DMIP3S: Distributed Algorithms for Power-Conserving Multicasting in Static Wireless Ad Hoc Networks

Dazhen Pan Purnima Mavinkurve Hung Q. Ngo Vikas Verma Amit Chandak
Computer Science and Engineering Department,
201 Bell Hall
State University of New York at Buffalo,
Amherst, NY 14260, USA.
{dpan, pmm24, hungngo, vpverma, achandak}@cse.buffalo.edu

***Abstract*— In a static wireless ad hoc network, given a distinguished source node, and a subset of nodes called multicast group members, the minimum-energy multicast problem is to assign appropriate power levels to nodes in the network so that all group members are reachable from the source, and the total power usage is as small as possible. Computing effective power assignments and the multicast trees in a distributed manner is a major challenge.**

In this paper, we devise two distributed algorithms for this problem. The central idea is called *Incremental Power with Potential Power Saving* (IP3S). Extensive simulations show that our distributed algorithms work even better than most known centralized ones. As far as we are aware of, these are the first distributed algorithms for this problem.

***Index Terms*—Power conservation, Multicast, Ad hoc networks.**

I. Introduction

Energy conservation in ad hoc networks is of paramount importance. Moreover, in many typical ad hoc network applications such as mobile conferencing, emergency services, and battle field communications, multicasting and broadcasting are the most natural communication primitives. Multicasting is a fundamental generalization of both unicasting and broadcasting. Consequently, the problem of devising energy-efficient multicast algorithms and protocols are of tremendous importance in ad hoc wireless networking. This problem and the corresponding broadcast problem have received special attention from various researchers in recent years [**?**, **?**, 1–9].

All the works cited above and this paper deal with static ad hoc wireless networks. Mobility adds a whole new dimension to the problem. We hope to extend our ideas to cope with it in the future.

The power-aware multicast problem we are facing is assumed to be *source initiated* (as opposed to *topology control oriented* [10]), namely some distinguished node among a set of given nodes initiates the process of finding a power assignment vector which indicates the transmission power level at each node in the network. Also, we assume *omnidirectional antennas* are used so that given a power assignment at a node, say u, all nodes within a certain radius of u are reachable from u in a single hop. The power assignment needs to satisfy the condition that all nodes in the multicast group are reachable from the source node, possibly in several hops. The problem is to find such a power assignment vector with minimal total power.

The problem of finding an energy-optimal power assignment, for either the broadcast or multicast version, is $\mathbf{NP}$-hard, as repeatedly shown by many authors [6, 11].

This paper addresses the aforementioned challenge by proposing two distributed algorithms for the power-conserving multicast problem. To the best of our knowledge, no prior work has addressed the distributed version of the multicast problem. One algorithm, called **DMIP3S**, is for small multicast groups and the other one called **P-DIP3S** is for larger groups. Extensive simulation results show that our distributed algorithms outperform most known centralized algorithms to date.

Our algorithms are based on a critical and interesting idea called *incremental power with potential power saving* (**IP3S**), which can be applied to any algorithms that "grow" the multicast tree from the source.

The rest of this paper is organized as follows. Section II formally describes the problem and related notions. Section III reviews related works. Section IV describes our main ideas and algorithms, while section V gives simulation results which show that the algorithms work better than most known centralized algorithms. Lastly, Section VI gives concluding remarks.

II. Preliminaries

A. Communication model

We assume that all nodes are equipped with omnidirectional antennas with continuously adjustable power levels. If the power levels are discrete, we can always "round" an assigned power level to the closest available level and the algorithms' performances are not changed by much if the granularity of the power level set is fine enough [5].

The most common, admittedly simplistic, attenuation model assumes that signal power falls proportional to d^α, where d is the signal traveling distance, and α is an environmentally dependent real constant between 2 and 4 [3, 12].

Suppose a node u is transmitting with power $p[u]$. A node v of distance d_{uv} from u can properly receive the signal from u if $p[u] \geq \gamma d_{uv}^\alpha$. Here, γ represents the receiver's power threshold for signal detection, often normalized to be 1. Thus, from here on we assume that v can properly receive u's signal iff $p[u] \geq d_{uv}^{\alpha}$.

