Minimum Power Broadcast Trees for Wireless Networks: Optimizing Using the Viability Lemma

Arindam K. Das, Robert J. Marks, Mohamed El-Sharkawi
Department of Electrical Engineering
University of Washington
Box 352500
Seattle, WA 98195.

Payman Arabshahi, Andrew Gray
Jet Propulsion Laboratory
4800 Oak Grove Drive, MS 238-343
Pasadena, CA 91109.

***Abstract*— Wireless multicast/broadcast sessions, unlike wired networks, inherently reaches several nodes with a single transmission. For omnidirectional wireless broadcast to a node, all nodes closer will also be reached. An algorithm for constructing the minimum power tree in wireless networks was first proposed by Wieselthier** $et\ al$**. The** $broadcast\ incremental\ power$ **(BIP) algorithm suggested by them is a "node-based" minimum-cost tree algorithm for wireless networks. We propose an alternate search based paradigm wherein minimum-cost trees in wireless networks are found through a search process. Two computationally efficient procedures for checking the feasibility (viability) of a solution in the search space are presented. A straightforward procedure for initializing the search using stochastically generated trees is also proposed.**

I. INTRODUCTION

For a given node constellation with an identified source node, the minimum power wireless broadcast problem is to communicate to all remaining nodes, either directly or hopping, such that the overall instantaneous transmission power consumed by the network is minimized. Although previous work in this area focuses on a "link-based solution", Wieselthier $et\ al$ [1] note that a "node based" approach is needed for wireless environments. Recently, an internal nodes based broadcasting procedure was suggested by Stojmenovic $et\ al$ [5].

Our approach, dubbed $optimizing\ using\ the\ viability\ lemma$ (OVL), relies on optimization search among the large number of possible node power settings for a given node constellation. Transmitter power levels at each node dictate the nodes to which it can communicate. Constructing a minimum-power broadcast tree is equivalent to finding the node power vector which minimizes the sum of the transmitter powers, subject to the constraint that the node power levels are sufficient to allow a $viable\ connection\ tree$. A viable connection tree is one where the source is able to reach all intended destination nodes, either directly or using other nodes in the network. Not all combinations of node powers allow viable connection trees. Those that do are called $viable$ power cuts. We discuss two computationally efficient methods for checking the viability of a power cut. Establishing cut viability is essential in the search process.

We assume a fixed 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. Nodes that receive a transmission but do not retransmit it are classified as $leaf nodes$. Nodes that transmit, including the source node, are called $hop nodes$. The remaining nodes are $unconnected$. For the broadcast problem, any tree which has unconnected nodes is not viable. All nodes in the network are assumed to have omnidirectional antennas, so that if node m transmits to node n, all nodes closer to m than n will also receive the transmission. This is the "wireless multicast advantage" [1].

For a transmission from node m to n, separated by a distance r_{mn}, the transmitter power at m is modeled to be proportional to r_{mn}^{α} where α is the channel loss exponent (typically between 2 and 4, depending on the channel medium). Throughout this paper, we assume $\alpha = 2$. Without any loss of generality, we can set the proportionality constant to one, so that the transmitter power, p_T, is given by: $p_T = r_{mn}^{\alpha}$.

For illustrative purposes, we find useful an alphabetic node-numbering system ($e.g.$, nodes A, B, C, etc.). However, for computational purposes, the numeric equivalents ($i.e.$, '1' representing node A, '2' representing node B, etc) are used for the node indices.

II. RELATED WORK

Wieselthier, Nguyen and Ephremides [1] proposed the BIP algorithm for constructing the minimum-power tree for 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. Figure 1 shows five nodes randomly distributed in a 5×5 square grid and the minimum-power broadcast tree constructed using the BIP algorithm. Node C is the source. The solid lines in the figure represent actual transmissions and the dashed lines represent implicit transmissions.

Assuming $\alpha = 2$, the power matrix, $\mathbf{P}$, of the above network is:

$$\mathbf{P} = \begin{bmatrix} 0 & 18.23 & 8.01 & 6.05 & 9.55 \\ 18.23 & 0 & 7.40 & 14.06 & 10.30 \\ 8.01 & 7.40 & 0 & 1.17 & 16.73 \\ 6.05 & 14.06 & 1.17 & 0 & 20.52 \\ 9.55 & 10.30 & 16.73 & 20.52 & 0 \end{bmatrix} \quad (1)$$

