Probabilistic Optimization for Energy-Efficient Broadcast in All-Wireless Networks

Fulu Li
The Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA USA 02139
Email: fulu@mit.edu

Shie Mannor
Dept. of Electrical and Computer Engineering
McGill University
Montreal, Quebec, Canada H3A 2A7
shie@ece.mcgill.ca

Andrew Lippman
The Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA USA 02139
lip@media.mit.edu

Abstract

We study optimization methods for source-initiated energy-efficient broadcast in all-wireless networks. The minimum energy broadcast (MEB) problem in wireless networks proved to be NP-complete [1,4]. Past studies on MEB problem [1,2,5,8,9] focus on *deterministic* optimization to construct energy-efficient broadcast tree. In this paper, we present a *probabilistic* optimization approach, the Randomized Tree Optimization (RTO) algorithm, which is based on the cross-entropy (CE) method introduced by Rubinstein et al in [3,6,7]. The experimental results indicate that the RTO method obtains the best performance compared to state-of-the-art methods.

Keywords: Wireless Networks, Optimization, Broadcast, Energy Efficiency;

I. Introduction

In this paper, we focus on the construction of the energy-efficient broadcast tree. The motivation of this work is the importance of energy efficiency in wireless networks. The lifetime of a wireless network is limited due to the limited power capacity of the energy sources at each node such as batteries. The broadcast tree is rooted at the source node and should reach all of the desired destination nodes. Following [9], we consider a wireless ad-hoc network in which the node locations are fixed and the channel conditions are unchanged. The situation with the mobility of the nodes in the construction of broadcast tree can be addressed by adjusting the transmitter power to accommodate the new locations of the nodes, which is a topic for future studies. We also assume that the power level of a transmission can be chosen within a given range of values and the use of *omni-directional* antennas. Thus, all nodes within communication range of a transmitting node can receive its transmission.

Moreover, we assume that sufficient bandwidth resources and ample transceiver resources are available at each node. The energy consumption between the transmitting node and the receiving node is *not* linear because of the nonlinear attenuation properties of radio signals. *Due to the non-linear path loss model of the transmission power, relaying information between nodes may lead to lower power attenuation than communication directly over large distances.*

Our major contribution is the presentation of the random tree optimization algorithm, which is based on the cross-entropy method [3,6,7]. The basic idea behind the cross entropy method is to translate the *deterministic* optimization problem into a related *stochastic* one and then use rare event simulation techniques to find the solution. We will show that the presented method obtains the best performance comparing to state-of-the-art heuristics for the MEB problem.

The rest of the paper is organized as follows. We discuss related work in Section II. We give the problem formulation in Section III. We present the RTO algorithm in Section IV. Experimental results are presented in Section V. Conclusions and future directions are given in Section VI.

II. Related Work

The minimum-energy broadcast (MEB) problem in wireless networks has been addressed in several recent studies [1,2,5,8,9]. Several interesting heuristics were proposed in [9], which can be roughly divided into two categories: link-based approaches and node-based approaches. As the names suggest, link based approaches, the BLU (Broadcast Least-Unicast-cost) algorithm and the **MST** (broadcast link-based Minimum-cost Spanning Tree) algorithm, use the link-based costs and further the shortest unicast paths and spanning trees. The node-based approach, i.e., the **BIP** (Broadcast Incremental Power) algorithm, constructs the broadcast tree incrementally in the sense that new nodes are added to the tree one at a time on a minimum incremental cost basis until all nodes are included in the tree. Compared with the link-based approaches, BIP better exploits the wireless multicast advantage in the construction of the broadcast tree and demonstrates better performance than BLU and MST [9]. However, due to the incremental nature of BIP, it lacks a global view of all the nodes and it can not *fully* utilize the wireless broadcast advantage to reduce the total required power of the broadcast tree further. In [8], Wan et al presents a quantitative and in-depth analysis on the approximation ratios of the heuristics presented in [9].

