MDLT: A Polynomial Time Optimal Algorithm for Maximization of Time-to-First-Failure in Energy Constrained Wireless Broadcast Networks

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

Abstract—We consider the problem of maximizing the time-to-first-failure, defined as the time till the first node in the network runs out of battery energy, in energy constrained broadcast wireless networks. We discuss a greedy algorithm and prove that it solves the problem optimally for a broadcast application, in polynomial time, provided the complete power matrix and the battery residual capacities are known.

I. Introduction

We consider the problem of maximizing the time-to-first-failure in energy constrained broadcast wireless networks where each node is powered by batteries. In applications where replacement/maintenance of such batteries is difficult or infeasible, it is of utmost importance to design routing protocols which maximize the $lifetime$ of the network. A metric commonly used to define the lifetime of a network is the duration of time before any node in the network runs out of its battery energy. We define this time to be the $time-to-first-failure$ (TTFF), also known as $system\ lifetime$ or $network\ lifetime$ in the literature. To the best of our knowledge, this problem was first addressed by Chang and Tassiulas in [1]. Subsequent research in this area for point-to-point as well as multicast applications have been reported in [2], [3], [4] and [5]. In this paper, we consider a broadcast application and show that maximization of the TTFF for such an application can be solved optimally by a greedy algorithm in polynomial time.

II. Network Model

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. 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. The power matrix of the network, $\mathbf{P}$, is defined to be an $N \times N$ symmetric matrix whose $(i,j)th$ element represents the power required for node i to transmit to node j and is given by:

$$\mathbf{P}_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2\right]^{\alpha/2} = d_{ij}^{\alpha}, \quad i \neq j \qquad (1)$$

where $\{(x_i, y_i) : 1 \leq i \leq N\}$ are the coordinates of the nodes in the network, α $(2 \leq \alpha \leq 4)$ is the channel loss exponent and d_{ij} is the Euclidean distance between nodes i and j. We assume that there is no 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 its $(i,j)th$ element is ∞ if d_{ij}^{α} is greater than the maximum power limit of node i.

We consider a centralized implementation where construction of the routing tree is done at the source node, which has complete knowledge of the locations of all nodes in the network as well as their residual energies.

Finally, we assume that power expenditures due to signal reception and processing are negligible compared to signal transmission and hence the lifetime is determined solely by the choice of transmitter power levels and residual energy levels at the nodes.

III. Problem Statement

Let $\vec{E}(t)$ be a vector of node residual energies at time t, the ith element of $\vec{E}(t)$ representing the residual energy of node i, and $\vec{Y}$ be a vector of node transmission powers. Assuming that all nodes in the network have omni-directional antennas, a transmission from node i to node j would also be received by all nodes geometrically closer to i than j. Let S be the set of nodes that are geometrically closer to i than j ($\Rightarrow \mathbf{P}_{ij} > \mathbf{P}_{ik} : \forall k \in S$). Nodes that belong in S are said to receive the transmission from i **implicitly** (in the sense that no additional cost is incurred to reach them) and the set of transmissions $\{i \rightarrow k : \forall k \in S\}$ are referred to as **implicit transmissions**. The transmission $i \rightarrow j$ is referred to as an **actual transmission**.

Following our discussion in the previous paragraph, we can write:

$$\vec{Y}_i = \max_j \{X_{ij}\mathbf{P}_{ij} : j \neq i\} \qquad (2)$$

where $X_{ij} = 1$ if node j is reached from node i (actually or implicitly) and 0 otherwise. Equation (2) implies that the cost of spanning in multiple nodes from node i is simply the cost incurred in reaching the farthest node.

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

Defining $L_i(t) \triangleq \vec{E}_i(t)/\vec{Y}_i$ to be the *lifetime of node* i, the problem of maximizing the TTFF can be written as:

$$maximize\,\{\min_{i\in V} L_i(t)\} \qquad (3)$$

where V is the set of all nodes in the network. The objective function in (3) is to be optimized subject to the following constraints:

1) All nodes, other than the source, must be reached, either actually or implicitly[1].
2) The source node must reach at least one other node.
3) The tree must be *connected*. We define *connectivity* formally in Section IV.
4) The tree must not have any *cycles*.

