MIP3S: Algorithms for Power-Conserving Multicasting in Static Wireless Ad Hoc Networks

Purnima M. Mavinkurve   Hung Q. Ngo   Himanshu Mehra
Computer Science and Engineering Department,
201 Bell Hall
State University of New York at Buffalo,
Amherst, NY 14260, USA.
{pmm24, hungngo, hmehra}@cse.buffalo.edu

***Abstract*— In a static ad hoc wireless 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 that the total power usage is as small as possible. In the centralized version of the problem, one finds such power assignment given the entire network topology. In the distributed version, a power assignment is found by exchanging information between neighboring nodes.**

In this paper, we proposed new algorithms based on the idea of *Multicast Incremental Power with Potential Power Saving* (MIP3S). Simulations show that the new algorithms work better than all known algorithms. Different versions of this idea, when made distributive, are of different time and message complexities, imposing an interesting trade-off in total saving in power and other complexity measures.

***Index Terms*— Power-aware routing, multicasting, wireless ad hoc networks.**

I. INTRODUCTION

Energy conservation in ad hoc networks is of paramount importance. In some applications, in fact, energy is entirely non-renewable [1, 2]. 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 [3–13].

All the works cited above and this paper deal with static ad hoc wireless networks, such as sensor networks (where power supply is very limited). The same problem for mobile networks is also very important, but it is not addressed here.

The power-aware multicasting problem we are facing is assumed to be *source initiated*, 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 assignment is such that all nodes in the multicast group are reachable, possibly in several hops, from the source node.

We would like to reduce the total power in the power assignment vector. Initially, several authors have casted this problem in terms of building a multicast tree rooted at the source [4, 5, 11]. In fact, the trees are *directed rooted trees*, also called *arborescences*, where there is a directed edge from u to v if and only if the power assigned at node u is large enough to reach v in one transmission hop. In reality, given a power assignment vector we could form a *reachability graph* which is a directed graph with an edge from u to v if and only if the power assigned at u is large enough to reach v in one hop. The reachability graph is not an arborescence in general. What we would like is to have a directed path from the root to every node in the multicast group, while keeping the total assigned power as small as possible. The arborescence is a subtree of the reachability graph, which spans all nodes in the multicast group. The use of the arborescence is obvious: it induces a routing table for all nodes.

The literature on the broadcast version of this problem is quite large, as shall be subsequently reviewed. On the other hand, there is very little done on the multicast version except the obvious idea of pruning some broadcast tree.

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 [9, 10, 14], hence this is naturally a difficult problem.

This paper addresses mainly the centralized version of the multicast problem, with an eye toward distributizing the algorithms we propose. In the centralized case, one finds a power assignment given the entire network topology. This is useful in cases where there is a controller for the network, or when some mechanism, such as flooding, is assumed to propagate network topology to all nodes. In the distributed version of the problem, a power assignment is found by exchanging information between neighboring nodes.

We propose several different algorithms, all of which work better than known algorithms, but are potentially of varying time and message complexities when distributized. Our multicast algorithms are not based on the common idea of pruning an energy efficient broadcast tree. However, once the multicast group sizes are large (at 65%-75% or more of the total number of nodes), it is better to apply our other algorithms designed for the broadcast problem and prune the tree.

The rest of this paper is organized as follows. Section II formally describe the problem. Section III selectively 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 known algorithms. Lastly, Section VI gives concluding remarks and future research directions.

II. Problem Formulation

A. Communication model

We assume that all nodes are equipped with omnidirectional antennas with adjustable power levels. There are two different basic assumptions: (1) the power levels can continuously be adjusted from 0 to some level p_{max}; (2) the power levels can only be chosen from a given discrete set $\{0, p_1, \dots, p_m\}$ of power levels. In fact, when the nodes are heterogeneous it is possible that each node has its own power level set in case (2), or different p_{max} in case (1).

To simplify our discussion, we shall mostly restrict ourselves to case (1). Algorithms and protocols with assumption (1) can easily be extended to handle case (2) in a variety of ways. A simple strategy is to assume (1) and then "round" an assigned power level up or down to the closest available level from the given set. Another strategy is to, instead of vary power levels continuously, vary the levels in a discrete manner. Whatever the strategy we choose, algorithms' performances are not changed by much if the granularity of the power level set is fine enough. The reader is referred to [8] for a discussion on the effect of different granularities of power level sets on energy-efficient protocols.

