r-shrink: A Heuristic for Improving Minimum Power Broadcast Trees in Wireless Networks

Arindam K. Das, Robert J. Marks, Mohamed El-Sharkawi, Payman Arabshahi, Andrew Gray

Abstract— Broadcasting in wireless networks, unlike wired networks, inherently reaches several nodes with a single transmission. For omni-directional wireless broadcast to a node, all nodes closer will also be reached. This property can be used to compute routing trees which minimize the sum of the transmitter powers. It has been shown that this problem is NP-complete. In this paper, we present the r-shrink procedure, a heuristic for improving the solutions obtained using fast sub-optimal algorithms. Specifically, we focus on the low-complexity BIP algorithm and Prim's minimum spanning tree algorithm and show through extensive simulations that better solutions are obtained almost always, with considerably lower tree power, if the proposed procedure is used to improve the trees generated using these algorithms.

I. Introduction

Broadcasting/multicasting in wireless networks is fundamentally different as compared to wired networks, since multiple nodes can be reached by a single transmission. This, of course, assumes that the nodes are equipped with omni-directional antennas, so that if a transmission is directed from node i to node j, all nodes which are nearer to i than j will also receive the transmission. This is known as the "wireless multicast advantage" [1]. For a given network with an identified source node, the minimum power broadcast (MPB) problem in wireless networks is to communicate to all remaining nodes, either directly or hopping, such that the overall transmission power is minimized. It is shown in [2] that the MPB problem in wireless networks is NP-complete, implying that optimal polynomial time algorithms are unlikely to exist.

Although previous work in this area focused on a "link-based solution", Wieselthier *et al* [1] note that a "node based" approach is needed for wireless environments. The *Broadcast Incremental Power* (BIP) algorithm suggested by them is a simple sub-optimal heuristic for constructing minimum power broadcast trees in wireless networks. In this algorithm, new nodes are added to the tree on a minimum incremental cost basis, until all intended destination nodes are included. It was subsequently shown in [3] that the BIP algorithm has an approximation ratio between 13/3 and 12. Other techniques that have been suggested for solving this problem include an internal nodes based broadcasting procedure by Stojmenovic *et al* [4], an evolutionary approach by Marks *et al* [5], a *swarm* based procedure by Das *et al* [6] and a localized algorithm by Cartigny *et al* [7]. Integer linear programming models for optimal solution of the MPB problem have been proposed in [8].

In this paper, we discuss the r-shrink procedure, a simple local search heuristic for *improving* sub-optimal MPB trees in wireless networks. Given an initial tree (which can be obtained using a minimum spanning tree algorithm or the BIP algorithm, for example), the transmission radii of the transmitting nodes in the tree are shrunk sequentially and an attempt is made to better accommodate the nodes, which have been disconnected from the tree as a result of the shrinkage operation, from other suitable nodes in the network. This process is repeated till no further improvement is possible.

II. Network Model

We assume a fixed[1] N-node network with a specified source node which has to broadcast a message to all other nodes in the network. Any node can be used as a relay node to reach other nodes in the network. All nodes are assumed to have omni-directional antennas, so that if node i transmits to node j, all nodes closer to i than j will also receive the transmission (provided line-of-sight exists).

We assume that, for a transmission from node i to j, the received signal power at j varies as $d_{ij}^{-\alpha}$, where

$$d_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2\right]^{1/2}$$

is the Euclidean distance between nodes i and j, (x_i, y_i) are the coordinates of node i and α (typically in the range $2 \leq \alpha \leq 4$) is the channel loss exponent. Consequently, the transmitter power at i necessary to support the link $i \to j$, $\mathbf{P}_{ij}$, is proportional (accounting for link and antenna gains) to d_{ij}^{α}.[2] Without any loss of generality, we set the proportionality constant to be equal to 1 and therefore:

$$\mathbf{P}_{ij} = d_{ij}^{\alpha} \tag{1}$$

A.K. Das, R.J. Marks and M. El-Sharkawi are with Department of Electrical Engineering, University of Washington, Box 352500, Seattle, WA 98195. e-mails: {arindam, marks, melshark}@ee.washington.edu.