The vector $L(t) \triangleq \{L_i(t) : \forall i \in V\}$ is the *node lifetime vector* at time t. Note that the value of the expression within curly braces in (3) is dependent on the time index t and hence, strictly speaking, should be termed $residual - time - to - first - failure$. However, we will refer to it simply as the time-to-first-failure, implicitly recognizing its dependence on the time origin t. Accordingly, henceforth in this paper, we will simply use the notation $\vec{E}_i$ instead of $\vec{E}_i(t)$.

IV. Notations and Definitions

We first establish the following notation.

V	= set of all nodes in the network
$\vec{tr}(T)$	= set of transmitting nodes in the tree T
k	= iteration number
$\mathbf{NR}^k$	= all nodes reached till iteration k
$\mathbf{NNR}^k$	= nodes not reached at the end of iteration k $\triangleq V \setminus \mathbf{NR}^k$
T^k	= connection tree grown till iteration k
τ^k	= lifetime of the partially grown tree T^k $\triangleq min_i\left(E_i/Y_i : \forall i \in \vec{tr}(T^k)\right)$
cr_node^k	= critical node of the connection tree grown till iteration k

The sets $\mathbf{NR}^k$ and $\mathbf{NNR}^k$ satisfy the following properties:

$$\mathbf{NR}^k \cup \mathbf{NNR}^k = V,\;\; \mathbf{NR}^k \cap \mathbf{NNR}^k = \emptyset \qquad (4)$$

For a given connection tree, T, we define its *critical node* to be the node whose lifetime is equal to the TTFF of the tree. That is:

$$\text{Critical node } = \text{argmin}_i\left(\vec{E}_i/\vec{Y}_i : i \in \vec{tr}(T)\right) \qquad (5)$$

Note that only the set of transmitting nodes in T can be considered in (5) since the residual lifetime of all non-transmitting nodes is ∞.

A transmission $(i \to j)$[2] is defined to be the *critical transmission* in a tree if:

$$\vec{E}_i/\mathbf{P}_{ij} = \text{TTFF } \triangleq \min_i\left(\vec{E}_i/\vec{Y}_i : i \in \vec{tr}(T)\right) \qquad (6)$$

[1]Note that the possibility of reaching a node implicitly is a consequence of the inherently broadcast nature of the wireless network and our assumption of omni-directional antennas.

[2]In this paper, we will use the notation $(i \to j)$ and (i, j) interchangeably to mean a directed edge from i to j. The notation $\{i, j\}$ will be used to refer to the node pair.

We conclude this section with definitions of the *connectivity* and *validity* of a tree. We define a tree to be *connected* if the transmitting node at each iteration k has been reached by iteration $k - 1$. In other words, the transmitting node at iteration k must belong to the set $\mathbf{NR}^{k-1}$. A broadcast tree is *valid* if it is connected and reaches all destination nodes.

V. The MDLT Algorithm

In this section, we explain the $minimum\ decremental\ lifetime$ (MDLT) algorithm for optimizing the objective function in (3). It is an iterative algorithm wherein one new node is spanned into the tree in each iteration. For $k = 0$, we initialize:

$$\mathbf{NR}^0 = \{\text{source}\}\;,\; \mathbf{NNR}^0 = \{V \setminus \text{source}\} \qquad (7)$$

For any $k \geq 1$, a list of candidate edges is prepared and an edge is chosen which minimizes the decremental lifetime (defined subsequently) due to addition of that edge. The set of possible edges to choose from at iteration k (denoted by $edge\_list^k$) is given by:

$$edge\_list^k = \{(i,j) : \forall i \in \mathbf{NR}^{k-1}, \forall j \in \mathbf{NNR}^{k-1}, \vec{E}_i/\mathbf{P}_{ij} \geq N_b/D\} \qquad (8)$$

where N_b is the total number of bits to be transmitted during the broadcast session[3] and D is the data rate. The first two conditions in (8) allow transmissions from any node which has been spanned into the tree by iteration $k - 1$, to any node not yet spanned in. The third condition in (8), $\vec{E}_i/\mathbf{P}_{ij} \geq N_b/D$, prevents nodes which lack sufficient battery capacity to support a broadcast session from being a transmitting node in the broadcast tree. For example, suppose that the edge $i \to j$ is chosen at the kth iteration. Let $\vec{E}_i = 5$ and $\mathbf{P}_{ij} = 50$. The residual lifetime of node i, transmitting at a power level of 50, is given by: $\vec{E}_i/\mathbf{P}_{ij} = 5/50 = 0.1$. Assuming that the number of data bits to be transmitted during the broadcast session is 10000, at a rate of 50000 bps, the required transmission time is $10000/50000 = 0.2$, which is more than the residual lifetime of node i. Clearly, node i will burn out in the middle of the session if (i, j) is chosen as a routing edge in the connection tree.

