gamma \frac{\bar{P} - L(\bar{\lambda})}{||\bar{\xi}||^2}. \tag{20}$$

To obtain good convergence in subgradient optimization, $\bar{P}$ should be as close to the optimum of the Lagrangian dual as possible. If $\bar{P}$ equals the dual optimum, γ should be set to 1. In our implementation of subgradient optimization, $\bar{P}$ is the total power of the multicast tree found by the MIP algorithm. The parameter γ is initially set to 1.0. Its values is then successively reduced by a scaling factor, whose value depends on the total number of subgradient optimization iterations. There exist more sophisticated rules for computing the step size and search direction in subgradient optimization, as well as other methods for maximizing the Lagrangian dual. We found, however, our implementation of subgradient optimization gives satisfactory results for the purpose of obtaining a sharp lower bound.

6 Performance Evaluation

We used networks of 10, 20, 50, and 100 nodes in our numerical experiments. The number of destination nodes varies from a few (a small multicast group) to $|N| - 1$ (broadcast). For every combination of $|N|$ and $|D|$, 100 networks were generated using the instance generation procedure by Wieselthier et al. [27, 28], and, for each network, we used two sets of transmission power for $\alpha = 2$ and $\alpha = 4$, respectively. In total, our performance evaluation involves 2800 networks instances, for which the following computations have been conducted.

- *The MIP algorithm.* The MIP algorithm has been applied to all network instances. In every step of the sweep procedure, the algorithm chooses the sweep operation that yields the largest reduction in total power, as explained in Section 3. The sweep procedure was applied iteratively until no further improvement can be made.

- *Integer programming solver.* All test networks of 10 and 20 nodes could be solved to optimality using an integer programming solver [13]. For 50-nodes

networks, most instances with few destinations ($|D| \leq 10$) permit an exact solution within a reasonable amount of computing time. For the rest of the test networks (more specifically, networks with $|N| = 50$ and $|D| \geq 25$, and networks with $|N| = 100$), the solver did not manage to find the optimal solution. In fact, even the LP relaxation of these networks becomes very hard to solve.

- *The Lagrangian relaxation scheme.* We applied the Lagrangian relaxation and subgradient optimization scheme to all test networks. The purpose of applying this solution approach to networks of small numbers of nodes and destinations, for which the optimal solutions are obtainable, is to examine the strength of the proposed relaxation. For large-sized networks, the Lagrangian scheme enables the performance evaluation of the MIP algorithm. The numbers of subgradient optimization iterations are 2000, 5000, 10000, and 50000 for networks of 10, 20, 50, and 100 nodes, respectively.

Our main results of performance evaluation are presented in Table 1 and Table 2. In Table 1, optimal multicast trees are used as benchmark in performance evaluation. The column 'I' shows the number of network instances for which optimal trees were obtained (using the integer programming solver). The columns 'MIP', 'LP', and 'Lag.' show, respectively, the total power of the trees constructed by the MIP algorithm, the lower bound found by the LP relaxation, and the lower bound given by our Lagrangian relaxation scheme. Each of these values is averaged over I network instances, and normalized with respect to the optimal tree, whose power is always 1.0.

Table 1: Performance evaluation using optimal multicast trees.

		$\alpha = 2$				$\alpha = 4$			
		I	MIP	LP	Lag.	I	MIP	LP	Lag.
$\|N\| = 10$	$\|D\| = 2$	100	1.1548	0.9995	0.9994	100	1.3585	1.0000	0.9998
	$\|D\| = 5$	100	1.1213	0.9988	0.9988	100	1.3325	1.0000	0.9998
	$\|D\| = 9$	100	1.0672	0.9979	0.9978	100	1.0185	0.9992	0.9991
$\|N\| = 20$	$\|D\| = 5$	100	1.1822	0.9965	0.9962	100	1.1643	0.9990	0.9988
	$\|D\| = 10$	100	1.2008	0.9892	0.9891	100	1.1587	0.9969	0.9964
	$\|D\| = 19$	100	1.1705	0.9811	0.9808	100	1.0329	0.9962	0.9953
$\|N\| = 50$	$\|D\| = 5$	100	1.3514	0.9862	0.9857	98	1.3302	0.9979	0.9975
	$\|D\| = 10$	95	1.2942	0.9685	0.9682	98	1.1929	0.9932	0.9928

Table 2 summarizes the results of evaluating the MIP algorithm using the Lagrangian relaxation scheme. For every combination of the three parameters, $|N|$, $|D|$, and α, the table displays the total power of the multicast tree found by the MIP algorithm averaged over 100 network instances. The power values are normalized with respect to the lower bound obtained by the Lagrangian relaxation scheme. The real performance of the MIP algorithm is therefore (slightly) better than those indicated in the table.