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 [1, 5, 15]. 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 if and only if $p_u \geq d_{uv}^\alpha$.

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

Let us continue with the u, v example above. If $p_{max} < d_{uv}^\alpha$, then it is impossible to get from u to v via a single hop. In this case, multi-hop transmission is necessary. In most cases, multi-hop transmission also saves the total power usage of the nodes involved.

We are now ready to define our multicasting problem. We first describe the problem in a highly general setting, then discuss the specialization reducing back to the original problem. Graph theoretic terminologies we use here are fairly standard (see, e.g., [16]). Terminologies and concepts for $\mathbf{NP}$-complete theory and approximation algorithms can be found in [17] and [18].

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

$\mathbf{MECMS}$ is the problem we address in this paper. In the problem, we are given a directed graph $G = (V, E)$ 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 u is at least the cost to reach v.

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

When $R = V - \{r\}$, the problem becomes the minimum energy consumption **broadcast** subgraph (MECBS) problem. A slightly more realistic variation of the problem also has a given $p_{max} > 0$ and requires that $p[v] \leq p_{max}, \forall v \in V$.

When the graph G is a complete graph whose nodes are points on a d-dimensional Euclidean space, $d \geq 1$, and $c(u, v) = d_{uv}^\alpha$, $\alpha \in [2, 4]$, we denote the problem as $\mathbf{MECMS}[N_d^\alpha]$. The case when $d = 2$ is of most interest, and is where most known results come from.

In this paper we focus our description on the case $d = 2$, although our algorithms and protocols work in the same way for the general d-dimension case. One reason for this restriction is that most known algorithms were designed for ad hoc networks on a 2-dimensional Euclidean plane, and hence it is natural to compare our algorithms with others' in this particular case.

III. Related works and our contributions

Most known multicast algorithms simply prune the broadcast tree constructed from an energy-efficient broadcast algorithm. Moreover, since boadcasting is a special case of multicasting, many theoretical analyses on broadcasting also apply on multicasting. Consequently, we review known results on the broadcast problem (MECBS) before the MECMS counterpart.

A. The broadcast case

Wieselthier et al. [5] studied several heuristics and studied their performances by simulations for the $\mathbf{MECBS}[N_2^\alpha]$ problem. The algorithms they studied include: MST (minimum spanning tree), BIP (Broadcast Incremental Power), and SPT (Shortest Path Tree). Although MST does not work well as compared to most other algorithms, the advantage is that it does have a constant approximation ratio. It can be shown that BIP works better than MST analytically [3].

Wan et al. [3] and Clementi et al. [19] gave upper and lower bounds on the performance ratios of several of these heuristics.

Cagalj et al. [10] came up with an algorithm called EWMA (Embedded Wireless Multicast Advantage) which tries to modify an MST to form a better power assignment. One advantage of EWMA is that it is a better modification of MST, hence its performance ratio is at least as good as MST, which is upper bounded by 12 for the $\mathbf{MECBS}[N_2^2]$ problem. The authors also showed by simulations that it works better than BIP on average.

The reader is also referred to [3, 9, 14] for some analytical results on the general graph version of the problem.

B. The multicast case

The easiest approach to build a multicast tree is to first build a broadcast sub-graph and *prune* it back, in much the same way the Internet multicast protocols work (DVMRP, PIM, etc.).

From now on, we 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. [5] have experimented the P-BIP, P-SPT, P-MST heuristics, and found out that for very small group sizes, P-SPT outperforms the other two algorithms, while for moderate to large group sizes, P-BIP performs the best.

Wan et al. [11] have done a theoretical evaluation of these algorithms, and found out 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. The algorithm starts from the root node r, grows out a set of covered nodes S with $S = \{r\}$ initially, and each time it finds a shortest path in G_S from any node in S to any uncovered node in R (the multicast group). This way, after each iteration at least a new node in M is covered. This algorithm can be done quite effectively by first running an all-pairs shortest path algorithm, such as the Floyd-Warshall algorithm [20].

Another algorithm based on Steiner minimum trees (SMT) [21] was also given in the same paper. The analog for BIP was called minimum incremental path first (MIPF), which works in much the same way as SPF, but we pick a new path which yields the least incremental power. It could be shown that both SPF and MIPF have constant approximation ratios.