P. Arabshahi and A. Gray are with Jet Propulsion Laboratory, 4800 Oak Grove Drive, MS 238-343, Pasadena, CA 91109. e-mails: {payman, gray}@arcadia.jpl.nasa.gov

[1] The accuracy of the proposed heuristic is constrained primarily by the need to know, with high accuracy, the locations of the nodes. As such, the "fixed network" restriction can be loosened to include *slowly mobile* networks, as long as the node locations can be determined quickly and accurately.

[2] Note that the triangle inequality (w.r.t $\mathbf{P}'_{ij}s$) may not hold because of the exponent α. The conditions under which the triangle inequality holds can be obtained using elementary geometry and is discussed in [1].

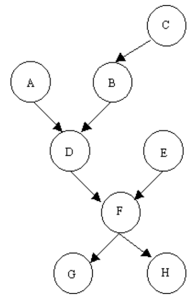

Fig. 1. Example digraph. • $ch(F) = \{G, H\}$. • $pa(F) = \{D, E\}$. • $de(D) = \{F, G, H\}$. • $nd(D) = \{A, B, C, E\}$.

The power matrix of a network, $\mathbf{P}$, is defined to be an $N \times N$ symmetric matrix whose $(i, j)th$ element, $\mathbf{P}_{ij}$, represents the power required to support the link $i \rightarrow j$. We do not assume any constraint on maximum transmitter power. However, the algorithm we discuss in this paper can be extended straightforwardly to the case where this assumption does not hold by redefining the power matrix such that $\mathbf{P}_{ij} = \infty$ if $d_{ij}^{\alpha} > Y_i^{max}$, where Y_i^{max} is the maximum allowable transmitter power of node i.

Finally, we assume that power expenditures due to signal reception and processing are negligible (as in [1]) and therefore the total cost of a broadcast tree is equal to the sum of the node transmitter powers.

III. Some Definitions

Given a directed graph (digraph) $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{V_1, V_2, \cdots V_N\}$ is a set of nodes and $\mathcal{E} = \{(V_i, V_j); i \neq j\}$ is a set of directed edges:

- The **parents** of a node i, denoted by $pa(i)$, is the set of nodes that are directed to it. If node j is a parent of node i, then i is a **child** of j (denoted by $ch(j)$). Thus:

  $$j \in pa(i) \Longleftrightarrow i \in ch(j)$$

  For example, referring to Figure 1, $pa(F) = \{D, E\}$ and $ch(F) = \{G, H\}$.
- A **path** between two nodes i and j (denoted by $i \mapsto j$) is a sequence $i, \alpha_0, \alpha_1, \cdots, \alpha_n, j$ of distinct nodes such that $(\alpha_{k-1}, \alpha_k) \in E$, for all k. In Figure 1, $< B, D, F, H >$ is a path between nodes B and H. $< H, F, D, B >$, however, is not a path between nodes H and B.
- The **descendants** of a node i, denoted by $de(i)$, is defined as the set of nodes $\{j\}$, such that there is a path from i to all nodes in $\{j\}$. Thus:

  $$de(i) \stackrel{\triangle}{=} \{j : i \mapsto j \text{ but not } j \mapsto i\}$$

  The $\mathbf{non-descendants}$ of a node i, denoted by $nd(i)$, is defined as:

  $$nd(i) \stackrel{\triangle}{=} V \setminus \{i \bigcup de(i)\}$$

  In Figure 1, $de(D) = \{F, G, H\}$ and $nd(D) = \{A, B, C, E\}$.
- The **indegree** of a node i, denoted by $d_{in}(i)$, is defined as the number of edges incident to it. Similarly, the **outdegree** of a node i, denoted by $d_{out}(i)$, is defined as the number of edges directed away from it. In Figure 1, $d_{in}(D) = 2$ and $d_{out}(D) = 1$.
- In the context of this paper, a node is defined to be a **leaf node** if its indegree is 1 and outdegree is 0. For example, nodes G and H are leaves in Figure 1.