In [5], Li et al proved that the MEB problem is NP-hard for the general case. Most recently, Cagalj et al in [1] presented a systematic proof on the NP-completeness of the MEB problem both for the general case and for the geometric one. A heuristic called **EWMA** (Embedding Wireless Multicast Advantage) was also suggested in [1]. In EWMA, a *distributed* MST (Minimum-Spanning Tree) algorithm is executed in the first phase to obtain a feasible broadcast tree. In the second phase of EWMA, the so-called *local* EWMA is executed at each node in order to exclude some transmitting node from its neighbors to reduce the total required power of the tree by broadcasting or re-broadcasting messages based upon the calculation of the overlapping set for a sender.

Das et al in [2] presented several different integer programming formulations of Minimum Power Broadcast (**MPB**) trees for wireless networks in order to achieve the optimal solution. The number of variables and constraints are at least in the order of $O(N^2)$, where N is the number of nodes in the network. The basic idea is to solve a linear relaxation of the problem first, if the solution is integer, then the algorithm terminates with an optimal solution. If the solution is not an integer it creates two sub-problems and branches down on a fractional variable until no active sub-problem exists any more. The downside of this approach is that problems with as few as **20** nodes can be practically intractable unless they demonstrate some simplified structures.

In Section IV, we propose a probabilistic optimization approach, the Random Tree Optimization (**RTO**) algorithm, which is based on the cross-entropy method [3,6,7], to tackle the MEB optimization problem from a different angle. Our experimental results in Section V indicate that the RTO method achieves the best results comparing to other existing methods.

III. Problem Formulation

3.1 The System Model

We consider an all-wireless network where numerous devices, i.e., nodes, that are equipped with micro-processor, memory, sufficient bandwidth and transceiver resources, limited power supply such as batteries are linked via short-range radio connections. We study the problem of source-initiated broadcast through which data are disseminated to each of the intended destination node in the network. As stated in Section I, we assume the use of *omni-directional* antennas and all nodes within communication range of a transmitting node can receive its transmission.

3.2 The Power Consumption Rules

We consider a commonly used wireless propagation model [1,2,5,8,9] whereby the received signal power attenuates $d^{-\lambda}$, where d stands for the distance between the transmitting node antenna and the receiving node antenna and λ takes a value between 2 and 4, depending on the characteristics of the communication medium.

Regarding the transmission energy, we have the following definitions:

Definition 3.1: The power required for a transmitting node, say T, to *directly* reach a set of destination nodes, say $D_1, D_2, ..., D_m$, is determined by the *maximum* required power to reach any of them individually. *For the sake of brevity, throughout this paper we will use d^{λ} to stand for the required power for a transmitting distance of d*. Let $d_1, d_2, ..., d_m$ stand for the distances from the transmitting node T to the destinations $D_1, D_2, ..., D_m$, respectively. The required power is determined by:

$$p_{req} = \max(d_1^{\lambda}, d_2^{\lambda}, ..., d_m^{\lambda}) . \qquad (1)$$

Definition 3.2: The power required for a broadcast tree is the sum of the energy required for each of the transmitting nodes in the tree. Let $S, T_1, T_2, ..., T_r$ denote the transmitting nodes in the given tree (S is the source node and $T_1, T_2, ..., T_r$ are the relaying nodes). Let $p_S, p_{T_1}, ..., p_{T_r}$ denote the required power for those transmitting nodes respectively. The total required power for this tree is given by:

$$p_{tree} = p_S + \sum_{i=1}^{r} p_{T_r} . \qquad (2)$$

We illustrate this by the following example.

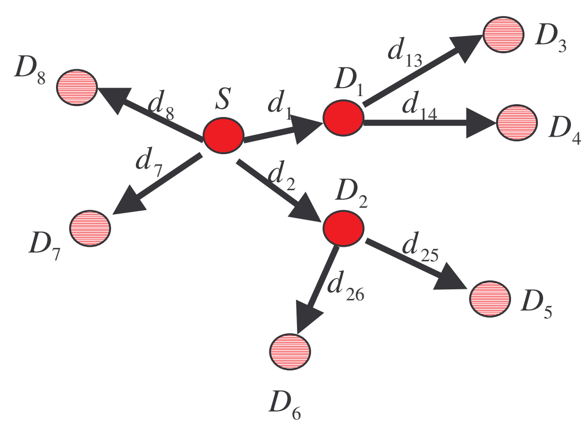

Figure 1: An example of a nine-node broadcast tree, rooted from S with destinations $D_1 \sim D_8$.