In general, $\mathbf{P}[m, n]$, the power required for node m to transmit to node n is given by:

$$\mathbf{P}[m, n] = \left[(x_m - x_n)^2 + (y_m - y_n)^2\right]^{\alpha/2}$$

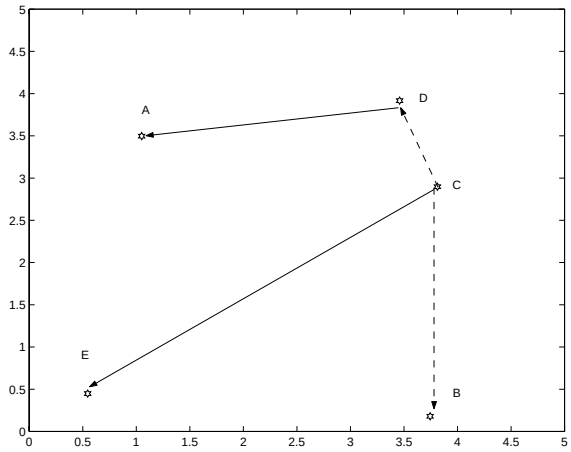

Fig. 1. BIP Tree: node C is the source

where $\{(x_i, y_i) : 1 \leq i \leq N\}$ are the coordinates of the nodes in the network.

The total broadcast power of the tree in Figure (1) is 22.78 Watts ($\mathbf{P}[3,5] + \mathbf{P}[4,1]$). Clearly, this is not optimal since the minimum power required to reach all the nodes in the network is 16.73 Watts, involving the transmission $C \rightarrow E$. Since all the other nodes (B, A, D) are nearer to C than E, these will be reached implicitly.

III. TERMINOLOGY

Prior to presentation of OVL, we offer the following definitions.

1. $Rank\ Matrix$: The rank matrix $\mathbf{R}$ is defined as the sorted power matrix. Each row of the power matrix ($\mathbf{P}$) is sorted in ascending order to give the matrix $\mathbf{R}$. For the power matrix in (1), the rank matrix is given by:

$$\mathbf{R} = \begin{bmatrix} 0 & 6.05 & \mathbf{8.01} & 9.55 & 18.23 \\ 0 & 7.4 & 10.30 & \mathbf{14.06} & 18.23 \\ 0 & 1.17 & 7.40 & 8.01 & \mathbf{16.73} \\ \mathbf{0} & 1.17 & 6.05 & 14.06 & 20.52 \\ 0 & \mathbf{9.55} & 10.30 & 16.73 & 20.52 \end{bmatrix} \tag{2}$$

2. $Cut\ Vector$: A cut vector, $\vec{\chi}_R$, referenced to the rank matrix, is an N-element vector (where N is the number of nodes in the network) with integers between 1 and N. It indicates the location of an element on each row of the rank matrix. For example, if we want to reference the highlighted elements in (2), the corresponding cut vector is: $\vec{\chi}_R = [3\ 4\ 5\ 1\ 2]^T$, where the superscript 'T' stands for transpose.

Alternately, we can use the power matrix $\mathbf{P}$ and define an equivalent cut vector, $\vec{\chi}_P$, referenced to $\mathbf{P}$. Mapping the highlighted elements in (2) to the power matrix (1), we have the equivalent cut vector $\vec{\chi}_P = [3\ 4\ 5\ 4\ 1]^T$. Since all diagonal elements in $\mathbf{P}$ are zero, the condition $\vec{\chi}_P[n] = n$ implies that node n is a non-transmitting node; $i.e.$, it is either an unconnected node or a leaf node in the tree.

3. $Threshold\ Vector$: A threshold vector, $\vec{t}$, of length N, is a vector of the elements of $\mathbf{R}$ specified by the cut vector $\vec{\chi}_P$. For the cut vectors in Item 2 above, the threshold vector is given by:

$$\vec{t} = [8.01\ 14.06\ 16.73\ 0\ 9.55]^T$$

If the n^{th} element of the threshold vector is zero, it implies that node n is non-transmitting. We thus have the equivalence condition:

$$\vec{t}[n] = 0 \Longleftrightarrow \vec{\chi}_P[n] = n$$

4. $The\ cost\ of\ a\ cut$: The cost of a cut is defined as the sum of the elements of the corresponding threshold vector.

$$s[\vec{\chi}_P] = \sum_{n=1}^{N} \vec{t}[n] \tag{3}$$

5. $Viability\ of\ a\ cut$: A cut is said to be $viable$ if it allows all intended destination nodes to be reached. Otherwise, it is $nonviable$. For example, if the source node (C) in Figure (1) wants to broadcast to all other nodes, the cut vector $\vec{\chi}_P = [1\ 2\ 5\ 4\ 5]^T$ is viable but $\vec{\chi}_P = [1\ 2\ 4\ 1\ 5]^T$ is not. In the first case, all nodes are reached by a single transmission from C, whereas only D and A are reached in the latter case.