IV. Constructing a digraph from an initial tree

In this section, we explain how to construct a digraph, given a wireless broadcast tree. First, we define the following sets:

k = transmission step number
$\mathbf{NA}^{(k)}$ = set of all nodes reached in transmission step k ($\mathbf{NA}^{(0)}$ = [source])
$\mathbf{NN}^{(k)}$ = set of new nodes reached in transmission step k ($\mathbf{NN}^{(0)}$ = [source])
$\mathbf{NA}^{(0:k)}$ = set of all nodes reached till transmission step $k = \bigcup_{m=0}^{k} \mathbf{NA}^{(m)} = \bigcup_{m=0}^{k} \mathbf{NN}^{(m)}$

We define a tree to be *connected* if the kth ($\forall k \geq 2$; for $k = 1$, the source is the transmitting node) transmitting node in the tree has been reached by any of the prior transmissions.

Consider the 6-node network in Figure 2, node 6 being the source.

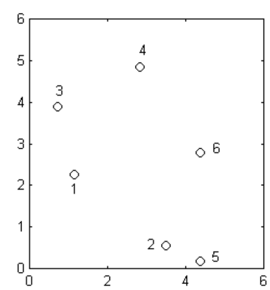

Fig. 2. Example 6-node network. Node 6 is the source.

The power matrix of the network, for $\alpha = 2$, is:

$$\mathbf{P} = \begin{bmatrix} 0 & 8.51 & 2.79 & 9.51 & 14.92 & 10.73 \\ 8.51 & 0 & 18.80 & 19.02 & 0.93 & 5.75 \\ 2.79 & 18.80 & 0 & 5.29 & 27.18 & 14.51 \\ 9.51 & 19.02 & 5.29 & 0 & 24.48 & 6.74 \\ 14.92 & 0.93 & 27.18 & 24.48 & 0 & 6.85 \\ 10.73 & 5.75 & 14.51 & 6.74 & 6.85 & 0 \end{bmatrix} \quad (2)$$

Assume that we are given the following broadcast tree, T.

$$T = \{6 \rightarrow 4, 4 \rightarrow 1, 1 \rightarrow 5\} \quad (3)$$

The nodes reached by the transmissions in T are shown in Table I. The second column in the table lists the $\{$*transmitting node*, *destination node*$\}$ pairs in the broadcast tree.

TABLE I

Nodes reached by the transmissions in (3).

k	$t \to d$	$\mathbf{NA}^{(k)}$	$\mathbf{NN}^{(k)}$	$\mathbf{NA}^{(0:k)}$
0	-	6	6	6
1	$6 \to 4$	2,4	2,4	2,4,6
2	$4 \to 1$	1,3,6	1,3	1,2,3,4,6
3	$1 \to 5$	2,3,4,5,6	5	1,2,3,4,5,6

A digraph corresponding to (3) is constructed using the $\mathbf{NN}^{(k)}$ entries in column (4) of Table I. From a graph theoretic viewpoint, using the $\mathbf{NN}^{(k)}$ entries in place of $\mathbf{NA}^{(k)}$ corresponds to restricting the indegree of all nodes (other than the source, whose indegree is 0) to 1. Figure 3 shows the digraph corresponding to the tree in (3). Note that while node 4 is reached both by the transmissions $6 \to 4$ and $1 \to 5$, it is shown as being connected to node 6 only in Figure 3. The solid lines indicate *actual transmissions* and the dotted lines indicate *implicit transmissions*. Node 2, for example, is reached implicitly by the (actual) transmission $6 \to 4$. The cost of the broadcast tree is defined as the sum of the costs of the actual transmissions in the digraph. For our example, it is $\mathbf{P}_{6,4} + \mathbf{P}_{4,1} + \mathbf{P}_{1,5} = 31.97$.