In Figure 1, the source node is S and the destination nodes are $D_1 \sim D_8$. In the above broadcast tree, the transmitting nodes are S, D_1 and D_2, i.e., the source node S and two relaying nodes D_1 and D_2. As stated in Definition 3.2, the total required power for the broadcast tree is only related these transmitting nodes. From Formula (1) and (2), the total required power for this broadcast tree can be expressed by:

$$p_{tree} = p_S + p_{D1} + p_{D2}$$
$$= \max(d_1^{\lambda}, d_2^{\lambda}, d_7^{\lambda}, d_8^{\lambda}) + \max(d_{13}^{\lambda}, d_{14}^{\lambda}) + \max(d_{25}^{\lambda}, d_{26}^{\lambda})$$

In summary, the problem can be stated as *how to construct a broadcast tree such that the total required energy is minimal.*

IV. The RTO Algorithm

The Cross Entropy (**CE**) method, (see, e.g., Rubinstein et al [3,6,7]) is a useful meta-heuristic optimization method for finding near-optimal solutions in a variety of combinatorial optimization problems. The basic idea is to translate the deterministic optimization problem into a related stochastic optimization one and then use Rare Event Simulation (**RES**) techniques to find the solution.

We call the specification of the CE method to MEB problem the Random Tree Optimization (**RTO**) algorithm. We will see in the following that the algorithm operates iteratively by *randomly* generating *improved* sample trees based on a transmission probability matrix until the optimization process converges based on our predefined performance function, i.e., the total required power of the tree.

First, we define the performance function $F(tree)$ as the total required power of a tree, which is given in Definition 3.2 (see Equation 2). There are two key components in the RTO process based on CE: (1) Generation of *random* sample trees; (2) Update of the parameters at each iteration. The update mechanism is supposed to encourage trees with high performance so that the randomization mechanism would lead to trees with even better performance.

We use a Markov chain that starts at the root node and stops after all of the destination nodes are reached to construct a sample tree. We define $Q = (q_{i,j})_{((N+1)\times(N+1))}$ as the one-step transition matrix, where $q_{i,j}$ denotes the probability that there is a transmission from node i to node j. We can observe that the sum of each column in the matrix has to be *one* as each destination has to be reached with *certainty*.

In order to find local neighbors first to avoid some "far-reached" transmissions to dominate the total required power of the tree, which could potentially degrade the performance of the RTO optimization process, the initial transition matrix Q_0 can be set as follows: (a) the column corresponding to transmissions to the root node in the matrix and the diagonal elements are set to *zero* as no node transmits to itself and no node transmits to the root node; (b) for other elements $q_{i,j}$, we have

$$q_{i,j} = \frac{1/(d_{i,j}+c)^{\lambda}}{\sum_{j:(i\neq j)} \frac{1}{(d_{i,j}+c)^{\lambda}}}, \quad (3)$$

where c is a constant that is related to the diameter of the network. When c is zero, the transition probability $q_{i,j}$ is solely determined by the distance between node i and node j. When c is much bigger than the diameter of the network, the transition probabilities in each column of the matrix are almost uniform. We will discuss this issue further in section V.

In the following, we give a brief description of the random tree generation algorithm:

4.1: Random Tree Generation for RTO

The random tree generation algorithm proceeds by randomly choosing a parent node based on the transition probability matrix for a given non-parented node (except the root node) among its non-descendent nodes till each destination node has a parent node.

We now turn our attention to the update algorithm. At each iteration of the RTO algorithm based on the CE method, we need to calculate the benchmark value of γ_t as follows:

$$\gamma_t = \min\{f : P_{Q_{t-1}}(F(T) \le f) \ge \rho\}, \quad (4)$$

where ρ normally takes a value of 0.01 so that the event of obtaining high performance is not too rare, $F(T)$ stand for the total required power of a randomly-generated sample tree, say T, based on the one-step transmission probability matrix in the $(t-1)^{th}$ round, e.g., Q_{t-1}, $P_{Q_{t-1}}(A)$ denote the probability of the event A conditioned on Q_{t-1}. Essentially, γ_t is the sample ρ-quantile of the performance of the randomly generated trees in the t^{th} round.

There are several choices to set the termination conditions. Normally, If for some $t \ge l$, say $l = 5$,

$$\gamma_t = \gamma_{t-1} = \ldots = \gamma_{t-l}, \quad (5)$$

then stop the optimization process.