The edge (i, j) is chosen to be included in the connection tree at iteration k if it satisfies the MDLT criterion:

$$\tau^{k-1} - \vec{E}_i/\mathbf{P}_{ij} < \tau^{k-1} - \vec{E}_m/\mathbf{P}_{mn};\; (m,n) \neq (i,j),\; (i,j),(m,n) \in edge\_list^k \qquad (9)$$

or, equivalently,

$$\frac{\vec{E}_i}{\mathbf{P}_{ij}} > \frac{\vec{E}_m}{\mathbf{P}_{mn}};\; (i,j),(m,n) \in edge\_list^k, (m,n) \neq (i,j) \qquad (10)$$

[3]Assuming connection-oriented systems. The case of connectionless systems can be handled by redefining N_b to be the packet length, so that the residual lifetime of any node is greater than or equal to the packet transmission time.

The expression on the left hand side of (9) is the decremental lifetime at iteration k due to addition of the edge (i, j). We assume that tie breaking is not necessary, *i.e.*, there is only one edge, (i, j), which satisfies (10).[4]

The lifetime of the connection tree after inclusion of the edge (i, j) at iteration k is given by:

$$\tau^k = \min\left(\tau^{k-1}, \vec{E}_i/\mathbf{P}_{ij}\right) : (i, j) \in edge\_list^k \tag{11}$$

The implicit assumption in (11) is that τ^k is a monotonically decreasing function of k; *i.e.*, inclusion of an edge at the kth iteration can never "improve" the tree constructed till iteration $k-1$. If this is not true, a better solution can be readily obtained by deleting the critical transmission from the tree, resulting in a higher TTFF, without affecting the validity of the tree. We clarify with an example. Suppose, we are given the tree

$$\{1 \to 3;\ 1 \to 2;\ 2 \to 4;\ \mathbf{3} \to \mathbf{5};\ 4 \to 6\}, \tag{12}$$

the critical transmission being $3 \to 5$. Assume that the edge chosen next is $6 \to 7$, $E_6/\mathbf{P}_{67} > E_3/\mathbf{P}_{35}$, and $\mathbf{P}_{65} < \mathbf{P}_{67}$ (*i.e*, node 5 is nearer to node 6 than 7 and hence receives the transmission $6 \to 7$ implicitly). Under these assumptions, an improved tree can be obtained by deleting the critical transmission from (12) and appending the current edge to the previous tree, as shown below.

$$\{1 \to 3;\ 1 \to 2;\ 2 \to 4;\ 4 \to 6;\ 6 \to 7\} \tag{13}$$

Note that the tree in (13) is still valid because it is connected and reaches all nodes. Later in Section VI, where we argue that our assumption holds and prove the optimality of the MDLT algorithm, we will show that the tree in (12) cannot be part of the optimal solution.

A high level description of the MDLT algorithm[5] is provided in Figure 1. We conclude this section with a discussion of the time complexity of the algorithm, assuming a full power matrix (*i.e.*, there is no maximum transmitter power constraint). First, we note that, at any iteration k, we do not need to consider all edges whose tails are in the set $\mathbf{NR}^{k-1}$ and heads are in the set $\mathbf{NNR}^{k-1}$. For any node $i \in \mathbf{NR}^{k-1}$, we only need to identify the node closest to i which is in the set $\mathbf{NNR}^{k-1}$. Completing this step for all nodes in $\mathbf{NR}^{k-1}$ and choosing the best edge can be accomplished in $O(|V|)$ time, if proper data structures (see [6], for example) such as those used in an efficient implementation of Prim's algorithm for the Minimum Spanning Tree problem are used. Since the algorithm runs for $|V|-1$ iterations, the overall time complexity of the algorithm is $O(|V|^2)$.