The results presented in Tables 1 and 2 give us indication of the average performance of the MIP algorithm as well as the proposed integer programming approach. Examining these results leads to the following observations.

- The numerical performance of the MIP algorithm is much better than its worst-case performance bounds [15, 25]. For network instances in Table 1, for example, the averaged tree power is always less than 36% away from optimum. However, the power of the multicast tree constructed by MIP diverges more

Table 2: Performance evaluation of the MIP algorithm using the Lagrangian relaxation scheme.

		$\alpha = 2$	$\alpha = 4$
$\|N\| = 50$	$\|D\| = 25$	1.2813	1.1233
	$\|D\| = 49$	1.2808	1.0728
$\|N\| = 100$	$\|D\| = 5$	1.4709	1.2998
	$\|D\| = 10$	1.4547	1.2024
	$\|D\| = 50$	1.3755	1.1169
	$\|D\| = 99$	1.3639	1.0771

from optimum when the number of nodes grows (and in particular for $\alpha = 2$). This algorithm behavior was anticipated because of the NP-hard nature of MEMP.

- The number of destinations $|D|$ has a significant impact on the performance of MIP. Note that this algorithm always starts by constructing a broadcast tree without accounting for the actual set of destinations, and, as a result, the algorithm performance improves as the number of destination nodes grows. This trend is particularly apparent for the network instances in Table 2. In other words, the MIP algorithm is more effective for broadcasting than for multicasting.

- A large value of α means a more rapid growth in transmission power with respect to distance, making it easier to identify and exclude non-optimal power levels. For most of our test networks, MIP achieved better results for $\alpha = 4$. This does not hold, however, for the first two rows in Table 1. A possible reason for this behavior is that for very small networks with few nodes and few destinations, the algorithm performance suffers seriously from the few non-optimal links for which the transmission power dominate the total tree power.

- Several authors have presented other heuristic algorithms that work differently than MIP [3, 8, 9, 10, 20]. These algorithms were developed for the broadcasting problem only. The performance improvement of these algorithms over MIP is roughly between 10 and 20 percent. In view of our results, we can conclude that the performance of these algorithms are near-optimal for $\alpha = 4$. For the other cases of MEMP, i.e., when $\alpha = 2$ or $|D| < |N|$, there is still quite much room left for improvement.

- The LP relaxation of our integer programming model yields a very good approximation to the integer optimum (the optimality gap is less than 2% for most of the networks in the first table). Moreover, the lower bound found using the Lagrangian relaxation technique is almost identical to that of the LP relaxation. Thus, our integer programming approach provides us with a very good tool for evaluating any heuristic algorithm for MEMP in large-scale networks.

For the multicast trees given by the MIP algorithm, Figures 3 and 4 show the distributions of normalized tree power within the range $[1.0, 2.0]$. These distributions give us some additional insights into the algorithm performance of MIP. In Figure 3, the value of 1.0 on the x-axes represents the power of the optimal tree, whereas in Figure 4 this value corresponds to the lower bound generated using the Lagrangian relaxation scheme. The curves show the percentage of network instances