We also assume, for simplicity of presentation, that nodes are homogeneous with respect to their power level sets or their p_{max}. Our algorithms work in the same way when network nodes are heterogeneous.

*B. The minimum energy consumption multicast subgraph (*MECMS*) problem*

MECMS is the problem we address in this paper. Let $G = (V, E)$ be a directed graph with a symmetric cost function $c : E \to \mathbb{R}^+$ on its edges, namely $c(u, v) = c(v, u), \forall (u, v) \in E$. A distinguished vertex $r \in V$, called the *source node*, and a subset $R \subseteq V - \{r\}$, called the *multicast group members*, are given. A *power assignment vector* is a function $p : V \to \mathbb{R}^+$, which assigns to each node of G some "power level." The *reachability graph* $G_p = (V, E_p)$ given a power assignment vector p is a graph on the same set V of vertices, such that there is an edge (u, v) from u to v in G_p if and only if $p[u] \geq c(u, v)$. That is, the power assigned to node u is at least the cost to reach vertex v.

We are to find a power assignment vector p such that there is a directed path, in G_p, from the source r to every node in the multicast group R, so as to minimize the sum $\sum_{v \in V} p[v]$.

III. RELATED WORKS

The easiest approach to build a multicast tree is to *prune* a broadcast tree, in much the same way the Internet multicast protocols work (DVMRP, PIM, etc.). Several algorithms have been proposed to build such a broadcast tree: MST (minimum spanning tree), BIP [3] (Broadcast Incremental Power), EWMA [6] (Embedded Wireless Multicast Advantage), SPT [3] (Shortest Path Tree), RBOP [9] (relative neighborhood graph broadcast oriented protocol). All these algorithms are centralized except for RBOP. In [?], we presented a distributed algorithm for the broadcast version of the problem, called DIP3S, which works better than RBOP.

We will append a prefix "P-" before a broadcast algorithm's name to denote the multicast algorithm based on pruning. (For example, P-MST is the pruning version of MST.)

Wieselthier et al. [3] have experimented the P-BIP, P-SPT, P-MST heuristics. Wan et al. [7] analytically showed that in the worst case P-MST, P-SPT, and P-BIP have linear performance ratios in terms of the number of nodes in the network. Hence, they do not perform well theoretically. In the same paper, the authors also proposed an analog of MST for the multicast case, called the *shortest path first* (SPF) algorithm which we will further explain in section IV. Readers are referred to [8] for a detailed survery on various algorithms.

In a recent paper [?], we proposed the MIP3S algorithm for the multicast version of the problem, MIP3S is centralized. By simulations, it was shown to outperform all known algorithms.

To the best of our knowledge, there has been no proposal for a distributed algorithm on MECMS problem. We address this challenge in this paper.

IV. THE ALGORITHMS

In this section, we first present the IP3S idea, which could be applied to many different algorithms, as long as a simple invariance is maintained. We then discuss our distributed algorithms based on this idea.

A. The IP3S idea

In many known algorithms, a central theme has been to iteratively "grow" a set S of covered nodes starting from the source, i.e. initially $S = \{r\}$. At each step, a node, say v, in S is chosen according to some objective function and $p[v]$ is expanded to cover one or more new nodes. The new nodes are then added into S, until all nodes in the multicast (broadcast) group are in S. The invariance is to keep S as the set of nodes reachable from the source r with the current power assignment vector.

The IP3S idea is a nice observation applied to this crucial "expansion step" of an algorithm. Consider a potential expan-

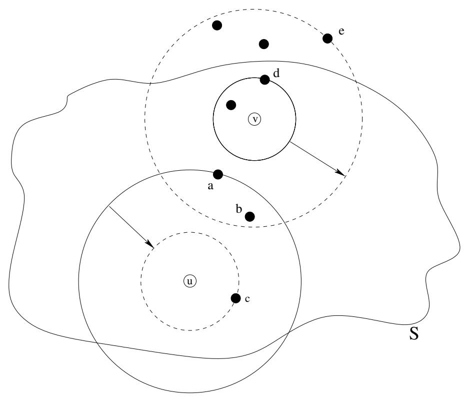

Fig. 1. A possible expansion at v and reduction at u.

sion at v, illustrated in Figure 1. Many known algorithms pick a node v with the least incremental power, disregarding the fact that *a larger expansion might actually be better*! The reason for this is simple, as shown in Figure 1. Nodes a and b, for example, were covered by node u before the expansion of v. After the potential expansion, however, $p[u]$ could be reduced significantly if nodes a and b are the farthest of the nodes which u cover.

