Maximizing Static Network Lifetime of Wireless Broadcast Adhoc Networks

Intae Kang and Radha Poovendran
Department of Electrical Engineering, Box 352500
University of Washington, Seattle, WA. 98195-2500
email: {kangit,radha}@ee.washington.edu

***Abstract*—We investigate the problem of energy-efficient broadcast routing over wireless static adhoc network where host mobility is not involved. We define the lifetime of a network as the duration of time until the first node failure due to battery depletion. We provide a globally optimal solution to the problem of maximizing a static network lifetime through a graph theoretic approach. We also provide extensive comparative simulation studies.**

I. Introduction

One of the important applications of wireless static adhoc networks includes wireless sensor networks. The technology of sensor networks has the potential to change the way we interact with physical environments. For example, inexpensive tiny sensors can be embedded or scattered onto target environments in order to monitor useful information in civil or military situations. Most of these applications make extensive use of a broadcast model of communications for disseminating data to neighboring sensors. In the cases where the communicating sensors are out of reach, they need to rely on intermediate sensor nodes to relay the data back and forth between the sensors.

Wireless networks experience severe attenuation of received signal power due to lossy wireless channel characteristics. Thus, a sender has to ensure that the transmitted signal power level is strong enough for meaningful decoding at receiver end. A salient feature of these networks is the use of microprocessor-embedded, energy-constrained devices. The battery energy of a node is depleted by: (i) computational processing and (ii) transmission/reception of signal to maintain the signal-to-noise ratio above a certain threshold. Although the energy consumption by computations can be further reduced with new developments in low power devices, the energy consumption by communications cannot be overcome. Therefore, it is essential to develop efficient networking algorithms and protocols that are optimized for energy consumption.

In this paper, we investigate the problem of maximizing the network lifetime of a broadcast session over a wireless static adhoc network, where the network is constrained with limited battery energy resource. Specifically, we explore a special case when the network is not self-configurable, but the initial setup of a routing structure is used throughout the session. It might seem limited at this stage, but the usefulness of the obtained solution to this problem and the adaptation to dynamic (self-configuring) networks will be demonstrated and published in the near future.

In broadcast routing, what kind of criteria of optimization will lead to extended lifetime has not been clearly answered up to now. For example, does the minimization of the *total* or *maximum* transmit power at each instance of time extend the lifetime? If so, which one will perform better? Can minimizing the maximum transmit power be optimal in certain circumstances? These are the typical fundamental questions that need to be addressed. To maximize the lifetime of broadcasting, instead of dealing with other (somewhat related) metrics, certain direct measures natural to the analysis of lifetime should be introduced and the characteristics of wireless broadcast channel should be fully utilized.

In unicast routing, energy-efficiency can be roughly achieved by routing a traffic to a path where nodes have sufficient residual battery energy and by avoiding the inclusion of nodes with scarce energy in the path [2], [4]. However, in broadcast routing, every node in the network should be included. Hence, a more important issue is to design algorithms and metrics which make the nodes with scarce energy be assigned with either a very small transmit power or no transmit power (leaf nodes).

The definition of the *network lifetime* as the time of the first node failure [4] is a meaningful measure in the sense that a single node failure can make the network become partitioned and further services be interrupted. We show that the choice of this network lifetime naturally leads to a max-min type (bottleneck) optimization problem, because we want to maximize the first node failure (minimum) time. In the case of broadcast, we model the network as a directed graph and provide a globally optimal solution to the bottleneck optimization problem through graph theoretic approaches.

Other definitions of lifetime used in the literature include: (i) fraction of surviving nodes in a network [8], [10] and (ii) mean expiration time [3]. We show that to extend the network lifetime as defined in [4], building a power-efficient tree such as in [1] is not enough but we also need to consider the cost metric incorporating the residual battery energy level. Therefore, we clearly distinguish between the terms *power-efficiency* (or equivalently short-term energy efficiency e.g., minimum *total* or *maximum* transmit power) and *energy-efficiency* (related to maximizing long-term network lifetime), because, as shown later in

Funded by NSF grant ANI–0093187 and ARO grant DAAD190210242.

this paper, power-efficiency does not necessarily translate to extended lifetime.