The updated value of $q_{i,j}$ can be estimated as:

$$q^e_{i,j} = \frac{\sum_{k=1}^{M} H_{\{F(T_k)\le\gamma\}} H_{\{T_k \in T_{i,j}\}}}{\sum_{k=1}^{M} H_{\{F(T_k)\le\gamma\}}}, \quad (6)$$

where M stands for the number of sample trees, $H_{\{\}}$ is an indicator function, $T_{i,j}$ denotes the set of trees in which there is a transmission from node i to node j. While there are solid theoretical justifications for Equation (6), we refer the readers to [3,6], and focus on the algorithms that were implemented in practice. In order to avoid overly quick convergence to 1s and 0s for the update of $q_{i,j}$, which could limit the randomness of the sample trees, normally we use a *smoothed* update procedure in which

$$q^t_{i,j} = \alpha \times q^e_{i,j} + (1-\alpha) \times q^{t-1}_{i,j}, \quad (7)$$

where $q^{t-1}_{i,j}$ is the value of $q_{i,j}$ in the previous round and $q^e_{i,j}$ is the estimated value of $q_{i,j}$ based on the performance in the previous round according to Equation (6), and $q^t_{i,j}$ stands for the value of $q_{i,j}$ for the current round. Empirically, a value of α between $0.4 \le \alpha \le 0.9$ gives the best results as in [6].

In summary, we have a brief description of the RTO algorithm as follows:

4.2: RTO Algorithm based on CE method

1. *Set $t = 1$ and set Q_0 according to the init. of $q_{i,j}$ in Equation (3).*
2. *Randomly generate sample trees (typically $20N^2$ sample trees).*
3. *Calculate γ_t according to Equation (4).*
4. *Update $q_{i,j}$ according to Equation (6) and Equation (7).*
5. *If for some $t \ge l$, say $l = 5$, such that $\gamma_t = \gamma_{t-1} = ... = \gamma_{t-l}$, then stop; otherwise, reiterate from step 2.*

As discussed before, the complexity of the random tree generation algorithm is in the order of $O(N^3)$ and typically we generate $20N^2$ sample trees for each round. Therefore, the overall computation complexity of the RTO algorithm is in the order of $O(N^5)$. For a large value of N, the RTO process could be slow. We use the following strategies to speed up the RTO algorithm: (a) Ignore small transition probabilities, say for $q_{i,j} < 0.01$, we force them to be zero and renormalize it for each column of the transition probability matrix. (b) Set a looser termination criterion. Normally, we set $l = 5$ for the termination condition according to Step 5 of the RTO algorithm. When N is large, we opted to set $l = 2$. We also use a fast localized greedy routine (see [5] for details) to further optimize the tree generated by RTO.

In the following section we present extensive experiments that evaluate the performance of the RTO algorithm.

V. Performance Evaluation

Following [9], we simulate networks of a varying number of nodes, N, placed randomly within a 5×5 plane, in a variety of circumstances, e.g., with the power attenuation factor, λ=2, 3, 4. For the performance comparison between the algorithms, we consider *normalized tree power*. For example, suppose we have three approaches to generate broadcast trees, say approaches A, B and C. Let p_A, p_B and p_C stand for the required tree power for the trees generated by approaches A, B and C, respectively for the *same* network topology. The normalized tree power for each of these approaches is given by the following:

$$p'_A = \frac{p_A}{\min(p_A, p_B, p_C)},$$

$$p'_B = \frac{p_B}{\min(p_A, p_B, p_C)},$$

$$p'_C = \frac{p_C}{\min(p_A, p_B, p_C)}. \quad (8)$$

As reported in [9], the performance of BLU approach is always much worse than BIP and MST. We therefore only compare the performance of RTO against that of BIP and MST.

In the following, we examine the dynamics of the RTO algorithm and its performance compared with existing approaches in a variety of setups.

Table 1 shows the average power saving ratio of the broadcast trees generated by RTO compared with BIP for the same network topologies with different number of nodes and different power attenuation factor values. The results are averaged over 50 random network instances. Clearly, the RTO algorithm outperforms BIP across all setups. Notably, the power saving are above 15% when power attenuation factor is 2.