There is a problem with the idea as it is, however. We may disconnect the network when reducing the power level. For example, if through u to a is the only way for the root to reach v, reducing the power level of u as above will disconnect v from the root. To cope with this problem, each $v \in S$ will maintain locally an (r, v)-path corresponding to the current power assignment. The check for disconnectivity could then be done locally at the common neighbors of u and v like a and b, which are involved in the potential power reduction. This way, only one-hop local information is needed if the idea is to be made distributed.

B. The DMIP3S algorithm

1) SPF and IP3S: The *distributed multicast IP3S* (DMIP3S) algorithm is based on the shortest path first (SPF) idea [7] and the potential power saving idea.

The idea of SPF is to build the multicast tree more locally. We maintain a set S of nodes reachable from r so far, which is initialized to be $S = \{r\}$. We "grow" S until $R \subseteq S$.

In each iteration of SPF, we find a "shortest" path P between **any** node in S and **any** node in $R \setminus S$ (see Figure 2). Mathematically, this shortest path is a shortest path between S and $R \setminus S$. In the figure, the black nodes represent nodes in R. For each pair (x, y) of nodes on the plane, the distance between x and y is the minimum power to get from x to y. The term "shortest path" is taken in this sense. In our case, we take the "length" of an (x, y)-edge to be d_{xy}^{α}. The idea will be more clear if we take each such shortest path as a "virtual edge".

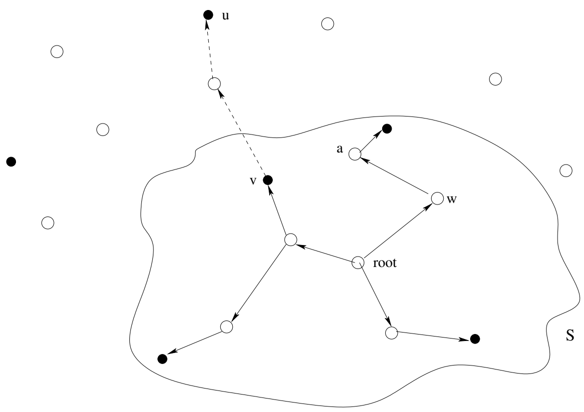

Fig. 2. Find a shortest path between S and $R \setminus S$

This power reduction can be applied systematically on each node whose power level is increased, as described in the previous section.

2) Distributed Algorithm: To turn the above description into a distributed algorithm, each node will store a list of power values (or distances) required to cover any of its neighbor. This information can be obtained locally in a variety of ways, including signal strengths, time delay, GPS, or microwave distance [13]. Each node also maintains a routing table indicating the next node along a shortest path to any node in the network. We have many options to build the routing table, including distance-vector like algorithms. As the network is static, this routing table only need to be constructed once,

Let S be the set of nodes in the current tree. Initially $S = \{r\}$ (source-initiated). Nodes in S are "white," or active nodes, the rest are "black."

Every white node keeps a list of uncovered multicast members. New white nodes will inherit this information from their parents when they become white, i.e. become active for the first time. The updating of the list shall be explained later.

Unless all its neighbors are white, every white node independently computes an object function value (OFV), which, in this case, is the amount of incremental power required to cover its nearest uncovered multicast member. Note that OFV could be computed by any objective function. The idea still works. In this sense, our idea applies to a class of algorithms.

The white nodes then set a timer proportional to OFV^2. (The values OFV^2 is mapped linearly to an interval from 0 to a maximum timer value Δ.)

Let's take a single white node s as example: s calculates its OFV, sets timer and sends a *report timer value* (RTV) message to its nearest multicast member d. The RTV packet contains the value of the OFV. After the timer expires s sends a *become white query* (BWQ) message to d. Node d then decides to send an *acknowledgement* (ACK) or a *negative-acknowledgement* (NACK) to s. An ACK is sent to a node with lowest OFV received so far, basically telling that node: "please cover me." A NACK is sent to all other nodes. If a white node gets a BWQ message, it also replies with a NACK. Upon the ACK reception node s sends a *become white* (BW) message to d and each node on the path from s to d will relay this BW message and increase its power to cover next hop neighbor.