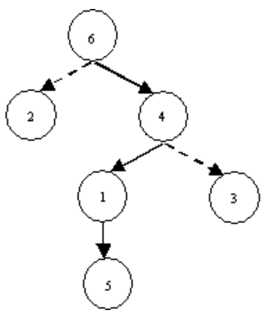

Fig. 3. Digraph corresponding to the broadcast tree in (3). Dashed lines indicate *implicit* transmissions. For example, node 2 is reached implicitly by the transmission $6 \to 4$. Cost of the tree $= \mathbf{P}_{6,4} + \mathbf{P}_{4,1} + \mathbf{P}_{1,5} = 31.97$

With respect to the digraph in Figure 3, node 6 (the source) is said to be at *level* 0, nodes 2 and 4 at *level* 1, nodes 1 and 3 at *level* 2, and node 5 at *level* 3. Clearly, if the maximum level in a digraph so constructed is MAX_LVL, all nodes at level $l = MAX\_LVL$ will be *leaf nodes*. Conversely, the maximum level parent nodes in such a digraph can occupy is $MAX\_LVL - 1$.

V. The r-shrink algorithm: $r = 1$

Given a transmission from node i to node j, with nodes $\alpha_0, \alpha_1, \cdots$ and α_k covered implicitly, let $\{\alpha_0, \alpha_1, \cdots, \alpha_k, j\}$ be an ordering of the nodes with respect to their distance from i. That is, α_0 is closest to i, α_1 is second closest, $\cdots$ and j is the farthest from i. For $r = 1$, the r-shrink operation applied to node i implies a reduction of its transmission power level (or, shrinkage of its transmission radius) by 1 *notch*, such that the farthest node now reached

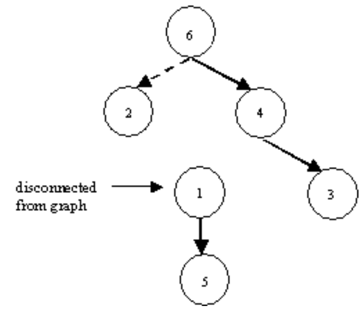

Fig. 4. Illustrating the effect of applying the 1-shrink operation to node 4 in Figure 3. Since node 3 is the second farthest node from 4, the edge (4,3) is now represented as a solid line in the graph. Also, since node 1 is not reached by any of the other transmissions, it is disconnected from the graph as a result of an 1-shrink operation on node 4.

is α_k instead of j. Similarly, for $r = 2$, the r-shrink operation implies a reduction of its transmission power level by 2 *notches*, such that the farthest node reached is α_{k-1}. For example, applying the 1-shrink operation to node 4 in Figure 3 would result in it transmitting to node 3, leaving node 1 disconnected, as shown in Figure 4. Applying a 2-shrink operation on node 4 would leave nodes 1, 3 and 5 disconnected from the rest of the tree.

Clearly, the r-shrink operation can only be applied to parent nodes in a digraph and the maximum 'r' by which the transmission radius of a parent node (say, node i, $i \neq source$) can be shrunk, denoted by $r_{max}(i)$, is equal to the number of children of i, or, the outdegree of node i. If $i = source$, the maximum 'r' by which its transmission radius can be shrunk is 1 less than its number of children.[3] That is:

$$r_{max}(i) = \begin{cases} d_{out}(i), & \text{if } i \neq source \\ d_{out}(i) - 1, & \text{otherwise} \end{cases} \tag{4}$$

In the discussion of the r-shrink algorithm that follows, we assume that $r = 1$.[4]

Given an initial digraph, the algorithm works by sequentially applying the 1-shrink operation to the parent nodes in the graph and checking whether the children, which have been temporarily disconnected from the graph as a result of the shrinkage operation, can be better accommodated from any of their $non-descendants$, *excluding the current parent*. If node i is a child of j, the set of the non-descendants of i, excluding its current parent, is given by $V \setminus \{nd(i) \cup j\}$. We will refer to members of such a set as the **foster parents**[5] of node i, denoted by $fpa(i)$. Therefore:

$$fpa(i) \triangleq V \setminus \{nd(i) \cup pa(i)\} \tag{5}$$

For example, in Figure 3, $fpa(1) = \{6, 2, 3\}$.

[3] This prevents the possibility of the source being a leaf in the graph.

[4] See [9] for a discussion of the algorithm when $r > 1$.