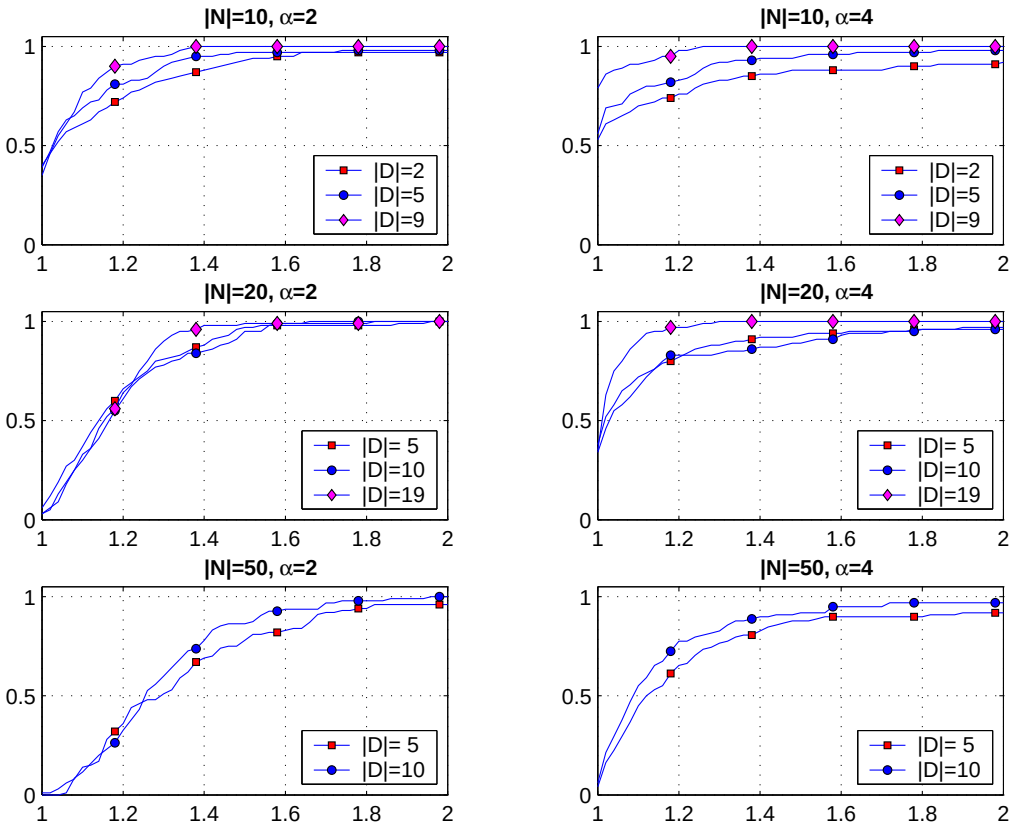

Figure 3: Tree power distribution of MIP with respect to optimal multicast trees.

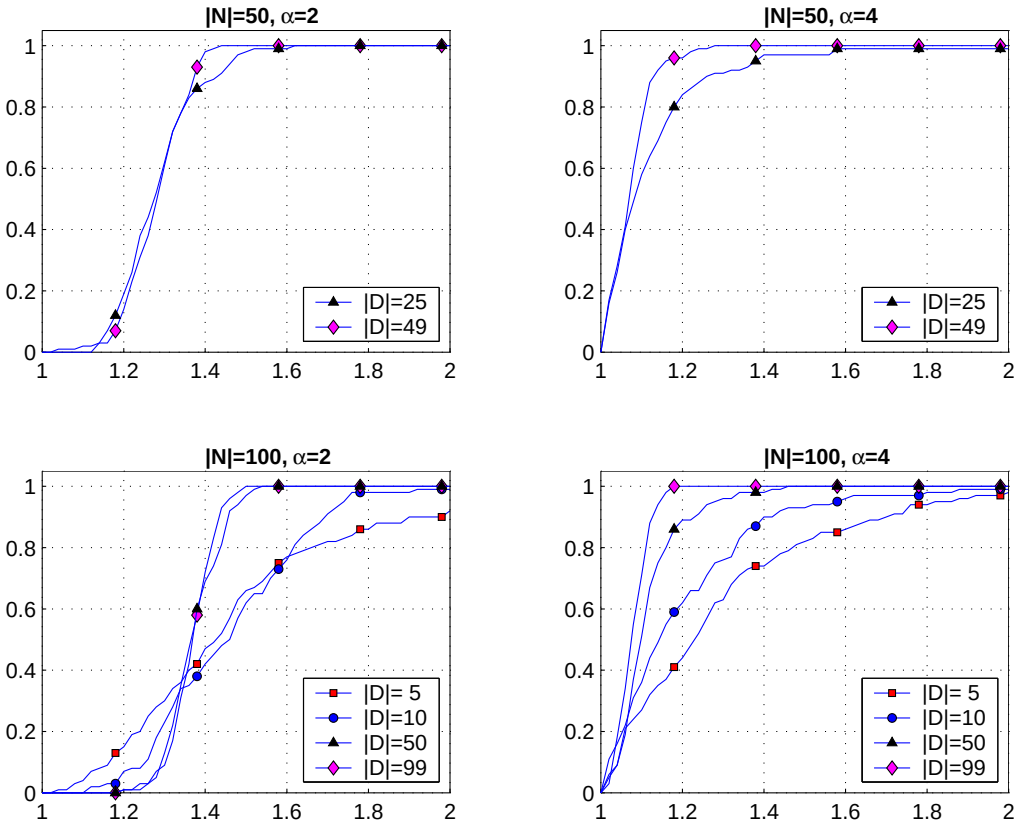

Figure 4: Tree power distribution of MIP with respect to the lower bound found by the Lagrangian relaxation.

for which the normalized tree power is less than or equal to the corresponding values of the x-axes. We make the following observations based on the two figures.