A white node will switch its parent and send a *reduce power* (RP) message to its previous parent after overhearing the BW message. A black node would become white and start its timer. The BW message also contains the sender's list of uncovered multicast members.

If a node reduces its power to zero and it is not a multicast member then it can be removed from the set S. The node shall be inactive, but keep its color white until it receives a STOP message from the root. Its parent may be able to further reduce its power via the RP messages.

A white node will remove the member's entry from its uncovered multicast member list if it receives NACK targeting itself, or overhears ACK from that member. It will stop the algorithm when the list is empty or when a STOP message is received.

For fast termination of the whole algorithm, we let ACKs always be forwarded back along the current tree to the root r. Lastly, r will send a list update message forward along the tree when it receives the ACKs from all multicast member. The last list update message is also a STOP message, for nodes to release their resources.

C. The P-DIP3S algorithm

The P-DIP3S algorithm is the pruning version of the distributed IP3S algorithm, which we designed for the broadcast problem. Readers are referred to [?] for the detailed description of DIP3S.

V. Simulation results

We evaluate our distributed algorithms with extensive simulations. Eight algorithms P-IP3S, P-DIP3S (distributed), DMIP3S (distributed), MIP3S, SPF, P-BIP (also called MIP in [3]), P-EWMA, and P-RBOP are to be compared.

Although some of these algorithms were designed for the broadcast problem, we implement their pruning versions to see how they compare to ours when multicast group sizes are large. For large group sizes, the multicast problem becomes the broadcast problem.

The number of nodes varies from 10 to 100 with an increment of 10 at each step. All these nodes are uniformly distributed on a $10 * 10$ square. For each network instance, three different values of the propagation loss exponent α were tested: $2, 3$ and 4.

In the simulations done in previous works [3, 6], a node's maximum power level is always assumed to be sufficient to cover all the nodes. In our simulations, the p_{max} is set to be enough to cover 25%, 50%, 75%, and 100% of the region.

Here, to cover $x\%$ of the region means p_{max} can only be used to reach the nodes within a distance of $x\%$ of the maximum possible distance of any two nodes. This is the length of the diagonal of the 10×10 square, which is $10\sqrt{2}$.

Varying p_{max} has several values. The obvious one is that in reality nodes may not have enough power to reach the furthest nodes. Secondly, even when a node can reach really far, restricting its power to a certain percentage of the maximum possible power could save energy.

The source node and multicast group members R are selected randomly. We vary the size of R from 20% of all nodes to 80% of all nodes, at a step of 20% each.

It should be clear that there are a lot of graphs to be reported. As all of them follow the same trend, and due to space limitation, we are only able to put here a few graphs.

For each simulation configuration, which is a combination of network size n, propagation loss exponent α, node's power level p_{max}, and the multicast group size, we generate 100 random instances and run the above algorithms on them.

The performance metric we adopt is the total power of the multicast tree. To compare among different algorithms, we use the idea of *normalized power* [3]. Let

$$\begin{aligned}\mathcal{A} &= \{\text{P-DIP3S, P-IP3S, DMIP3S,}\\ &\quad \text{MIP3S, SPF, MIP, P-EWMA, P-RBOP}\}\end{aligned}$$

be the set of all eight algorithms to be evaluated. Let $T_A(I)$ denote the total power of the multicast graph of a network instance I, computed by algorithm $A \in \mathcal{A}$. Then, the normalized power of algorithm A on instance I is

$$T'_A(I) = \frac{T_A(I)}{\min\{T_A(I), A \in \mathcal{A}\}}.$$

As indicated in [3], this metric has the advantage that it does not depend on the size of the region being tested.