C. Our contribution

Our first contribution is an important observation: building a (broadcast or multicast) tree starting from the source often involves increasing the power level at a certain node, say v, to cover some new nodes until all nodes (in the multicast group) are covered. The power expansion at v, however, could make the current power assignment at some other nodes redundant, in the sense that v's new power level covers nodes which have been covered before. Hence, by increasing v's power, we might be able to reduce some other nodes' powers! This idea is referred to as *potential power saving*.

There are two variations of this idea, which shall be made clearer in the next section. In one variation, the potential saving is taken into account to decide which node v to expand. This variation works better, but requires more computation. The other variation only performs power saving without accounting it in the objective function. This variation works slightly worse than the first, but requires less computation.

The potential power saving idea can be used to design energy-efficient broadcast and multicast algorithms. We shall refer to the broadcast version as IP3S (Incremental Power with Potential Power Saving), and the multicast version as MIP3S (Multicast IP3S). Naturally, we refer to the prune version of IP3S as P-IP3S.

MIP3S is a combination of SPF and potential power saving, and simulations show that it works very well when group sizes are not too large (approximately $\leq 60\%$ of all nodes).

IP3S is a combination of the ideas in Prim's MST algorithm and potential power saving. P-IP3S works very well when group sizes are relatively large (approximately $> 65\%$ of all nodes).

In all cases, our new algorithms work better than all known energy-efficient multicast algorithms, as simulation results shall demonstrate.

IV. Our algorithms

In this section, we present the main ideas of our algorithms. In fact, we have a class of algorithms applying the same idea with slightly different objective functions. The reason for considering different objective functions shall be explained later.

We shall first present ideas for P-IP3S, which works well for large multicast groups. Next, multicast algorithms for small groups named MIP3S and MIP3S-b are presented, which are slightly different in the way we choose a node to expand power. MIP3S has better performance, while MIP3S-b is simpler.

A. IP3S and P-IP3S

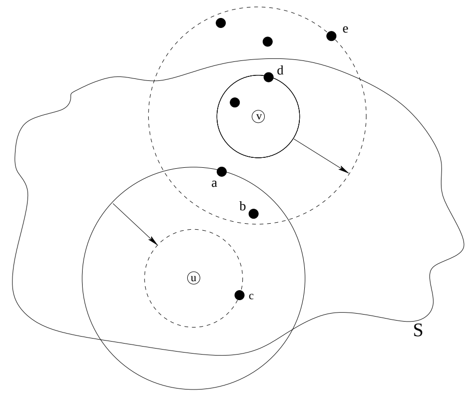

Fig. 1. A possible expansion at v and reduction at u.

For the broadcasting problem, the main line of attack has been to grow a set S of nodes reachable from the source r until all nodes are covered. We start from a power assignment vector of all 0's. The set of reachable nodes from the root, given the current power assignment vector, is then $S = \{r\}$.

Suppose at some step in the algorithm, we consider a possible power expansion of a node $v \in S$ to cover a few more uncovered nodes. Which $v \in S$ should be picked for expansion? In BIP, it was a $v \in S$ with the least incremental power needed to cover **one** new node.

Our idea here is to allow the potential expansion to cover a few more nodes at the same time, taking advantage of the potential saving in power by a more aggressive move! Consider node v in Fig. 1. The cost of expansion includes the incremental power needed to cover the new nodes, **minus** the potential saving of power by the new expansion. 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 covers.

Consequently, *we should pick a $v \in S$ so that the increase in power at v minus the sum of power savings at all the u's is as small as possible*. Note that the value of this objective function can be negative.

There is a problem with the idea as it is, however. Consider the situation in Fig. 2. After reducing $p[u]$ so that u does not cover a and b, node v may not be reachable from the source anymore. What we have done is that we broke the links (u, a) and (u, b). When all directed paths from r to v contain either (u, a) or (u, b), then breaking up (u, a) and (u, b) shall disconnect v from r. (Note also that, in general the paths from r to v may take a few hops until they get to a and/or b, and then a few more hops before reaching v. The situation in the figure is only illustrative.)

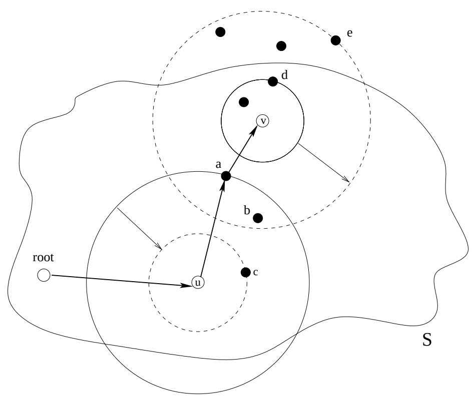

Fig. 2. A problem with naive expansion: potential disconnectivity.