- The algorithm has a fairly good chance to find the optimal multicast tree for small networks. As shown by Figure 3, for all networks of 10 nodes, as well as for networks with 20 nodes and $\alpha = 4$, the tree constructed by the algorithm is optimal in over 40% of the cases.
- Comparing the figures to the results presented in Tables 1 and 2, we observe that for a majority of network instances the algorithm achieved better results than its average performance. This holds for all parameter settings of the test networks. In addition, for over 90% of the network instances, the algorithm managed to find a multicast tree whose power is at most twice as that of the optimal tree.
- When $\alpha = 4$, the algorithm is able to find near-optimal solutions for a large percentage of network instances. Indeed, for this value of α, over 40% of the trees are less than 20% away from optimum. The percentage of near-optimal trees is much lower for $\alpha = 2$, in particular for large-sized networks.
- Most of the plots in the two figures show performance improvement of the algorithm as the number of destination nodes increases. Noticeably, the percentage of test networks for which the algorithm found optimal or near-optimal trees is very low when $|N| \geq 50$ and when $|D|$ is small in relation to $|N|$. This gives motivation for further research on energy-efficient multicast routing algorithms for small multicast groups in large networks.

7 Concluding Remarks

In this paper we have evaluated the performance of a heuristic algorithm for minimum-energy broadcasting and multicasting in wireless ad hoc networks. Bounds on the minimum energy have been found through integer programming models, which for small cases are solvable to optimality. For larger instances, a Lagrangian relaxation provides lower bounds to the optimal power, and a subgradient algorithm is used to make the bounds as tight as possible. At best, these bounds are identical to the continuous (LP) relaxation bound.

Computational experiments demonstrate that the faster the power function grows with the distance (the larger the α), the more likely is it that the MIP algorithm finds the optimal solution. Emphasizing its preference for large α further, in cases where it is unable to locate the optimal solution, it gets closer if α is large. Also, the algorithm gives best results if a large proportion of the nodes are destination nodes. When only five out of 100 nodes are destination nodes, and $\alpha = 2$, the relative distance from optimality may in average over 100 cases be as large as 47%. This is however the case set for which the average performance is the poorest. The average distance from optimality is for no case set where $\alpha = 2$ as large as 36%.

Our integer programming model has proved to be very suitable for performance evaluation of heuristics for MEMP. For small-scale networks (no more than 20 nodes, or no more than 50 nodes and only ten of them destination nodes), the model is solved to optimality in reasonable time. In a large proportion of these cases, the continuous relaxation has integral optimal solutions. For larger networks, a lower bound on the minimum total power is given by this relaxation, and based on the observation from small cases there are reasons to believe that this bound is tight. When also the continuous relaxation requires too much computation time,

our Lagrangian relaxation scheme can be applied. By relaxing coupling constraints, we reduce the problem to a set of trivial problems. Putting this into a subgradient framework, we are able to compute bounds very close to the continuous relaxation bounds.

Although the flow formulation used to optimize the power assignments has some promising characteristics, its theoretical properties are not very much studied. Further research in order to find and analyze strong flow formulations is therefore needed. With the purpose of solving larger cases to optimality, integer programming models that are not flow-oriented can also be of interest. However, it cannot be expected that large-scale instances can be solved to optimality, and therefore the need for fast and effective heuristics is evident. The BIP and MIP approaches are very simple and intuitive. In forthcoming work we therefore intend to refine the ideas presented by these approaches, and hopefully achieve better approximations without paying too much in computational effort. Another line of future research is the generalization of our integer model for other energy-optimization problems in wireless ad hoc networks, such as message routing using a single broadcast tree (e.g., [22]) and various problems related to topology control (e.g., [16]).

References

[1] K. Altinkemer, F. S. Salman, P. Bellur, Solving the minimum energy broadcasting problem in ad hoc wireless networks by integer programming, in: Proceedings of the second workshop on modeling and optimization in mobile, ad hoc, and wireless Networks (WiOpt '04), April 2004, pp. 48–54.

[2] A. Balakrishnan, T. L. Magnanti, R. T. Wong, A dual-ascent procedure for large-scale uncapacitated network design, Operations Research 37 (1989), 716–740.

[3] M. Čagalj, J.-P. Hubaux, Energy-efficient broadcasting in all-wireless networks, Mobile Networks and Applications (MONET), in press.

[4] M. Čagalj, J.-P. Hubaux, C. Enz, Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues, in: Proceedings of ACM MobiCom '02, September 2002, pp. 172–182.

[5] J. Cartigny, D. Simplot, I. Stojmenovic, Localized minimum-energy broadcasting in ad-hoc networks, in: Proceedings of IEEE INFOCOM '03, April 2003, pp. 2210–2217.