We present the results for the cases when $\alpha = 2, 4$, for multicast group sizes 20%, 40%, 60%, 80% and maximum power levels 100%, 50%, as shown in Figures 3, 4, **The rest of the graphs follow the exact same trend.**

From the simulations, we can make the following conclusions:

- Within the range of small multicast groups ($\leq 65\%$ the total number of nodes) or of large groups ($\geq 65\%$) the relative algorithms' performances are roughly the same no matter what the value of α is. However, when α increases the performance difference between algorithms becomes less. This is possibly due to the fact that large values of α forces nodes to use large power levels, leaving not much room for algorithms to leverage.
- The relative performances of the algorithms are also quite similar independent of the number of network nodes.
- The relative performances of the algorithms are the same no matter what the value of p_{max} is. (The figures are omitted because of the page limit.)
- When group sizes are very large ($\geq 80\%$ of the total number of nodes), i.e. when the problem becomes the broadcast problem, our centralized P-IP3S algorithm works the best amongst all known algorithms. The distributed version P-DIP3S is much better than the other distributed algorithm P-RBOP.
- There is a group size cut point of around 60 to 65%, where DMIP3S starts to work very well. In fact, the distributed DMIP3S works better than all known multicast algorithms, except for our own centralized MIP3S.
- The centralized MIP3S algorithm is the best multicast algorithm in the group.
- Since RBOP trades the power efficiency for less information exchange, the resulting tree is not very power efficient. It depends on the lasting time of the multi/broadcasting session. A power efficient tree is preferred when the session is long.
- P-RBOP's performance is highly unstable for large values of α (4).

VI. Conclusions

In this paper we have addressed the problem of devising distributed algorithms to solve the minimum energy consumption multicast problem in static wireless ad hoc networks.

By extensive simulations, we have shown that the proposed algorithm-DMIP3S outperform most known centralized algorithms when multicast groups are small, about 60% to 65% or less of the total number of nodes. In fact, MIP3S - the centralized version of DMIP3S - is the best centralized multicast algorithm.

On the other hand, for large group sizes, i.e. when the multicast problem tends to the broadcast problem, algorithms pruning on broadcast tree work better. Our P-DIP3S performs the best among the distributed algorithms and better than most of the centralized ones.

To the best of our knowledge, these are the first distributed algorithms for the power-conserving multicast problem.

References

[1] P.-J Wan, G. Calinescu, X.-Y. Li, and O. Frieder, "Minimum-energy broadcast routing in static ad hoc wireless networks," in *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. 2001, vol. 2, pp. 1162–1171, IEEE.

[2] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides, "Algorithms for energy-efficient multicasting in static ad hoc wireless networks," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 251–263, 2001.

[3] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. 2000, vol. 2, pp. 585–594, IEEE.

[4] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides, "Multicasting in energy-limited ad-hoc wireless networks," in *Proceedings - IEEE Military Communications Conference MILCOM*. 1998, vol. 3, pp. 723–729, IEEE.

[5] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides, "The effect of discrete power levels on energy-efficient wireless broadcast in ad hoc networks," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*. 2002, vol. 4, pp. 1655–1659, IEEE.

[6] Mario Cagalj, Jean-Pierre Hubaux, and Christian Enz, "Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues," in *Proceedings of the 8th annual international conference on Mobile computing and networking (MOBICOM)*, Atlanta, Georgia, USA, 2002, pp. 172–182, ACM Press.

[7] P.-J Wan and G. Calinescu, "Minimum-energy multicast in routing in static ad hoc wireless networks," 2001, manuscript.