	$\lambda=2$	$\lambda=3$	$\lambda=4$
N=20	15%	8%	6%
N=40	19%	9%	7%
N=60	16%	6%	6%

Table 1: Average power saving ratio of the normalized tree power of the broadcast trees generated by RTO compared with BIP.

Table 2 demonstrates the average power saving ratio of the broadcast trees generated by RTO compared with MST for the same network topologies for three values of λ. Clearly, RTO outperforms MST in all circumstances, in particular when λ equals 2, the average power saving is above 25%.

	$\lambda=2$	$\lambda=3$	$\lambda=4$
N=20	28%	14%	8%
N=40	26%	15%	9%
N=60	27%	9%	9%

Table 2: Average power saving ratio of the normalized tree power of the broadcast trees generated by RTO compared with MST.

In order to provide some more insight into the inner working of the RTO algorithm, we show the evolution of the transition probability matrix for a single run of the RTO algorithm in Figure 1. We randomly generated *N=15* nodes, and set λ to 2. The initial sequence of the $q_{i,j}$ s is based on Equation (3), with c set to zero.

In the Figure 2, we plot a histogram of all the $q_{i,j}$ s for different iterations during the RTO process. The y-axis denotes the one-step transition probability. The x-axis stands for the sequence of the elements in the probability matrix and we have a total of 15×15 = 225 (*N=15*) elements in the matrix. As typical to the CE method, the transition probabilities quickly converges with some transition probabilities converging to one and others to zero. Essentially, transitions that lead to good solutions are reinforced and transitions that lead to poor solutions become smaller.

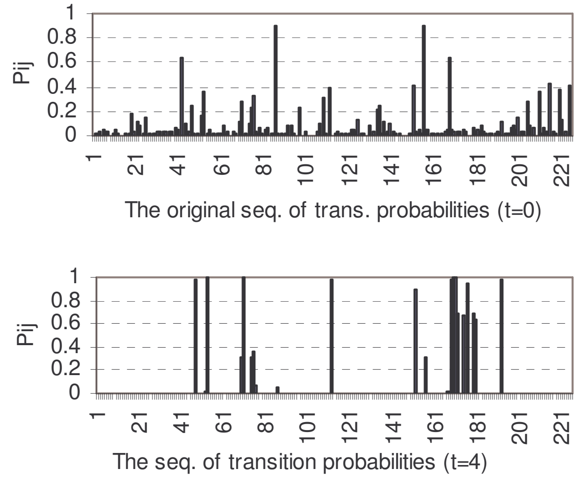

Figure 2: An illustration of an example run of the evolving of the transition probability matrix (N =15, λ =2) .

The empirical results with varying number of nodes in the network are shown in Figure 3. From Figure 3, we can observe that when the number of nodes in the network is less than 20, RTO performs slightly better compared with the case for a larger number of N . This could result from several factors: firstly, we choose to generate $20N^2$ sample trees for each round and this could certainly give some advantage for an N smaller than 20. If we choose to generate N^3 sample trees, we expect that the curve could be flat but the algorithm would be slower. Secondly, we use a loose termination criterion for a large N to speed up the RTO process and this could also cost some performance gains.

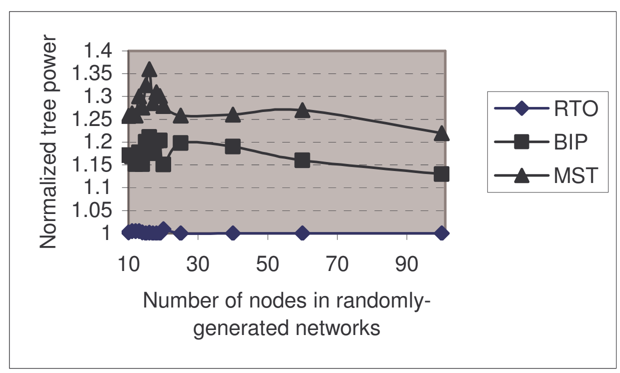

Figure 3: Average normalized tree power by RTO, BIP and MST (average over 60 random network instances for networks with a fixed number of nodes) with varying number of nodes in the network and adaptive transition probability matrix initialization for RTO. The value of λ is equal to 2 in this set of experiments.