[6] T. Chu, I. Nikoladis, Energy efficient broadcast in mobile ad hoc networks, in: Proceedings of the First International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW), September 2002, pp. 177–190.

[7] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, A. Gray, Minimum power broadcast trees for wireless networks: integer programming formulations, Proceedings of IEEE INFOCOM '03, April 2003, pp. 1001–1110.

[8] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, A. Gray, A cluster-merge algorithm for solving the minimum power broadcast problem in large scale wireless networks, in: Proceedings of IEEE MILCOM '03, October 2003, pp. 416–421.

[9] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, A. Gray, r-Shrink: A heuristic for improving minimum power broadcast trees in wireless networks, in: Proceedings of IEEE GLOBECOM '03, December 2003, pp. 523–527.

[10] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, A. Gray, e-Merge: A Heuristic for improving minimum power broadcast trees in wireless networks, Technical Report, Department of Electrical Engineering, University of Washington, WA, 2003.

[11] Ömer Eğecioğlu, T. F. Gonzalez, Minimum-energy broadcast in simple graphs with limited node power, Proceedings of IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS) '01, August 2001, pp. 334–338.

[12] L. Gouveia, Using hop-indexed models for constrained spanning and steiner trees models, in: B. Sansó, P. Soriano (Eds.), Telecommunications Network Planning, Kluwer Academic Publishers, 1999, pp. 21–32.

[13] ILOG, Ilog CPLEX 7.0, user's manual, August 2000.

[14] F. Ingelrest, D. Simplot, I. Stojmenovic, Energy-efficient broadcasting in wireless mobile ad hoc networks, in: M. Cardei, I. Cardei, D. Zhu (Eds.), Resource Management in Wireless Networking, Kluwer Academic Publishers, in press.

[15] 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 & Computing (MOBIHOC '02), June 2002, pp. 112–122.

[16] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, S. S. Ravi, Algorithmic aspects of topology control problems for ad hoc networks, in: Proceedings of the Third ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC '02), June 2002, pp. 123–134.

[17] X. Y. Li, I. Stojmenovic, Broadcasting and topology control in wireless ad hoc networks, in: A. Boukerche, I. Chlamtac (Eds.), Handbook of Algorithms for Mobile and Wireless Networking and Computing, CRC Press, in press.

[18] F. Li, I. Nikolaidis, On minimum-energy broadcasting in all-wireless networks, in: Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN '01), November 2001, pp. 193–202.

[19] S. Lindsey, C. S. Raghavendra, Energy efficient all-to-all broadcasting for situation awareness in wireless ad hoc networks, Journal of Parallel and Distributed Computing 63 (2003) 15–21.

[20] R. J. Marks, A. K. Das, M. El-Sharkawi, P. Arabshahi, A. Gray, Minimum power broadcast trees for wireless networks: optimizing using the viability lemma, Proceedings of the NASA Earth Science Technology Conference, June 2002, pp. 273–276.

[21] R. Montemanni, L. M. Gambardella, A. K. Das, The minimum power broadcast problem in wireless networks: a simulated annealing approach, Technical Report, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Manno, Switzerland, 2004.

[22] I. Papadimitriou, L. Georgiadis, Minimum-energy broadcasting in wireless networks using a single broadcast tree, in: Proceedings of the second workshop on modeling and optimization in mobile, ad hoc, and wireless Networks (WiOpt '04), April 2004, pp. 38–47.

[23] A. Pelc, Broadcasting in wireless networks, in: Handbook of wireless networks and mobile computing, I. Stojmenovic (Ed.), John Wiley and Sons, Inc., New York, 2002, pp. 509–528.

[24] B. T. Poljak, Minimization of unsmooth functionals, USSR Computational Mathematics and Mathematical Physics 9 (1969) 14–29.

[25] P.-J. Wan, G. Călinescu, X.-Y. Li, O. Frieder, Minimum-energy broadcast routing in static ad hoc wireless networks, in: Proceedings of IEEE INFOCOM '01, April 2001, pp. 1162–1171.

[26] P.-J. Wan, G. Călinescu, X.-Y. Li, O. Frieder, Minimum-energy broadcast routing in static ad hoc wireless networks, Wireless Networks 8 (2002) 607–617.

[27] J. E. Wieselthier, G. D. Nguyen, A. Ephremides, On the construction of energy-efficient broadcast and multicast trees in wireless networks, in: Proceedings of IEEE INFOCOM '00, March 2000, pp. 585–594.

[28] J. E. Wieselthier, G. D. Nguyen, A. Ephremides, Algorithms for energy-efficient multicasting in static ad hoc wireless networks, ACM Mobile Networks and Applications (MONET) 6 (2001) 251–263.