[5] Note that restricting the foster parents to the set of the non-descendants, excluding its current parent, prevents the occurrence of cycles in the improved graph.

Incremental and *decremental costs* are used to determine whether a temporarily disconnected child retains its existing parent or is assigned a new parent from the set of its foster parents.

- For a node i transmitting to node j, the incremental cost of adding node k to its reach is $\mathbf{P}_{i,k} - \mathbf{P}_{i,j}$. If i is a non-transmitting node, the incremental cost of adding node k to its reach is $\mathbf{P}_{i,k}$.
- For a node i transmitting to node j, with nodes α_0, $\alpha_1, \cdots, \alpha_k$ (arranged in order of increasing distance from i) covered implicitly, the decremental cost for letting node j out of its reach is $\mathbf{P}_{i,j} - \mathbf{P}_{i,\alpha_k}$. If no node is covered implicitly (*i.e.*, $d_{out}(i) = 1$), the decremental cost for letting node j out of its reach is $\mathbf{P}_{i,j}$.

For example, if we apply the 1-shrink operation to node 1 in Figure 3, the incremental cost of assigning node 5 as a child of node 6 is 0.11 ($\mathbf{P}_{6,5} - \mathbf{P}_{6,4}$), which is much smaller than the cost of retaining it as a child of node 1 (decremental cost), 14.92. Node 5 can therefore be better accommodated from 6 instead of 1.

In general, if $i \in ch(j)$ and $k \in fpa(i)$, i can be better accommodated from k if:

$$\textit{incremental cost at } k < \textit{decremental cost at } j$$

If there is more than one foster parent better able to accommodate the temporarily disconnected child, the one which would lead to a maximum reduction in overall tree cost is chosen to be the new parent. Ties, if any, are broken arbitrarily.

The sequence in which the parent nodes are checked is bottom-up; *i.e.*, parents at level $MAX\_LVL - 1$ are checked first, followed by those at level $MAX\_LVL - 2$, terminating with the source at level 0. If an improvement is found at any step, the graph is modified and the process is repeated on the new graph. Figure 5 provides a high level description of the 1-shrink algorithm.

1. Given an initial broadcast tree, construct a digraph G, as explained in Section IV.
2. Set MAX_LVL = maximum level in G.
3. Set $l = MAX\_LVL - 1$.
4. Set no_pa = no. of parent nodes at level l.
5. Set $\vec{pa}_l$ = set of parent nodes at level l.
6. Set $n = 1$.
7. Apply the 1-shrink operation to the parent node $\vec{pa}_l(n)$. Check whether its temporarily disconnected child, i, can be better accommodated from any of the nodes in the set $fpa(i)$.
8. **if**(better accommodation possible)
  /* True if the incremental cost at the foster parent node is smaller than the decremental cost at $\vec{pa}_l(n)$. */
    • Identify the foster parent which will lead to a maximum reduction in tree cost. Modify G by assigning this node to be the parent of i.
    • Repeat steps 2 to 8 on the new G.
  else
    if($n < no\_pa$)
       /* Not all parent nodes checked at level l */
      $n = n + 1$;
      Repeat steps 7 and 8.
    else /* All parent nodes checked at level l */
      if($l > 0$)
        $l = l - 1$;
        Repeat steps 4 to 8.
      else /* All parent nodes checked at all levels */
        Stop and print G. /* End of procedure */
      endif
    endif
  endif

Fig. 5. High level description of the 1-shrink algorithm.