The remainder of this paper is organized as follows. In the next section, we give background and define the terms used in this paper. In Section III, we look at the problem of maximizing a static network lifetime without tree updates. Section IV summarizes our simulation results and Section V concludes this paper with our future research.

II. Background and Definitions

In this section, we give background and precisely define the terms used throughout this paper. We assume that each node (host) in a wireless static adhoc network is equipped with an omnidirectional antenna. Because the received power at a node varies as $d^{-\alpha}$ where α is the path loss (attenuation) factor satisfying $(2 \le \alpha \le 4)$, the transmission power required to reach a node at a distance d is proportional to d^{α} assuming that the proportionality constant is 1 for notational simplicity.

Our main concentration in this paper is to investigate topology structures (e.g. trees) in a network layer which enable prolonged network operation. Channel contention and retransmissions in MAC or link layers do affect the energy efficiency but, at this stage, are not covered in this paper. Also, protocol level description is not made, which is our future work.

We denote a network as a directed graph $G = (N, A)$ with a set N of nodes and a set A of directed edges (links). Each node is labeled with a node ID $\in \{1, 2, \ldots, |N|\}$. If it is not specifically stated, we assume that the network is directed.

As opposed to conventional wired networks, there is no permanent connection between the nodes in wireless networks. Whereas, the transmit power level $\{P(i)\}$ assigned to each node i (and node mobility, if it is a mobile adhoc network) determines the network topology.

Definition 1: The *topology control* τ by transmit power $\{P(i)\}$ for $i \in N$ is a mapping $\tau : G \longrightarrow G'$ from a directed graph $G = (N, A)$ to a subgraph $G' = (N^{'}, A') \subset G$ satisfying $N' = N$ and $A' = \{(i,j) \mid (i,j) \in A(G),\ P_{ij} \le P(i)\}$. Given a transmit power distribution $\{P(i)\}$, we will call G' as a *topology induced by* $\{P(i)\}$ [7], [8].

In topology control, the power levels are assigned to satisfy the specified requirements of applications (e.g., connectivity, strong connectivity, and biconnectivity). For example, if a power assignment fails to meet the requirement for the network connectivity, this leads to the network being partitioned. In general, assigning higher power leads to rich interconnections between the nodes providing many redundant paths but at the cost of larger interference to other nodes and faster battery depletion. On the other hand, by assigning the right amount of power level to maintain the network connectivity, we can reduce the battery consumption rate and hence extend the overall network operation time [1], [7], [9].

In a wireless network, there is always a trade-off between reliability (fault tolerance) and network lifetime. There exists a whole spectrum of topology control problems in between. At one extreme, there is flooding (uncontrolled topology) where a network is most reliable at the cost of minimal network lifetime. The other extreme is a spanning tree. Since a spanning tree is the minimal graph structure supporting the network connectivity, it is intuitively clear that a topology which maximizes the network lifetime under transmission should be a tree. But any tree-based scheme comes at the cost of relatively weak link connectivity, because a single node or edge failure results in network partition. A tree-based approach is suitable for the networks where the topology change is not as frequent, such as wireless sensor networks.

The objective in this paper is to construct a broadcast routing tree[1] rooted at the source node with the longest possible lifetime (in the sense of [4]). In broadcast routing, a message originating from the source node should reach every destination node.

Definition 2: We define that a *network is connected* if there exists a directed path denoted as $(S \rightsquigarrow i)$ from the source node S of the broadcast to every destination node i in the network.

The network connectivity is equivalent to the strong connectivity (or reachability) from the root (source) node. The connectivity in this paper does not require duplex links, namely, if there is a directed edge $(u, v) \in (S \rightsquigarrow i)$, it is not necessary for an edge (v, u) to exist. The connectivity of a network can be easily represented with an $|N| \times |N|$ adjacency matrix $\mathcal{A} = [a_{ij}]$, where $a_{ij} = 1$, if there is a link between node i and j, and $a_{ij} = 0$, otherwise. In addition, we assume that the graph is loop-free so that the diagonal entries of $\mathcal{A}$ are zero (i.e., $a_{ii} = 0$ for all $i \in N$). In general, the adjacency matrix is not symmetric for a directed graph (digraph). Without loss of generality, we assume that the source node ID is 1 and, hence, the first row of $\mathcal{A}$ represents the connection from the source node. For a network of size $|N|$, we denote the connectivity matrix [16] as:

$$C = \mathcal{A} + \mathcal{A}^2 + \cdots + \mathcal{A}^{|N|-1}. \tag{1}$$

If the non-diagonal entries of the first row of C are nonzero (i.e., $c_{1j} > 0$ for $j = 2 \ldots |N|$), the network is said to be strongly connected from the source node to every destination node. Practically, the test for connectivity is done using the standard depth-first search (DFS) algorithm [6] which has a running time of $\Theta(|V| + |A|)$.

Note that given a spanning tree T, the required *pairwise transmit power* to maintain a link $(i, j) \in T$ from node i to j is $P_{ij} = d_{ij}^{\alpha}$ where d_{ij} is the distance between the node i and j. If the current battery energy level of node i at time t is $E_i(t)$ and the node i is transmitting to node j, this link can be used for the remaining $E_i(t)/P_{ij}$ units of time.

Definition 3: We define the *link longevity* l_{ij} of a link $(i, j) \in T$ as

$$l_{ij} \equiv \frac{E_i(t)}{P_{ij}}. \tag{2}$$

Definition 4: The *node longevity* ℓ_i of a node $i \in N$ is defined as follows:

$$\ell_i \equiv \min_{j \in \Re_i} \{l_{ij}\} = \frac{E_i(t)}{\max\limits_{j \in \Re_i} \{P_{ij}\}} = \frac{E_i(t)}{P(i)}, \tag{3}$$

[1]Unless otherwise mentioned, a tree in this paper means an arborescence (or branching) [14], [15] from now on.

where $\Re_i = \{k : \pi(k) = i\}$ is the set of neighbors of node i and

$$P(i) = \max_{j \in \Re_i} \{P_{ij}\}, \text{ for } i = 1, \ldots, N \tag{4}$$

is the actual *transmit power* assigned to the node i. Note that $\pi(k)$ denotes the parent node of node k.

Hence, a node i transmitting data with power $P(i)$ can live for ℓ_i units of time. If a node i is a leaf node in the spanning tree, then $P(i) = 0$ and thus $\ell_i = \infty$. Otherwise, the source and relay nodes have a finite node longevity. Note that, given a tree with node i transmitting with power $P(i)$ given by (4), the *total transmit power* is:

$$\mathcal{P}_{total}(T) = \sum_{i \in N} P(i) \tag{5}$$

and the residual battery energy $E_i(t)$ is related to $P(i)$ as follows:

$$E_i(t) = E_i(0) - \int_0^t P(i, \tau)\, d\tau.$$

Definition 5: Given a broadcast routing tree T, (i) the *network lifetime* $\mathcal{L}(T)$ is defined as the duration of the network operation time until the first node failure due to battery depletion at the node, assuming that broadcast from the source node takes place at the beginning of the network initialization. (ii) The *static network lifetime* refers to the lifetime when the tree T does not change once the tree is setup at the initialization phase. (iii) The *dynamic network lifetime* refers to the case when the tree T is updated either periodically or whenever there are changes in the network topology.

Given an initial distribution of current battery energy levels $\{E_i(t)\}$ and the transmit power levels $\{P(i)\}$ determined by a routing algorithm, the *network lifetime* of a connected tree T induced by $\{P(i)\}$ at time t is related to the link and node longevity as follows:

$$\mathcal{L}(T) \equiv \min_{i \in N} \left\{ \frac{E_i(t)}{P(i)} \right\} = \min_{i \in N} \{\ell_i\} \tag{6a}$$

$$= \min_{i \in N} \left\{ \min_{j \in R_i} l_{ij} \right\} = \min_{(i,j) \in A(T)} \{l_{ij}\}, \tag{6b}$$

where $A(T)$ is the edge set induced by a tree T. Hence, the network lifetime of a tree T is determined by a node with the minimum node longevity, which in turn depends upon a link with the minimum link longevity.

Now, we will provide a few examples on how these definitions can be applied to different scenarios.