As a consequence, in order to realize this "potential power saving" idea, we need to make sure that, before reducing any of the $p[u]$, there is still some (directed) path from r to v which does not use (u, a) or (u, b). If there was an r, v-path which does not involve (u, a) but does involve (u, b), then we can reduce $p[u]$ to d_{ub}^{α}, which would break the link (u, a) but still keep the link (u, b).

To cope with this *potential disconnectivity* problem, we adopt the following solution: for each $v \in S$, maintain a pointer to a parent node along **one** r, v-path; the check for disconnectivity could then be done by tracing back these parent pointers to r.

After IP3S has constructed a broadcast tree rooted at r, one can easily prune the tree back to cover up to the multicast members only, in much the same way that IP-multicast algorithms work. The resulting algorithm is referred to as P-IP3S.

B. MIP3S and MIP3S-b

Recall that we have a subset $R \subseteq V - \{r\}$ of nodes where multicast data from r are to be delivered to. When $|R|$ is large, building the entire broadcast tree, and then pruning a few branches makes sense, and has been proven to work very well. However, when $|R|$ is small, building the broadcast tree is too greedy. A good broadcast tree looks more at the global picture of overall saving, hence pruning it back does not necessarily give a good local solution.

MIP3S and MIP3S-b are based on the shortest path first (SPF) idea [11] and the potential power saving idea.

The idea of SPF is to build the multicast tree more locally. We also 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 Fig. 3). 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}.

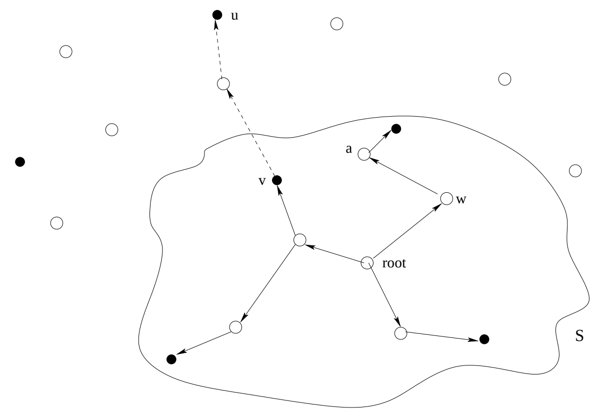

Fig. 3. Find a shortest path between S and $R \setminus S$

To this end, let us see how the potential power saving idea can be applied here. In Fig. 3, after the shortest path from v to u is found, v's power level is supposed to be expanded to cover the next node on the path. However, this expansion allows v to also cover a, which was previously covered by w. Hence, we can reduce w's power significantly.

This power reduction can be applied systematically on each node whose power level is increased, as described in the previous section. Due to space limitation, we are only able to give the reader the intuitive idea behind this method. The full pseudocode with all its rigorous analysis shall be presented somewhere else. Another twist we implemented was that we find a shortest *incremental* path between S and $R \setminus S$.

After the new path is found, all nodes on the path are assigned with the power level needed to reach the next node on the path. The set S is now expanded to include all nodes on the path along with all nodes covered by the new power assignment.

What we have just described is the algorithm named MIP3S-b, whose actions in each iteration can be summarized as follows.

(1) Find a shortest incremental path P from S to $R \setminus S$. **(2)** Assign each node on P the new power level to reach the next node. **(3)** For each node v along P, run the power reduction method as in IP3S.

The other variation – MIP3S – requires a little bit more work. In step 1 of each iteration, we find a path P from S to $R \setminus S$

with the least total incremental power **minus** the total potential saving (which was supposed to be done afterward in step 3).

The description of the algorithms may sound to require a lot of work. However, if we run an all-pair shortest path algorithm on the entire network as a preprocessing step, such as the Floyd-Warshall algorithm [20], then the total running time is at most $O(n^3)$.

V. SIMULATION RESULTS

We evaluate our idea with extensive simulations. Six algorithms P-IP3S, MIP3S-b, MIP3S, SPF, P-BIP (also called MIP in [5]), P-MST are to be compared. In fact, we have implemented also P-EWMA but it does not work well, so we omit the result in the graphs.