VI. Simulation Results

We tested the 1-shrink algorithm on 10, 25, 50, 75 and 100-node networks in a 5 × 5 grid. In each case, 50 networks were randomly generated and the tree powers averaged to obtain the mean tree power. 'α' was chosen to be equal to 2 for all cases. The mean tree powers for the BIP solutions are shown in column (2) in Table II. The mean tree powers for the BIP solutions followed by the sweep algorithm proposed in [1] are shown in column (3). Column (4) lists the mean tree powers obtained by applying the 1-shrink procedure to the BIP solutions. Column (5) shows the average costs of the directed minimum spanning trees (dMST) obtained using Prim's algorithm[6] and column (6) represents the mean tree powers obtained by applying the 1-shrink procedure to the dMST's. Finally, column (7) shows the best-known results. For $N = 10$, these were obtained using an optimal integer programming (IP) approach discussed in [8]. The freely available LP solver, LPSOLVE [12], was used to solve the IP models. For $N = 25$ and 50, the best known results were obtained using an Ant Colony System (ACS) approach discussed in [6]. For $N = 75$ and 100, the best known results were computed using a "cluster-merge" variation of the ACS algorithm [10]. Figures in parentheses in columns (3) to (7) represent the percentage improvement in mean tree power over the BIP solutions.

It can be seen from Table II that there is a considerable improvement in tree power if the 1-shrink procedure is applied to the BIP solutions as opposed to the sweep algorithm. The mean tree powers obtained using the 1-shrink procedure on BIP solutions are between 5% and 10% of the best known solutions in all cases. In fact, except for $N = 25$, the improved BIP solutions are all within 5-7% of the best solutions.

Additionally, it may be noted that, while the dMST's themselves are approximately 4-6% worse (compare columns 2 and 5 in Table II) than the BIP solutions, applying the 1-shrink improvement heuristic on them yields solutions which are approximately within 2% of the BIP based solutions for $N = 10$ and 25, and within 1% for network sizes 50 and higher (compare columns 4 and 6 in Table II). This is highly encouraging since the fastest known implementation of Prim's algorithm using a Fibonacci heap is of complexity $O(E + N\log_2 N)$, compared to an $O(N^3)$ complexity of the BIP algorithm, where E is the number of edges in the graph and N is the num-

[6] See Chapter 4 of [11] for a description of Prim's MST algorithm.

TABLE II

Mean tree powers for • BIP (column 2) • BIP followed by sweep (column 3) • BIP followed by 1-shrink (column 4) • *d*MST (column 5) • *d*MST followed by 1-shrink (column 6) and • best known solutions (column 7). The figures in parentheses in columns (3) to (7) represent the percentage improvement in mean tree power over the BIP solutions.

N	BIP	BIP(sweep)	BIP(1-shrink)	*d*MST	*d*MST(1-shrink)	*Best Known*
10	11.57	11.08 (−**4.23**%)	10.60 (−**8.38**%)	12.09 (+**4.50**%)	10.84 (−**6.30**%)	10.01 (−**13.48**%)
25	12.46	12.14 (−**2.57**%)	11.25 (−**9.71**%)	13.09 (+**5.06**%)	11.43 (−**8.27**%)	10.21 (−**18.05**%)
50	11.67	11.45 (−**1.89**%)	10.68 (−**8.48**%)	12.29 (+**5.31**%)	10.76 (−**7.80**%)	10.04 (−**13.97**%)
75	11.63	11.37 (−**2.23**%)	10.67 (−**8.25**%)	12.10 (+**4.04**%)	10.80 (−**7.14**%)	9.88 (−**15.05**%)
100	11.60	11.35 (−**2.16**%)	10.55 (−**9.05**%)	11.97 (+**3.19**%)	10.66 (−**8.10**%)	9.87 (−**14.91**%)

ber of nodes. Interestingly, the gap between the *d*MST solutions and the BIP solutions also gets narrower as the network size increases. Since distributed implementations of MST algorithms already exist, we are currently working on a localized version of the 1-shrink algorithm, which would enable a completely distributed implementation of the minimum power broadcast tree.

The average number of iterations for which the 1-shrink algorithm ran before converging are given below:

- $N = 10:$ 0.78 (BIP-based), 1.76 (*d*MST-based)
- $N = 25:$ 2.86 (BIP-based), 6.26 (*d*MST-based)
- $N = 50:$ 6.62 (BIP-based), 10.54 (*d*MST-based)
- $N = 75:$ 9.78 (BIP-based), 16.82 (*d*MST-based)
- $N = 100:$ 13.72 (BIP-based), 21.08 (*d*MST-based)

Clearly, while computing the *d*MST is computationally less expensive than computing the BIP tree, it requires more iterations of the improvement procedure to generate similar quality solutions. This tradeoff between the running times of the tree-growing and tree-improvement phases needs to be carefully considered before any algorithm is selected.