Example 1: Fig.1 gives a sample instance of a network configuration where node IDs and link longevity values are displayed next to the corresponding nodes and edges. In this example, the network lifetime is determined by the edge $(1, 6)$ because the link longevity (3) of the edge has the minimum value $l_{16} = 1\,\text{sec}$. The node longevity (4) for the source and relay nodes are $\ell_1 = 1\,\text{sec}, \ell_3 = 7\,\text{sec}, \ell_4 = 25\,\text{sec}$ and $\ell_6 = 2\,\text{sec}$. The leaf nodes 2,5,7 and 8 have infinite node longevity. As we can clearly see from this example, the lifetime extension problem is one special case of bottleneck optimization problem [17].

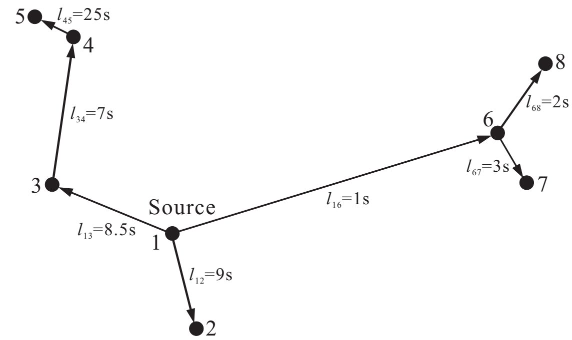

Fig. 1. A sample network configuration: the node with ID=1 is the source node.

Example 2—Equally Distributed Energy Network (EDEN): As a special case, when the nodes of a network have identical initial energy levels (i.e., $E_1(t) = \ldots = E_{|N|}(t) = \mathcal{E}$), we will denote the network as an equally distributed energy network (EDEN). The lifetime of EDEN can be expressed as:

$$\mathcal{L}_{EDEN}(T) = \frac{\mathcal{E}}{\max_{(i,j) \in A(T)} \{P_{ij}\}}. \tag{7}$$

This scenario roughly simulates the real situation where, at the beginning of a conference session, attendees try to bring their laptop fully charged. It is assumed that the battery capacity of each host is the same.

III. Maximizing Static Network Lifetime

In this section, we investigate an optimization problem of finding a routing (spanning) tree which maximizes the network lifetime without tree update. We assume that once a routing tree is established at the beginning of a broadcast session, the same tree is used as a broadcast routing tree for the whole remaining time. We want to find a static routing tree T^* which gives the maximum network lifetime.

Definition 6: The (globally) *optimal static network lifetime* $\mathcal{L}^*$ is defined as

$$\mathcal{L}^* \equiv \max_{T \subset G(N,A)} \{\mathcal{L}(T)\} = \max_{T \subset G(N,A)} \min_{(i,j) \in A(T)} \{l_{ij}\} \tag{8}$$

where l_{ij} is the link longevity of an edge (i, j).

This max-min (bottleneck) optimization problem (8) is a kind of *network design* problem which finds an optimal power assignment to each node by finding an optimal spanning tree. We will show that we can, in fact, find a (global) optimum solution to this problem by polynomial-time bounded greedy algorithms.

A. A special case (EDEN)—undirected graph

We initially look at a special case (Example 2) when all the nodes have identical battery energy levels. Although this constraint is possibly too stringent in real situations, we solve this problem first because this provides insights into the more general case of unequal battery energies. Due to this assumption, the graph can be considered as an undirected graph because $l_{ij} = \mathcal{E}/P_{ij} = \mathcal{E}/P_{ji} = l_{ji}$.

First, we will derive a lemma which will be used later.

Lemma 1—MST minimizes the maximum weight: Let's denote the maximum edge weight of a spanning tree $T \subset G(N, A)$ as $\mu(T) \equiv \max_{(i,j)\in A(T)} w(i,j)$ and we will denote the edge (i,j) satisfying this condition as the *bottleneck edge*. Let μ^* denote the minimum of $\mu(T)$ over all possible spanning trees of a weighted undirected graph, i.e.,

$$\mu^* \equiv \min_{T\subset G(N,A)} \{\mu(T)\}. \tag{9}$$

If we denote the Minimum Spanning Tree (MST) as T_{MST}, then

$$\mu(T_{MST}) = \mu^*. \tag{10}$$

Proof: A *Bottleneck Spanning Tree* (BST) denoted as T_{BST} is defined as a spanning tree which has the minimum bottleneck edge weight:

$$T_{BST} \equiv \arg\min_{T\subset G(N,A)} \mu(T). \tag{11}$$

This problem can be rephrased as proving that a minimum spanning tree is also a bottleneck spanning tree. The following three cases are considered in our proof. First, the case $\mu(T_{MST}) < \mu(T_{BST})$ is not possible by the definition (11), since $\mu(T_{BST}) = \mu^*$. Now, suppose that the minimum spanning tree is not a bottleneck spanning tree, then

$$\mu(T_{MST}) > \mu(T_{BST}) = \mu^*. \tag{12}$$

Now, denote the bottleneck edge of T_{MST} as $e^* = \arg\max_{(i,j)\in T_{MST}} w(i,j)$. Removing this edge $e^* \in T_{MST}$ introduces a *cut* $\mathcal{C}$, which is by definition a partition of a node set N into two nonempty subsets N_1 and N_2 or, equivalently, a set of edges connecting N_1 and N_2, i.e., $\mathcal{C} = \{(i,j) | i \in N_1, j \in N_2\}$. Note that since T_{BST} is a tree, exactly one edge $e^\circ \in T_{BST}$ should be included in the cut $\mathcal{C}$ so that connectivity of a tree can be satisfied (e° is not necessarily the bottleneck edge of T_{BST}). By the cut optimality of a minimum spanning tree [6], we know that $w(e^*) \le w(e^\circ)$. Therefore

$$\mu(T_{MST}) = w(e^*) \le w(e^\circ) \le \mu(T_{BST}). \tag{13}$$

This leads to a contradiction with (12). Therefore, the only possible case is $\mu(T_{MST}) = \mu(T_{BST})$. Thus, every minimum spanning tree is also a bottleneck spanning tree. ■

Lemma 1 proves that a minimum weight spanning tree is a sufficient condition to be a bottleneck spanning tree. However, it is not a necessary condition in general, which can be simply proved with a counter example [6]. Hence, MST is a globally optimal solution but *not* unique. In fact, a class of trees (bottleneck spanning trees) with the same maximum weight edge serve our purpose equally well.

Theorem 1: If all the nodes in a network have identical battery energy $\mathcal{E}$, then the minimum spanning tree is a globally optimal solution to the static network lifetime maximization problem.

Proof: If we use the edge weight $w(i,j) = P_{ij} = d_{ij}^\alpha$, then $w(i,j) = w(j,i)$ because $d_{ij} = d_{ji}$ and the network is modeled as an undirected graph. Hence, Lemma 1 is applicable. Using (7), (8), and (9), the (global) maximum lifetime $\mathcal{L}^*$ is

$$\begin{aligned}
\mathcal{L}^* &= \max_{T\subset G(N,A)} \mathcal{L}_{EDEN}(T) \\
&= \mathcal{E} \cdot \max_{T\subset G(N,A)} \left\{ \frac{1}{\max_{(i,j)\in A(T)} P_{ij}} \right\} \\
&= \frac{\mathcal{E}}{\min_{T\subset G(N,A)} \max_{(i,j)\in A(T)} P_{ij}} \\
&= \frac{\mathcal{E}}{\mu(T_{MST})}.
\end{aligned}$$

Since $\mu(T_{MST}) = \mu^*$ is minimum among all possible spanning trees by Lemma 1, the maximum lifetime $\mathcal{L}^*$ is obtained by using a minimum spanning tree with cost metric $w(i,j) = P_{ij}$ and is also globally optimal. ■

Corollary 1: Minimizing the total pairwise transmit power gives the optimal solution to the static lifetime maximization problem only when the network is equally energy distributed (EDEN).

Proof: By definition [6], MST is a spanning tree with minimum total cost:

$$T_{MST} = \arg\min_{T\subset G(N,A)} \sum_{(i,j)\in A(T)} w(i,j). \tag{14}$$

If we use $w(i,j) = P_{ij}$, a pairwise transmit power, it is obvious by Theorem 1. ■

We strongly emphasize that MST is *not* a tree with minimum total transmit power (5) and does not exploit the wireless broadcast advantage property during its construction as in [1]. In general, when there exist links with equal weights (costs), minimum spanning tree is not unique. However, it is a well-known property that when the weights are distinct the minimum spanning tree is unique [6]. This uniqueness property facilitates the implementation of a distributed algorithm for MST [5].