1. Let DS be the intended destination set.
2. Set $k = 0$; $T^k = \emptyset$;
3. Set $\mathbf{NR}^{(k)} = \{source\}$ and $\mathbf{NNR}^{(k)} = V \setminus \mathbf{NR}^{(k)}$;
4. Set $k = k + 1$;
5. /* Let $[m^k, n^k]$ be the edge added to the tree during the kth iteration where m^k is the transmitting node and n^k is the destination node. */
6. **while**($DS \neq \emptyset$)

    /* Select the next edge according to the MDLT criterion (See equations 9 or 10) */

    • $[m^k, n^k] = \text{argmax}_{i,j}\{\vec{E}_i/\mathbf{P}_{ij} : \forall i \in \mathbf{NR}^{k-1}, \forall j \in \mathbf{NNR}^{k-1}, \vec{E}_i/\mathbf{P}_{ij} \geq N_b/D\}$;

    /* Updates */

    • $\mathbf{NR}^k = \mathbf{NR}^{k-1} \bigcup n^k$, $\quad \mathbf{NNR}^k = V \setminus \mathbf{NR}^k$;

    /* Append chosen edge to current tree. */

    • $T^k = [T^{k-1};\ \{m^k \to n^k\}]$;

    /* Compute the lifetime of the partially constructed tree and identify its critical node */

    • $\tau^k = \min\{\tau^{k-1}, \vec{E}_{m^k}/\mathbf{P}_{m^k n^k}\}$;

    • cr_node^k = critical node of T^k;

    if($n^k \in DS$) /* Destn. node reached. Update DS. */

        $DS = DS \setminus n^k$;

    endif

    if($DS == \emptyset$) /* All destination nodes reached. */

        /* End of procedure. */

        • $\hat{L} = \tau^k$; /* TTFF $= \tau^k$ */

        • Print T^k and $\hat{L}$;

    else

        $k = k + 1$; /* Increment k */

    endif

   endwhile

Fig. 1. High level description of the MDLT algorithm.

VI. Optimality of the MDLT algorithm

We first show that the MDLT algorithm is optimal for any 3-node network. Subsequently, we extend the proof to any N-node network. Consider the 3-node network in Figure 2. Assume node 1 is the source and $\mathbf{P}_{12} < \mathbf{P}_{13}$. The only possible broadcast trees for this network are: (1) $T_1 = \{1 \to 2; 2 \to 3\}$ and (2) $T_2 = \{1 \to 3\}$. Note that T_2 is a valid broadcast tree since it also reaches node 2 implicitly.[6]

Case 1 : If T_1 is optimal, we have:

$$\min\left(E_1/\mathbf{P}_{12}, E_2/\mathbf{P}_{23}\right) > E_1/\mathbf{P}_{13} \tag{14}$$

$$\Rightarrow \text{(a) } E_1/\mathbf{P}_{12} > E_1/\mathbf{P}_{13} \text{ (since } \mathbf{P}_{12} < \mathbf{P}_{13}) \tag{15}$$

$$\text{(b) } E_2/\mathbf{P}_{23} > E_1/\mathbf{P}_{13} \tag{16}$$

It is apparent from the MDLT criterion (10) that the first edge to be included in the tree will be (source $\to$ node nearest to source), which, for our example, is $1 \to 2$. The next edge that will be chosen is determined by the following decision rules:

$$\text{Choose edge } 1 \to 3 \ \text{ if } E_1/\mathbf{P}_{13} > E_2/\mathbf{P}_{23} \tag{17}$$

$$\text{Choose edge } 2 \to 3 \ \text{ if } E_2/\mathbf{P}_{23} > E_1/\mathbf{P}_{13} \tag{18}$$

[4]This will be the case if all the elements of the upper triangular portion of the matrix, $\mathbf{W}$, whose $(i, j)th$ element is given by $E_i/\mathbf{P}_{ij}$, are distinct.

[5]Even though we focus on broadcast applications in this paper, the MDLT algorithm is general and can be used for broadcast or multicast.

[6]Strictly speaking, $\{1 \to 3; 3 \to 2\}$ is also a valid broadcast tree, but it is worse than T_2 since it involves unnecessary energy expenditure at node 3. In other words, the transmission $3 \to 2$ is redundant since node 2 is already reached by the first transmission $1 \to 3$.