6. $The\ Transfer\ Matrix$: For a given network, the transfer matrix, $\mathbf{H}$, is a function of the threshold vector. The transfer matrix is computed by thresholding the power matrix as follows:

$$\mathbf{H}[m,n] = \begin{cases} 1, & \text{if } \mathbf{P}[m,n] \leq t[m] \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The set of nodes which can be reached from node m, transmitting at a power level $\vec{t}[m]$, is given by the column indices of the m^{th} row of $\mathbf{H}$ for which $\mathbf{H}[m,n] = 1$.

IV. THE TRANSFER MATRIX POWER FORM OF THE VIABILITY LEMMA

Randomly generated cut vectors do not necessarily correspond to viable connection trees. The $viability\ lemma$ provides a straightforward way of determining whether a cut is viable or not.

For a given node constellation and cut vector $\vec{\chi}_P$, with a corresponding threshold vector $\vec{t}$, we apply the iteration

$$\vec{\gamma}^{(k+1)} = \mathbf{H}^T \otimes \vec{\gamma}^{(k)} \tag{5}$$

where k is the iteration index and $\otimes$ denotes a matrix product with additions replaced by logical OR operations and multiplications replaced by logical AND's. $\vec{\gamma}^{(k)}$ is a binary $coverage\ vector$ which keeps track of the nodes reached till the k^{th} iteration. (A $reached\ node$ is either a hop node or a leaf). If the n^{th} element of $\vec{\gamma}^{(k)}$ is '1', it indicates that node n has been reached, in the k^{th} iteration or a previous iteration. Nodes that have not been reached till the k^{th} iteration are tagged by a '0' in

the coverage vector. The iteration proceeds with the initialization $\vec{\gamma}^{(0)} = [0 \cdots 0\ 1\ 0 \cdots 0]^T$ which is a zero vector of length N with a '1' at the source node. [1]

The transfer matrix power form of the viability lemma states that a necessary and sufficient condition for a cut to be viable is:

$$\vec{\gamma}^{(N-1)} = \vec{1} \tag{6}$$

where $\vec{1}$ is a vector of 1's of length N. Equation (6) recognizes that, for a network with N nodes and one source node, the iteration (5) will converge in, at most, N-1 iterations. If the cut is viable, all nodes will be reached resulting in a vector of all ones. Using equation (5), (6) can be equivalently expressed as:

$$\mathbf{H}_{N-1} \otimes \vec{\gamma}^{(0)} = \vec{1} \tag{7}$$

where

$$\mathbf{H}_N = \left[\mathbf{H}^{\otimes N}\right]^T$$

Determination of cut viability using the viability lemma does not immediately present a connection tree. Viability simply says that total coverage is possible and such a tree exists. Section VI explains how to grow a routing tree from the transfer matrix for a viable cut.

The number of Boolean matrix multiplies in (7) is on the order of $\log_2(N-1)$. $\mathbf{H}_4$ is the square of $\mathbf{H}_2$, $\mathbf{H}_8$ is the square of $\mathbf{H}_4$, etc. The transfer matrix parsing form of the viability lemma, described next, requires no matrix multiplies.

V. THE TRANSFER MATRIX PARSING FORM OF THE VIABILITY LEMMA

An alternate procedure for determining the viability of a cut vector uses parsing of the transfer matrix. For broadcast applications in an N-node network with one source node, the procedure converges in at most $N-1$ iterations. We begin by defining the following sets:

$\mathbf{S}$	= destination nodes
k	= iteration number
$\mathbf{NR}^{(k)}$	= $new\ nodes$ reached in iteration k
$\mathbf{NR}^{(0:k)}$	= nodes reached till iteration k = $\bigcup_{m=0}^{k} \mathbf{NR}^{(m)}$
$\mathbf{NNR}^{(k)}$	= nodes not reached at the end of iteration k

Note that $\mathbf{NNR}^{(k)} = \mathbf{S} \setminus \mathbf{NR}^{(0:k)}$ where '$\setminus$' denotes the $set\ difference$ operation. Also, it follows from the definition of $\mathbf{NR}^{(k)}$ that $\mathbf{NNR}^{(k-1)} \supseteq \mathbf{NR}^{(k)}$. The sets defined above can be expressed equivalently in terms of the coverage vector $\vec{\gamma}^{(k)}$.

$$\begin{aligned}
\mathbf{NR}^{(0:k)} &\equiv \{n : \vec{\gamma}^{(k)}[n] = 1\} \\
\mathbf{NNR}^{(k)} &\equiv \{n : \vec{\gamma}^{(k)}[n] = 0\} \\
\mathbf{NR}^{(k)} &\equiv \{n : \vec{\gamma}^{(k)}[n] - \vec{\gamma}^{(k-1)}[n] = 1\}
\end{aligned}$$

[1] Note that

$$\mathbf{H}^T \otimes \vec{\gamma}^{(k)} = u\left(\mathbf{H}^T \vec{\gamma}^{(k)}\right)$$

where the unit step, $u(\cdot)$, equals zero for a negative argument and is otherwise one.