B. General case—directed graph

Once we begin to consider the general distribution of the residual battery energy levels (i.e. there exist i and j such that $E_i(t) \ne E_j(t)$), the graph is no longer undirected. Note that $w(i,j) = E_i(t)/P_{ij}$ and $w(j,i) = E_j(t)/P_{ji}$. Hence, although $P_{ij} = P_{ji}$, $w(i,j) \ne w(j,i)$.

Finding a minimum weight arborescence (branching) [14], [15] has exactly the same formula as undirected one (14), except that the underlying graph is changed to a directed one [14], [15]. Therefore, a minimum weight arborescence is conceptually a direct extension of MST for a directed case with the same underlying optimization principle.

In Section III-A, we found that a global optimal solution for EDEN is MST and then obtained an important result that *minimization of the total cost leads to minimization of the maximum cost for an undirected graph (but not vice-versa).* It is natural to

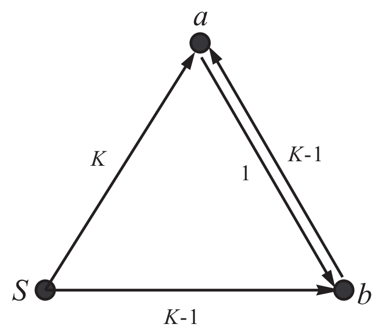

Fig. 2. Illustration for Lemma 2.

ask the question whether this analogy carries over to a directed graph as well. Unfortunately, the answer is negative and a proof is provided, which is due to Khuller [12].

Lemma 2: Minimum (total) weight arborescence rooted at the source node does not necessarily minimize the maximum weight of a directed graph [12].

Proof: This can be easily verified with a simple counter example. Consider the configuration in Fig.2 with a node set $N = \{S, a, b\}$. The edge weights are shown in Fig.2 next to the directed edges, where $K > 3$ is assumed. The minimum weight branching rooted at S is $\{(S,a),(a,b)\}$ of total weight $K+1$ and maximum weight K. The branching minimizing the maximum weight of any edge is $\{(S,b),(b,a)\}$ with total weight $2(K-1)$ and with maximum weight $K-1$. Although the total weight is smaller $(K+1 < 2(K-1))$, the maximum weight is larger $(K > K-1)$. Therefore, the minimum weight arborescence does not minimize the maximum weight of a directed graph. ■

Thus, the results for undirected graph in Lemma 1 cannot be directly applied to a directed graph and we need to solve a new bottleneck optimization problem similar to Lemma 1 for a directed graph. We will show that there is a polynomial-time bounded algorithm that will compute what we need.

Algorithm 1: Consider $\mathcal{S}$ the sorted set of weights on edges in an increasing order. Let τ_i be the topology (subgraph) formed by the first i edges in $\mathcal{S}$. Check if τ_i has the property that the root can reach every node. If so, we can discard all the edges with index larger than i. We are looking for the smallest i for which this property holds. We can do binary search to find this i, since there is some value j such that τ_j does not have this property and τ_{j+1} does [12].

By inspection, it is clear that this algorithm will give us a desired optimal solution (i.e., the bottleneck edge weight of a directed graph), because this algorithm finds the first edge which makes the graph connected. The computational complexity is polynomial-time bounded, since the binary search will only add a worst cast running time $\Theta(\log|A|)$ factor to a linear time $\Theta(|V|+|A|)$ connectivity test using the standard depth first search (DFS) algorithm. Therefore, the running time of Algorithm 1 is $\Theta((|V|+|A|)\log|A|)$.

Algorithm 2—DMST: By applying Algorithm 1, a topology τ_j can be found where all the edge weights are bounded by the bottleneck edge weight. We applied Prim's algorithm [6] on τ_j rooted at the source node by inspecting only outgoing edges at each step. We will call this algorithm as a *directed minimum spanning tree* (DMST).