Since $E_2/\mathbf{P}_{23} > E_1/\mathbf{P}_{13}$ is true (15), the next edge chosen is $2 \rightarrow 3$, as in the optimal tree T_1.
Case 2 : If T_2 is optimal, we have:

$$\min\left(E_1/\mathbf{P}_{12}, E_2/\mathbf{P}_{23}\right) < E_1/\mathbf{P}_{13} \tag{19}$$
$$\Rightarrow E_2/\mathbf{P}_{23} < E_1/\mathbf{P}_{13} \tag{20}$$

since $E_1/\mathbf{P}_{12} \not< E_1/\mathbf{P}_{13}$ given $\mathbf{P}_{12} < \mathbf{P}_{13}$ (by assumption).

As in Case 1, the first edge chosen is $1 \rightarrow 2$. The decision rules governing which edge is chosen next are the same as in Case 1. Since $E_2/\mathbf{P}_{23} < E_1/\mathbf{P}_{13}$ (20), the next edge chosen in this case is $1 \rightarrow 3$. The only transmitting node in the tree being node 1, the effective broadcast tree generated by the MDLT algorithm is therefore $\{1 \rightarrow 3\}$, as in the optimal tree T_2.

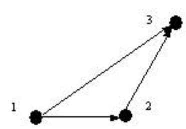

Fig. 2. A 3-node network. Possible broadcast trees are $\{1 \rightarrow 3\}$ and $\{1 \rightarrow 2; 2 \rightarrow 3\}$.

Having proven that the MDLT algorithm generates optimal TTFF trees for 3-node networks, we next prove its optimality for any N-node network. In the proof, we make use of our assertion that "inclusion of an edge at the *kth* iteration can never improve the tree constructed till iteration $k-1$".

Suppose that the tree grown till iteration $k-1$ using the MDLT algorithm, T^{k-1}, is optimal and its TTFF is $\tau^{k-1} = \hat{\tau}^{k-1}$, where $\hat{\tau}^{k-1}$ represents the optimal TTFF. Let (m, n) be the next edge which should be chosen optimally, where $m \in \mathbf{NR}^{k-1}$ and $n \in \mathbf{NNR}^{k-1}$. The TTFF of the resulting optimal tree $(\hat{T}^k)$ is therefore:

$$\hat{\tau}^k = \min_i \left(\vec{E}_i/\vec{Y}_i : i \in \vec{tr}(\hat{T}^k)\right) \tag{21}$$
$$= \min\left(\min_i\left(\vec{E}_i/\vec{Y}_i : i \in \vec{tr}(\hat{T}^{k-1})\right), \vec{E}_m/\mathbf{P}_{mn}\right) \tag{22}$$
$$= \min\left(\hat{\tau}^{k-1},\ \vec{E}_m/\mathbf{P}_{mn}\right) \tag{23}$$

Note that (22) follows from (21) only if our assertion is true. Assuming $\hat{\tau}^k < \hat{\tau}^{k-1}$, eqn. (23) further implies:

$$\min\left(\hat{\tau}^{k-1},\ \vec{E}_m/\mathbf{P}_{mn}\right) > \min\left(\hat{\tau}^{k-1},\ \vec{E}_p/\mathbf{P}_{pq}\right) \tag{24}$$
$$\Rightarrow \hat{\tau}^{k-1} - \vec{E}_m/\mathbf{P}_{mn} < \hat{\tau}^{k-1} - \vec{E}_p/\mathbf{P}_{pq} \tag{25}$$

where $p \in \{\mathbf{NR}^{k-1} \setminus m\}$ and $q \in \mathbf{NNR}^{k-1}$. Since (25) is exactly the MDLT criterion discussed in (9), the algorithm will also choose the optimal edge, (m, n), at the *kth* iteration.