The viability checking procedure is as follows:

1. For $k = 0$, initialize $\mathbf{NR}^{(0)} = \{source\ node\}$.
2. For each subsequent iteration, $1 \le k \le N-2$, check whether any of the node(s) reached for the first time in the previous iteration (the entries in the set $\mathbf{NR}^{(k-1)}$) connect to a new destination node(s) (the entries in the set $\mathbf{NNR}^{(k-1)}$). This is easily done by examining the corresponding rows of the transfer matrix as mentioned in item (F), Section III.[2] If none of the nodes in $\mathbf{NR}^{(k-1)}$ connect to a new node from the set $\mathbf{NNR}^{(k-1)}$, the connection tree "breaks" at the k^{th} iteration and the cut is $nonviable$. Conversely, if all nodes in $\mathbf{NNR}^{(k-1)}$ are $jointly$ reached by the nodes in $\mathbf{NR}^{(k-1)}$ ($\Rightarrow \mathbf{NNR}^{(k)} = \emptyset$, where $\emptyset$ is the null set), the connection tree is complete and the cut is $viable$. In both these cases, the iteration process terminates after iteration k. If only some of the nodes in the set $\mathbf{NNR}^{(k-1)}$ are reached by the nodes in $\mathbf{NR}^{(k-1)}$ during iteration k, update the sets $\mathbf{NR}^{(0:k)}$ and $\mathbf{NNR}^{(k)}$ and continue with the next iteration.
3. If the iteration process continues till $k = N-1$, check whether $\mathbf{NNR}^{(N-1)} = \emptyset$. If so, the cut vector is $viable$. Otherwise, it is $nonviable$.

VI. FINDING ROUTING FROM A VIABLE CUT

All connections in a routing tree can be expressed as an $N \times N$ binary $connection\ matrix$, $\Phi = \{\phi[m,n] : 1 \le m,n \le N\}$. The matrix can be constructed from a tree or $vice\ versa$. If $\phi[m,n] = 1$, the node responsible for communicating to node n is m. For a fully connected tree, each column of Φ should therefore contain a single one and the rest of the elements should be zeros. The connection matrix is built iteratively, using equation (5). Let $\Phi^{(k)}$ be the connection matrix corresponding to the k^{th} iteration, the initialization being:

$$\Phi^{(0)}[n,m] = \begin{cases} 1, & \text{if } n = m = \text{source node index} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Let $\vec{\beta}^{(k)}$ be a $difference\ vector$ as defined below:

$$\vec{\beta}^{(k)} = \vec{\gamma}^{(k)} - \vec{\gamma}^{(k-1)}, \quad \vec{\beta}^{(0)} = \vec{\gamma}^{(0)} \tag{9}$$

The vector $\vec{\beta}^{(k)}$ has 1's corresponding to the nodes reached during the k^{th} iteration.[3] If an element in $\vec{\beta}^{(k)}$ is 1, it must have originated from a node corresponding to a 1 in $\vec{\beta}^{(k-1)}$. For example, let $\vec{\beta}^{(k)}[n]$ and $\vec{\beta}^{(k-1)}[m]$ be equal to 1. If $\mathbf{H}[m,n] = 0$ ($\Rightarrow \vec{t}[m] < \mathbf{P}[m,n]$), node m cannot connect to node n. If, however, $\mathbf{H}[m,n] = 1$ ($\Rightarrow \vec{t}[m] \ge \mathbf{P}[m,n]$), node m can connect to node n and we set $\Phi^{(k)}[m,n] = 1$.