The difference between Algorithm 1 and 2 (DMST) is that Algorithm 1 finds a connected topology, whereas DMST finds a connected arborescence. It is intuitively clear that DMST has the same bottleneck edge as Algorithm 1, since otherwise the arborescence cannot be connected. Therefore, DMST finds the minimum of the maximum edge weight out of all possible arborescences.

Theorem 2: Let $w(i,j)$ be the inverse of link longevity (or normalized pairwise transmit power):

$$w(i,j) = l_{ij}^{-1} = \frac{P_{ij}}{E_i(t)}, \tag{15}$$

then DMST is a (globally) optimal broadcast routing tree solution of static network without tree update.

Proof: Applying Algorithm 1 and 2 on a directed graph leads to an arborescence that minimizes the maximum edge weight among all possible arborescences. From (15), DMST has the minimum of the maximum $P_{ij}/E_i(t)$. Equivalently, it has the maximum of the minimum link longevity l_{ij}. Since the globally optimal network lifetime is (8), this proves that DMST can achieve global optimality in maximizing the general static network lifetime. ■

IV. Simulation Model and Results

In this section, we will compare the lifetime performance of three different metrics and corresponding algorithms:

- *maximize the static network lifetime* (MSNL)—DMST: DMST algorithm is explained in the previous section and we will call the metric as MSNL.
- *minimize the maximum transmit power* (MMTP)—MST: In section III-A, we proved that MST corresponds to MMTP.
- *minimize the total transmit power* (MTTP)—BIP: Here, we introduce another metric minimizing the total transmit power (5). It was shown in [11] that finding a tree with a minimum total transmit power with limited node power is NP-hard. However, there is a suboptimal centralized algorithm called Broadcast Incremental Power (BIP) algorithm [1], which is currently the best known algorithm for that purpose.

Simulations are performed according to the following simplified network model. Within a 10×10 m^2 square grid region, network configurations are randomly generated according to uniform distribution, and broadcast trees rooted at the source node are constructed based on the cost metrics. Path loss exponent $\alpha = 2$ is considered in the simulation. To isolate the effect of each metric, all the generated nodes are assumed to be in the multicast group (broadcasting). The initial battery energy distribution $\{E_i(0)\}$ is drawn according to specified probability distributions (e.g., uniform distribution unif(η,ξ) ranging from the minimum value η to the maximum value ξ). A broadcast

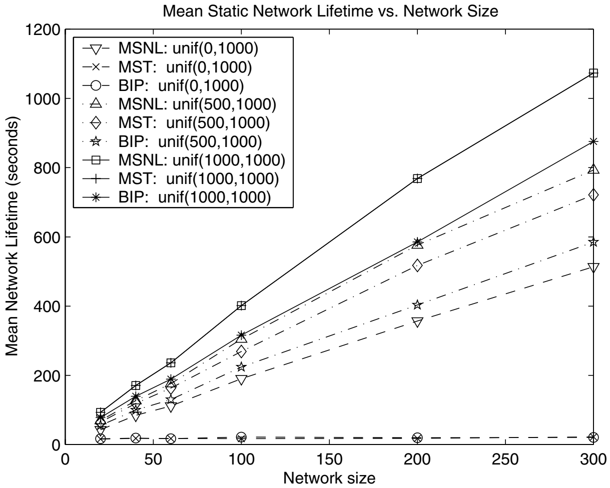

Fig. 3. Static Network Lifetime for MSNL, MST and BIP ($\alpha = 2$)

session initiates at time $t = 0$ and is assumed to carry a constant bit rate (CBR) traffic. The simulation results are for the static (non-mobile) network topology (e.g. wireless sensor networks) and the maximum transmission range is assumed to be $10\sqrt{2}$m. We consider only energy consumption by transmit power but our model can be easily adapted to include signal receive power.

Fig.3 summarizes the lifetime performance of static trees[2] (i.e., MSNL, MST and BIP) for various distributions of the initial battery energy $\{E_i(0)\}$ and the size of the networks $|N|$. Each point in the figure represents an average network lifetime of 100 different network topologies ($\alpha = 2$). The same random seeds are used for a valid comparison of each metric. The initial energy $\{E_i(0)\}$ is distributed according to three uniform probability distributions: (i) unif$(1000, 1000)$=constant, (ii) unif$(500, 1000)$, and (iii) unif$(0, 1000)$.