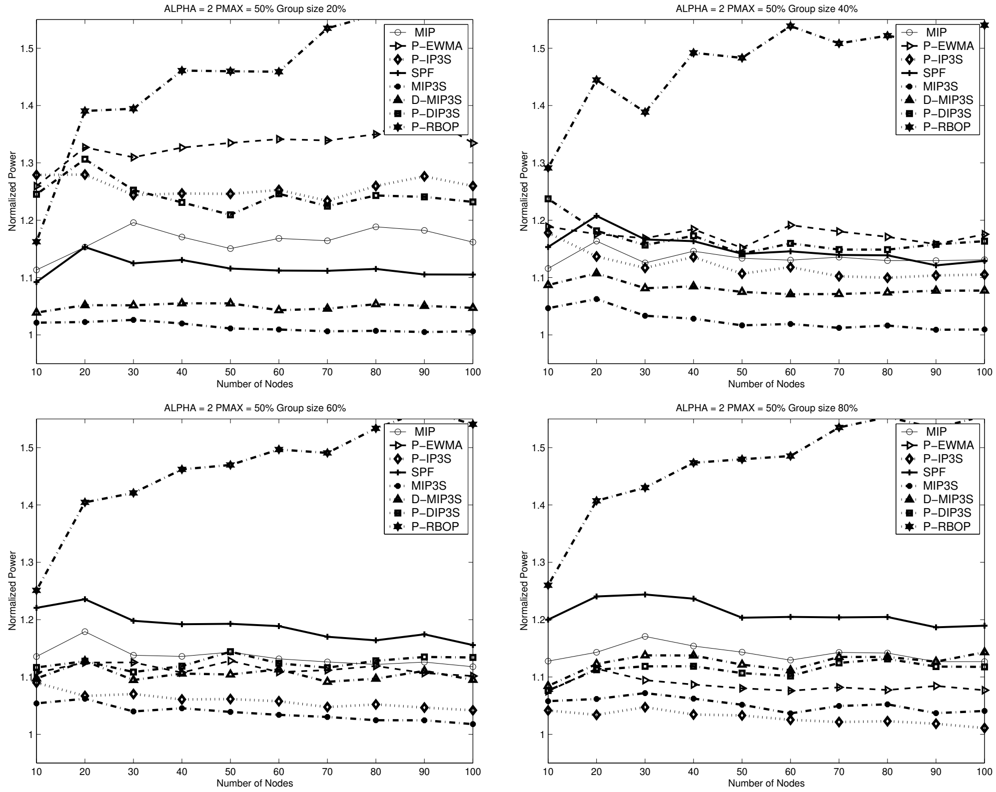

Fig. 3. The effect of different groups sizes. p_{max} is 50% of the region, $\alpha = 2$, group size is 20%, 40%, 60%, 80%.

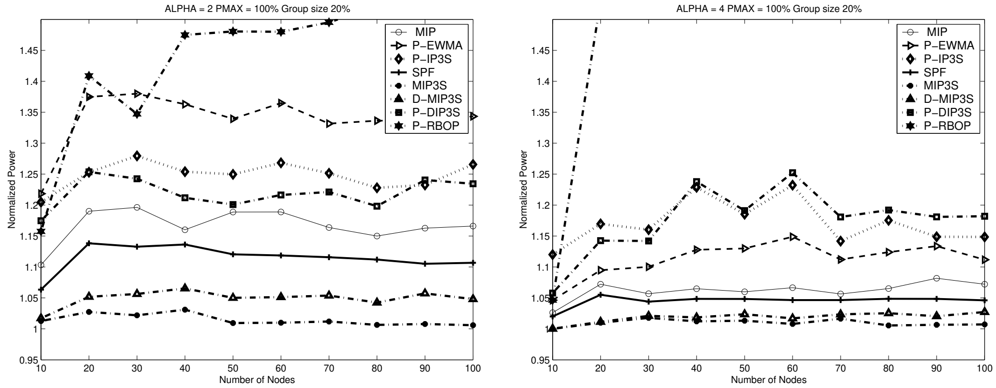

Fig. 4. The effect of different values of α. Both scenarios have p_{max} is 100% of the region, group size is 20%, $\alpha = 2, 4$.

[8] Hung Q. Ngo, Dazhen Pan, and Vikas Verma, "Power-conserving algorithms and protocols in ad hoc networks," in *Ad Hoc Networking: Recent Advances*, X. Cheng and D.-Z. Du, Eds. Kluwer Academic Publishers, 2003.

[9] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized minimum-energy broadcasting in ad-hoc networks," in *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. 2003, IEEE.

[10] T. Chu and I. Nikolaidis, "Energy efficient broadcast in mobile ad hoc networks," in *Proceedings of the 1st International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW)*, 2002, pp. 363–374.

[11] Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos, "New results for energy-efficient broadcasting in wireless networks," in *Proceedings of the 13th International Symposium 2002 on Algorithms and Computation (ISAAC '02)*. Nov 2002, pp. 332–343, Springer Verlag, Lecture Notes in Computer Science.

[12] Anthony Ephremides, "Energy concerns in wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 48–59, Aug 2002.

[13] A. Benlarbi, J.-C. Cousin, R. Ringot, A. Mamouni, , and Y. Leroy, "Interferometri c positioning systems by microwaves," in *Proceedings of Microwaves Symposium (MS)*, 2000.