If there is more than one node (say m_1 and m_2) which can connect to a destination node (say node n) in iteration k, the node which is closest to n (least cost involved) is chosen as the transmitting node. For example, if m_1 is closer to n than m_2,

[2] For example, if $\mathbf{NR}^{(k-1)} = \{m,n\}$, check the m^{th} and n^{th} rows of the transfer matrix in iteration k. If $\mathbf{H}[m,p] = 1$, $\mathbf{H}[m,q] = 1$, $\mathbf{H}[n,r] = 1$, $\mathbf{H}[n,s] = 1$, $\{p,s\} \in \mathbf{NR}^{(0:k-1)}$ and $\{q,r\} \notin \mathbf{NR}^{(0:k-1)}$, the set of new nodes jointly reached in iteration k ($\mathbf{NR}^{(k)}$) is $\{q,r\}$.

[3] If $\vec{\beta}^{(k)} = \vec{0}$, the iteration has converged.

we set $\Phi^{(k)}[m_1, n] = 1$. If more than one node (say n_1 and n_2) is reached in the k^{th} iteration from a single transmitting node (say node m), we set $\Phi^{(k)}[m, n_1] = 1$ and $\Phi^{(k)}[m, n_2] = 1$. Transmitting to multiple nodes should be interpreted as a single transmission from the source to the farthest node, with the other nodes being reached implicitly.

VII. STOCHASTIC TREE GENERATION

The $stochastic\,tree\,generation$ method is used to generate a set of viable cut vectors, which can then be used to initialize a search for the optimum tree. It is an iterative procedure which starts with a transmission from the source node to a randomly chosen destination node and continues till all the intended destination nodes are reached. This implies that for a broadcasting session in an N-node network with a source node, the iteration must converge in at most $N - 1$ iterations. If the transmitting node in iteration k is m and the destination node is n, the k^{th} update of the cut vector is: $\vec{\chi}_P^{(k)}[m] = n$. Choice of transmitting and destination nodes for each iteration is dictated by the following heuristics.

1. For $k = 0$, initialize $\vec{\chi}_P^{(0)}[m] = m, 1 \leq m \leq N$; $i.e.$, all nodes are initially set as non-transmitting nodes in the tree.
2. For $k = 1$, the source node is the transmitting node. For $k \geq 2$, the node chosen for transmission in iteration k should be a leaf node in the tree by iteration $k - 1$. If there are more than one leaves in the tree at the end of iteration $k - 1$, the transmitting node for the k^{th} iteration is chosen randomly from this set.
3. $Generally$, the transmitting node in the k^{th} iteration can either choose to be a leaf in the tree or transmit to a node drawn randomly from the set of nodes which have not been reached by the end of iteration $k - 1$. This can be implemented by augmenting the set of unreached nodes with the index of the transmitting node and then choosing an element from the augmented set at random. The option to stay in the tree as a leaf is withdrawn if (1) $k = 1$, since the source node cannot be a leaf, or, (2) if there is only one possible choice of the transmitting node for iteration k ($k \geq 2$) and not all destination nodes have been reached by iteration $k - 1$.
4. If there are multiple unreached nodes at the start of the final iteration, the transmitting node must transmit at a power level sufficient to reach the farthest unreached node. This would ensure that all remaining nodes are covered and the cut vector is viable.

Since no matrix operations of the type $\mathbf{H}^{\otimes N}$ are involved, this method can be particularly useful for generating random viable cuts in large networks. Also, since the transmitting node always chooses its destination node from the set of nodes not reached by the end of the previous iteration, connection trees developed from cuts generated using this method will be inherently loop-free.

VIII. OPTIMIZATION

The foundation for OVL is now established. Stochastic tree generation is used to initiate the search. The search seeks to minimize the cost of a cut subject to the cut being viable.

Numerous search algorithms exist which can be applied to find the optimal cut. The performance of different evolutionary search techniques, in terms of improvement in tree power and convergence speed, is currently under investigation and will be discussed in a subsequent paper. Preliminary simulation tests on 50 randomly generated 25-node networks in a 5×5 grid using a genetic search algorithm with no mutation show an average improvement in tree power of approximately 10.4% over the BIP algorithm. The search was done using 100 initial viable solutions and 50 evolutions.

IX. CONCLUSION

We have proposed a paradigm to search for minimum power broadcast trees in wireless networks. Preliminary experimental results indicate that such techniques can generate better solutions than that provided by the BIP algorithm, albeit at a higher computational cost. The procedure can be straightforwardly generalized to multicast sessions and points-to-points communication.

REFERENCES

1) J.E.Wieselthier, G.D. Nguyen and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks", *IEEE INFOCOM* 2000, pp. 585- 594.
2) D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs: Prentice Hall, 1992.
3) T.M. Cover, "The best two independent measurements are not the two best", *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-4, pp.116-117, January 1974.
4) Russell D. Reed and R.J. Marks II, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, MIT Press, Cambridge, MA, 1999.
5) 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.