We conclude this section with an argument on why our assertion that "inclusion of an edge at the *kth* iteration can never improve the tree constructed till iteration $k-1$" holds. First, we define and illustrate a procedure for labeling the sequence of transmissions in a connection tree. Given a tree, let us create its sorted version, such that all transmissions from a node are grouped together, in the order they appear in the original tree. For example, if the input tree is:

$$T = \{1 \rightarrow 2;\ 2 \rightarrow 3;\ 1 \rightarrow 4;\ 2 \rightarrow 5;\ 1 \rightarrow 6\}, \tag{26}$$

its sorted version is:

$$T \equiv \mathit{sortedT} = \{\underbrace{1 \rightarrow 2; 1 \rightarrow 4; 1 \rightarrow 6};\ \underbrace{2 \rightarrow 3; 2 \rightarrow 5}\} \tag{27}$$

The sequence of actual transmissions in the tree can be readily identified by reading the sorted tree from left to right and choosing the last transmission in each group (the groups are marked by underbraces in (27)). An equivalent tree to (27) is therefore:

$$T \equiv \mathit{sortedT} \equiv \mathit{sortedT}_{eq} = \{1 \rightarrow 6;\ 2 \rightarrow 5\} \tag{28}$$

Note that the sorting procedure does not alter the TTFF of a tree or affect its validity.

Next, we assign labels to the transmitting nodes in *sortedT* (or $\mathit{sortedT}_{eq}$); the first transmitting node (which is the source) is assigned label 1, the second transmitting node is assigned label 2, etc.[7] All non-transmitting nodes are then assigned a common dummy label, greater than the highest label assigned to any transmitting node. For our example in (27), the labels are shown in Table I. Note that the non-transmitting nodes, {3, 4, 5 and 6}, have been assigned a dummy label, 3, greater than the highest label, 2, assigned to any transmitting node.

TABLE I
Labels assigned to the transmitting nodes in (27).

Node	1	2	3	4	5	6
Label	1	2	**3**	**3**	**3**	**3**

Let us now consider an arbitrary tree generated using the MDLT algorithm, as in (29), where $\{m^k, n^k\}$ represent the transmitting and destination nodes at iteration k. While the set of destination nodes $\{n^1, n^2, \cdots n^k\}$ in (29) are unique, the set of transmitting nodes $\{m^1, m^2, \cdots m^k\}$ need not be so. Let the critical transmission be (m^c, n^c), *i.e.*, $\tau^{c-1} > \tau^c = \tau^{c+1} = \cdots = \tau^{k-1}$. We assume that the *ith* transmitting node is n^c, *i.e.*, $m^i = n^c$, where $i > c$.

$$T^k = \begin{bmatrix} T^{k-1} \\ m^k \rightarrow n^k \end{bmatrix} = \begin{bmatrix} m^1 & \rightarrow & n^1 \\ m^2 & \rightarrow & n^2 \\ & \cdots & \\ m^c & \rightarrow & n^c \\ m^{c+1} & \rightarrow & n^{c+1} \\ & \cdots & \\ m^i = n^c & \rightarrow & n^i \\ m^{i+1} & \rightarrow & n^{i+1} \\ & \cdots & \\ m^{k-1} & \rightarrow & n^{k-1} \\ m^k & \rightarrow & n^k \end{bmatrix} \tag{29}$$

[7]Note that labels can be assigned directly from the input tree, without having to compute its sorted version. We include this additional step purely for the sake of clarity.

For the kth transmission, (m^k, n^k), to be able to improve the tree grown till iteration $k-1$, without affecting its validity, the following conditions must be satisfied.

- **Condition 1**: The transmission $m^k \rightarrow n^k$ must reach node n^c implicitly (*i.e.*, $\mathbf{P}_{m^k n^c} < \mathbf{P}_{m^k n^k}$).
- **Condition 2**: $E_{m^k}/\mathbf{P}_{m^k n^k} > E_{m^c}/\mathbf{P}_{m^c n^c}$
$\Rightarrow\ E_{m^k}/\mathbf{P}_{m^k n^c} > E_{m^k}/\mathbf{P}_{m^k n^k} > E_{m^c}/\mathbf{P}_{m^c n^c}$ (using condition 1 above)
- **Condition 3**: $label(m^k) \nleq label(m^c)$. The condition $label(m^k) < label(m^c)$ implies that node m^k is reached before iteration c (*i.e.*, $m^k \in \{n^1, n^2, \cdots n^{c-1}\}$), since a node can transmit only after it has been reached[8]. If this condition is true, it implies that (m^k, n^c) was a candidate edge for the cth step in (29) and therefore should have been chosen instead of (m^c, n^c) (following condition 2). Since it was not, this condition cannot be true.
The condition $label(m^k) = label(m^c)$ implies that $m^k = m^c$ since no two transmitting nodes can share the same label. Since node n^k was a candidate destination node at iteration c but was not chosen, it must be that $\mathbf{P}_{m^c n^k} > \mathbf{P}_{m^c n^c}$. Obviously, in this case, τ^k must be less than τ^c and therefore no improvement to the tree is possible.
- **Condition 4**: Node m^k must be reached between iterations c and $i-1$, but before it first transmits in the tree T^k.
As argued in Condition 3, if m^k is reached before iteration c, (m^k, n^c) was a candidate edge for the cth step in (29) and therefore should have been chosen instead of (m^c, n^c) (following condition 2). Since it was not, this condition cannot be true.
If m^k is reached at or after iteration i, clearly, the critical transmission (m^c, n^c) cannot be deleted since node m^i $(= n^c)$ would then be transmitting without it being reached first, violating the connectivity of the tree.