In general, the static network lifetime increases linearly as a function of the network size per 10×10 m^2 region. If a network has a larger total inital energy $\sum_{i \in N} E_i(0)$, then the lifetime also increases. Notice that, when all the nodes have identical energy of 1000 units (EDEN), the lifetime by MST perfectly overlaps with MSNL, which is consistent with the theoretical result given in Section III-A. In other distributions, unif(0,1000) and unif(500,1000), MST is no longer optimal and MSNL always produces longer lifetime. The separation of MSNL from other metrics, MST or BIP, becomes even more significant when $\{E_i(0)\}$ is uniformly distributed from 0 to 1000 energy units. This is because the max-min lifetime is heavily dependent on the nodes with very scarce initial energy. Also note that MST performs better than BIP, on average. This is because BIP algorithm is optimized for a global property (MTTP), whereas the MST algorithm is optimized for a local property (MMTP). In summary, our simulation results support that, when maximization of network lifetime as in [4] is considered, MSNL always performs better than MST and BIP, because the residual battery energy is considered in its cost metric.

[2]From now on, we will use the terms for the optimization metrics and the corresponding algorithms interchangeably.

V. CONCLUSIONS

We investigated the problem of maximization of network lifetime (in the sense of [4]) in energy-efficient broadcast routing over wireless adhoc networks where the host mobility is not involved and the network is not self-configurable. The lifetime of a network is defined as the duration of time until the first node failure due to battery exhaustion. We proved that the optimal static broadcast routing tree is given by MSNL with edge cost as an inverse of link longevity (or normalized pairwise transmit power). Minimizing the maximum transmit power can be (temporarily) optimal only when the network has equally distributed energy, which is not likely in the real situations.

The performance of energy-aware (lifetime-aware) algorithm MSNL was compared with other power-aware algorithms (MST and BIP) and we showed that MSNL significantly outperforms them, which is due to the inclusion of the residual battery energy in its cost metric.

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. IEEE INFOCOM 2000, pp. 586–594.
[2] S. Singh, M. Woo, and C.S. Raghavendra, "Power-aware routing in mobile ad hoc networks," Proc. ACM/IEEE MOBICOM, pp. 181-190, Dallas, TX, Oct. 1998
[3] C.-K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," IEEE Communications Magazine, Jun. 2001, pp. 138-147.
[4] J.H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," INFOCOM '2000, March 2000.
[5] R.G. Gallager, P.A. Humblet, and P.M. Spira, "A distributed algorithm for minimum-weight spanning trees," ACM Trans. on Programming Languages and Systems, vol.5, no. 1, Jan. 1983, pp. 66-77.
[6] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*, 2nd ed., MIT press, Cambridge, MA, 1990.
[7] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in Proc. IEEE INFOCOM 2000, March 2000.
[8] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," Proc. IEEE INFOCOM 2001
[9] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless ad-hoc network," Proc. ACM/IEEE MOBICOM, 2001, pages 97-107, Rome, Italy, July, 2001.
[10] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," ACM SIGMOBILE 7/01 Rome, Italy.
[11] O. Egecioglu and T. Gonzalez, "Minimum-energy broadcast in simple graphs with limited node power," Proc. IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2001), Anaheim, CA, August 2001, pp. 334-338
[12] S. Khuller, Private communication.
[13] P.A. Humblet, "A distributed algorithm for minimum weight directed spanning trees," IEEE Trans. on Communications, vol. COM-31, No. 6, June 1983.
[14] Y.J. Chu and T.H. Liu, "On the shortest arborescence of a directed graph," Sci. Sinica, vol. 14, pp. 1396-1400, 1965.
[15] R.E. Tarjan, "Finding optimum branchings," Networks, vol.7, 1977, pp. 25-35.
[16] J.M. Aldous, R.J. Wilson, *Graphs and Applications: An Introductory Approach*, Springer-Verlag, 2000.
[17] R. Tyrrell Rockafellar, *Network Flows and Monotropic Optimization*, Athena Scientific, July 1998.