We conclude this section by noting the number of times better solutions were found by the 1-shrink procedure than the sweep algorithm discussed in [1], when the initial tree was chosen to be the BIP solution.

- $N = 10:$ 25, out of 50 instances.
- $N = 25:$ 46, out of 50 instances.
- $N = 50/75/100:$ 50, out of 50 instances.

As can be seen from above, the 1-shrink procedure is able to find a better solution almost always, even for small networks such as $N = 25$.

VII. Conclusion

In this paper, we have presented the 1-shrink algorithm (a special case of the generalized r-shrink algorithm when $r = 1$), a heuristic procedure for improving minimum power broadcast trees in wireless networks. Simulation results show that considerably better solutions can be obtained if the proposed procedure is applied to trees grown using the Broadcast Incremental Power (BIP) algorithm, instead of the sweep technique discussed in [1]. Similar improvements are possible if it is applied to improve directed minimum spanning trees of the networks.

Acknowledgment

This work was supported by the Advanced Information Systems Technology (AIST) program at the NASA Office of Earth Sciences. The authors would also like to thank the anonymous reviewers for their detailed and insightful comments. While we have endeavored to accommodate as many of their suggestions as possible, we have not been able to adequately address some of their suggestions in this paper due to space constraints.

References

1) J.E.Wieselthier, G.D. Nguyen and A. Ephremides, “On the construction of energy-efficient broadcast and multicast trees in wireless networks”, Proc. INFOCOM, 2000.
2) M. Cagalj, J.P. Hubaux, C. Enz, “Minimum-energy broadcast in all-wireless networks: NP- completeness and distribution issues,” Proc. MOBICOM, Sep. 2002, Atlanta, Georgia, USA.
3) P.-J. Wan, G. Calinescu, X.-Y. Li, O. Frieder, “Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks”, Proc. INFOCOM, 2001.
4) Ivan Stojmenovic, Mahtab Seddigh and Jovisa Zunic, “Internal Nodes Based Broadcasting in Wireless Networks”, Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.
5) Robert J. Marks II, Arindam K. Das, Mohamed El-Sharkawi, Payman Arabshahi & Andrew Gray, “Minimum Power Broadcast Trees for Wireless Networks: Optimizing Using the Viability Lemma”, Proceedings of the IEEE International Symposium on Circuits and Systems, Scottsdale, Arizona, 2002.
6) Arindam K. Das, R.J. Marks II, M.A. El-Sharkawi, P. Arabshahi and A. Gray, “The Minimum Power Broadcast problem in Wireless Networks: An Ant Colony System Approach,” Proceedings of the IEEE CAS Workshop on Wireless Communications and Networking, Pasadena, CA, Sept. 5-6, 2002.
7) Cartigny, J., Simplot, D., and Stojmenovic, I., “Localized minimum-energy broadcasting in ad- hoc networks,” Proc. INFOCOM, 2003.
8) Arindam K. Das, R.J. Marks II, M.A. El-Sharkawi, P. Arabshahi and A. Gray, “Minimum Power Broadcast Trees for Wireless Networks: Integer Programming Formulations,” Proc. INFOCOM, 2003.
9) Arindam K. Das, “Minimum Power Broadcast and Maximization of Time-to-First-Failure in Broadcast Wireless Networks,” Ph.D dissertation, University of Washington, 2003.
10) Arindam K. Das, R.J. Marks II, M.A. El-Sharkawi, P. Arabshahi and A. Gray, “A Cluster-Merge algorithm for solving the minimum power broadcast problem in large scale wireless networks,” Proc. MILCOM, Boston, MA, Oct. 13-16, 2003.
11) Jonathan Gross and Jay Yellen, “Graph Theory and its Applications”, CRC Press, 1999.
12) LPSOLVE LP solver, downloadable from <ftp://ftp.es.ele.tue.nl/pub/lp_solve/>