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 [5, 10], 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 100%, 75%, 50%, 25% of the region.

The source node is selected randomly among all the nodes. The multicast group members R are also selected randomly. We vary the size of R from 10% of all nodes to 100% of all nodes.

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. Interested readers certainly can contact us for all the graphs.

For each simulation configuration - a combination of network size n, propagation loss exponent α, node's power level p_{max}, multicast group size - we generate 100 random instances and run the above 6 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* [5]. Let

$$\mathcal{A} = \{\text{P-IP3S, MIP3S-b, MIP3S, SPF, MIP, P-MST}\}$$

be the set of all six 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 [5], this metric has the advantage that it does not depend on the size of the region being tested. If nodes are distributed in a larger region, then the overall power consumption is scaled by a certain factor, which does not effect the normalized powers.

Due to space limitation, we only present the results for the cases when $\alpha = 2, 4$, for multicast group sizes 10%, 75%, and maximum power levels 100%, 50%, as shown in Fig. 4, 5. **The rest of the graphs follow the exact same trend.**

From the simulations, we can make the following conclusions:

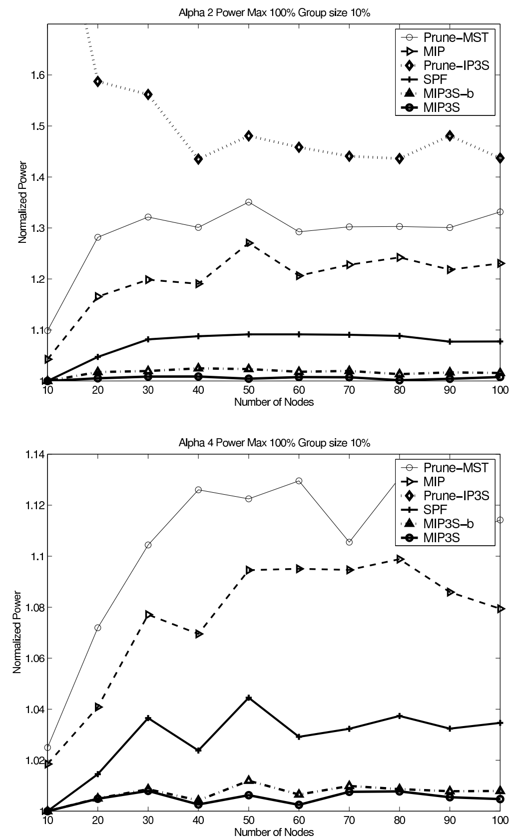

Fig. 4. p_{max} is 100% of the region, group size is 10%

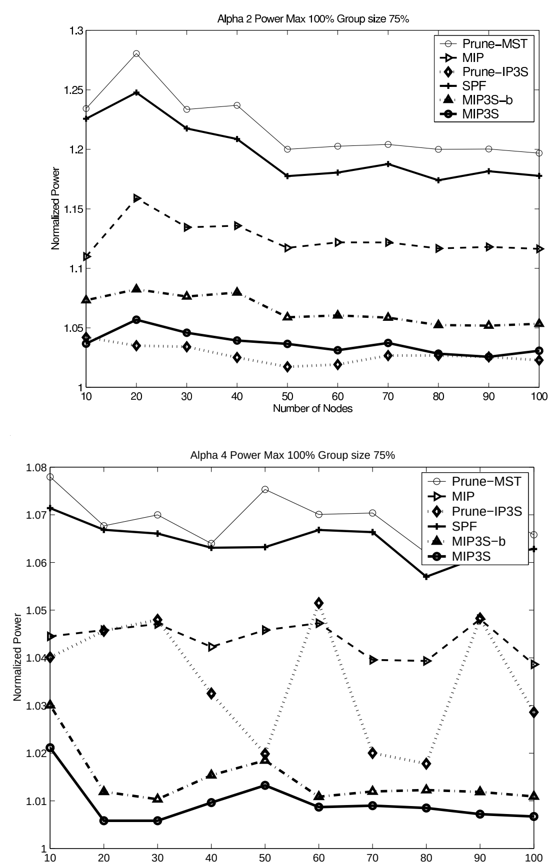

Fig. 5. p_{max} is 100% of the region, group size is 75%

- The relative performances of the algorithms are the same no matter what the value of α is.
- The relative performances of the algorithms are also quite similar independent of the number of network nodes.
- MIP and MST perform worse as α increases
- P-IP3S is worse for larger α and smaller group sizes
- The performance of SPF tends to MST as group sizes are larger. This makes sense, since SPF is MST when the multicast group contains all nodes, i.e. in the broadcasting case.
- When group sizes are up to about 60%-65% of the total number of nodes, MIP3S and MIP3S-b outperform every one else.
- When group sizes are more than about 65%, MIP3S and MIP3S-b are still not too bad, but P-IP3S starts to work much better. In fact, when group sizes are high, namely when the multicasting problem is becoming a broadcasting problem, P-IP3S performs the best.