In Table 3 and Table 4, we provide the normalized tree powers by RTO, BIP and MST for 30 randomly-generated network instances with different values of the transition probability update smoothing factor α .

The number of nodes in the network, i.e., N , is 15, and the value of λ is 2. For the results reported in Table 3, the value of α is **0.8**. For the results reported in Table 4, the value of α is **0.7**. In Table 3, the average normalized tree power for RTO, BIP and MST are 1.0, 1.20 and 1.325, respectively. The *standard deviation* for RTO, BIP and MST are 0.0, 0.12 and 0.15, respectively. In Table 4, the average normalized tree power for RTO, BIP and MST are 1.0006, 1.23 and 1.31, respectively.

It can be observed that RTO outperforms the other two algorithms in terms of both average performance and low *standard deviation*. By examining the values in Table 3 and Table 4, we observe that RTO significantly outperforms BIP and MST for N = 15 and λ = 2. In some cases, RTO saves as much as 60% to 90% power compared with BIP and MST.

	MEAN NORMALIZED TREE POWER	STANDARD DEVIATION
RTO	1.0	0.0
BIP	1.20	0.12
MST	1.325	0.15

Table 3: Mean normalized tree power by RTO, BIP and MST over 30 randomly generated networks. The number of nodes in the network, i.e., N , is 15, the value of λ is 2 and the value of α is **0.8**.

	MEAN NORMALIZED TREE POWER	STANDARD DEVIATION
RTO	1.0006	0.0029
BIP	1.23	0.19
MST	1.31	0.17

Table 4: Mean normalized tree power by RTO, BIP and MST over 30 randomly generated networks. The number of nodes in the network, i.e., N , is 15, the value of λ is 2 and the value of α is **0.7**.

VI. Conclusion and Future Directions

We considered energy-efficient broadcast in all-wireless networks since energy-efficiency is an important consideration for the design of wireless communication protocols due to the fact that the lifetime of a wireless network depends on the power consumption of each node, each of which is normally equipped with a limited power supply such as batteries. This problem has been introduced in [9] and it has received much attention in some recent studies [1,2,5,8,9].

The major contribution of this work is the development of the RTO algorithm, which is based on the cross-entropy method ([3,6,7]). We proposed a random tree generation algorithm based on the transition probability matrix and explored efficient ways to initialize the probability matrix in different circumstances. We conducted extensive experiments to examine the performance of RTO compared with other existing approaches. Our empirical results indicate that it demonstrates the best performance of its kind.

We plan to consider other techniques for speeding-up the RTO algorithm. An interesting direction we plan to pursue is to implement the RTO algorithm in a distributed fashion, so that the nodes are divided into groups, on which sub-trees are generated separately and then merged to a single tree.

Acknowledgement

The authors would like to thank the Digital Life Consortia at MIT Media Lab for the support.

Reference

[1] Cagalj M., J. Hubaux, C. Enz, “Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues”, *in the proc. of ACM MOBICOM ‘2002*.

[2] Das, A., Marks, R., El-Sharkawi, M., Arabshahi, P., Gray, A., “Minimum Power Broadcast Trees for Wireless Networks: Integer Programming Formulations”, in the Prof. of IEEE INFOCOM ‘2003.

[3] De Boer P., D.P. Kroese, S. Mannor, R.Y. Rubinstein, “A Tutorial on the Cross-Entropy Method”, in *Annals of Operations Research*, 2005.

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

[5] Li F., I. Nikolaidis, “On Minimum-Energy Broadcasting in All Wireless Networks”, *in the proc. of IEEE LCN’2001*, pp. 193-202.

[6] Rubinstein R., “The Cross-Entropy Method for Combinatorial and Continuous Optimization”, Methodology And Computing in Applied Probability, 1999.

[7] Rubinstein R., D. Kroese, “The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning”, Springer, 2004.

[8] Wan P., G. Calinescu, X. Li, O. Frieder, “Minimum-Energy Broadcast Routing in Static Ad-hoc Wireless Networks”, *in the proc. of IEEE INFOCOM ‘2001*.

[9] Wieselthier J., G. Nguyen, A. Ephremides, “On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks”, *in the proc. of IEEE INFOCOM ‘2000*, pp. 585-594.