Now, let us assume that the transmitting node at the $(c+1)th$ iteration, m^{c+1}, is not the node n^c. Clearly, if that's the case, m^{c+1} must have been reached between steps 1 and $c-1$, since the only new node which has been reached at iteration c is n^c. The edge (m^{c+1}, n^{c+1}) must therefore have been a candidate edge at iteration c and should have been chosen by the MDLT algorithm. Since it was not, we can conclude that the transmitting node at the $(c+1)th$ iteration must be the new node reached during the cth iteration, which is n^c, implying that $i = c+1$ in (29).

Conditions (3) and (4) can therefore be combined as follows:

- Node m^k must be reached at iteration c, before it first transmits in the tree T^k, **and** $label(m^k) > label(m^c)$.

This is possible only if $m^k = n^c = m^{c+1}$, the first equality following from the stipulation that "node m^k must be reached at iteration c" and the latter equality following from the argument in the previous paragraph. However, in this case too, the critical transmission (m^c, n^c) cannot be deleted since node m^k $(= m^{c+1} = n^c)$ would then be transmitting without it being reached first, violating the connectivity of the tree.

Thus, there can be no situation where all four conditions are satisfied and therefore our assertion that "inclusion of an edge at the kth iteration can never improve the tree constructed till iteration $k-1$" holds. Revisiting our example tree (12) in Section V, it is now clear why it cannot be part of an optimal solution. The fifth transmission in the tree must in fact be from node 5, and not node 4.

[8]Note that the converse is not necessarily true. That is, if m^k is reached before iteration c, it does not have to necessarily transmit before m^c. In other words, $label(m^k)$ need not be smaller than $label(m^c)$.

VII. Conclusion

In this paper, we have addressed the problem of maximizing the time-to-first-failure in energy constrained wireless broadcast networks. We discussed a polynomial time algorithm and proved that it can solve the problem optimally, provided the complete power matrix and the battery residual capacities are known.

VIII. Acknowledgment

This work was supported by the Advanced Information Systems Technology (AIST) program at the NASA Office of Earth Sciences.

References

1) Jae-Hwan Chang and Leandros Tassiulas, "Energy Conserving Routing in Wireless Ad-Hoc Networks", Proceedings of INFOCOM, 2000.
2) Robert J. Marks II, Arindam K. Das, Mohamed El-Sharkawi, Payman Arabshahi and Andrew Gray, "Maximizing Lifetime in an Energy Constrained Wireless Sensor Array Using Team Optimization of Cooperating Systems", Proceedings of the World Congress on Computational Intelligence, Honolulu, Hawaii, 2002.
3) Archan Misra and Suman Banerjee, "MRPC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments", Proceedings of WCNC, 2002.
4) Intae Kang and Radha Poovendran, "On the Lifetime Extension of Energy-Efficient Multihop Broadcast Networks", Proceedings of of the World Congress on Computational Intelligence, Honolulu, Hawaii, 2002.
5) C.K. Toh, "Maximum Battery Life Routing to support ubiquitous Mobile Computing in Wireless Ad Hoc Networks", IEEE Communications Magazine, June 2001.
6) C.H. Papadimitriou and K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity", Dover Publications, New York, 1998.
7) D. Bertsekas and R. Gallager, "Data Networks", Englewood Cliffs: Prentice Hall, 1992.