  We have implemented known power-aware broadcast algorithms like EWMA, BIP, MST, etc. and IP3S outperforms all those algorithms. The exact performance nature of IP3S and P-IP3S for larger group sizes, along with simple methods to make them distributive, are to be reported in a different paper.
- MIP3S-b is quite a bit simpler than MIP3S, yet the performance of MIP3S-b is almost always comparable to that of MIP3S.

VI. Conclusions and future works

In this paper we have addressed the problem of devising algorithms to solve the minimum energy consumption multicast subgraph problem in wireless ad hoc networks. Our algorithms are based on the idea of potential power saving, and are named P-IP3S, MIP3S, and MIP3S-b.

The algorithms differ in complexities and performances depending on the multicast group sizes. MIP3S and MIP3S-b perform better than all known power-aware multicast algorithms when multicast group sizes are relatively small. On the other hand, P-IP3S is the best when the group sizes are larger.

In fact, when the multicast group is all network nodes, P-IP3S is a broadcast algorithm which is better than all known broadcast algorithms in terms of power consumption. More detail results on P-IP3S, its performance, and methods to make it distributive are to be reported in a different paper.

The variation MIP3S-b involves less message exchanges than MIP3S, but it performs slightly worse than MIP3S. However, MIP3S-b is competitive as compared to other known multicast algorithms such as P-MST, MIP, and SPF.

There are several problems we are working on, arising from the ideas of this paper. Firstly, the precise nature of the trade-off involved when distributizing these algorithms needs to be thoroughly addressed. In particular, precise values of timers, delays, and power saving need to experimentally verified. Secondly, an mathematical analysis of P-IP3S, MIP3S and MIP3S-b needs to be done to see if they give better approximation ratios than the known upper bound of 24 for SPF. A good lower bound on approximation ratio should also be devised.

References

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

[2] Charles E. Perkins, *Ad Hoc Networking*, Pearson Education, New Jersey, USA, Dec 2000.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides, "Energy-efficient multicasting of session traffic in bandwidth- and transceiver-limited wireless networks," *Cluster Computing*, vol. 5, no. 2, pp. 179–192, 2002.

[8] 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.

[9] Weifa 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)*, Lausanne, Switzerland, 2002, pp. 112–122, ACM Press.

[10] 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.

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

[12] Vikas P. Verma, Amit Chandak, and Hung Q. Ngo, "Distributive routing algorithms for power-conserving broadcasting in wireless ad hoc networks," 2003, submitted.

[13] 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.

[14] 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.

[15] Theodore Rappaport, *Wireles Communications: Principles and Practices*, Prentice Hall PTR, New Jersey, USA, second edition, Dec 2001.

[16] Douglas B. West, *Introduction to graph theory*, Prentice Hall Inc., Upper Saddle River, NJ, 1996.

[17] Michael R. Garey and David S. Johnson, *Computers and intractability*, W. H. Freeman and Co., San Francisco, Calif., 1979, A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences.

[18] Dorit S. Hochbaum, Ed., *Approximation Algorithms for NP Hard Problems*, PWS Publishing Company, Boston, MA, 1997.

[19] Andrea E. F. Clementi, Pilu Crescenzi, Paolo Penna, Gianluca Rossi, and Paola Vocca, "On the complexity of computing minimum energy consumption broadcast subgraphs," in *STACS 2001 (Dresden)*, vol. 2010 of *Lecture Notes in Comput. Sci.*, pp. 121–131. Springer, Berlin, 2001.

[20] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to algorithms*, MIT Press, Cambridge, MA, second edition, 2001.

[21] Frank K. Hwang, Dana S. Richards, and Pawel Winter, *The Steiner tree problem*, vol. 53 of *Annals of Discrete Mathematics*, North-Holland Publishing Co., Amsterdam, 1992.