Mobile Networks and Applications 10, 879–896, 2005

DOI: 10.1007/s11036-005-4445-5

Maximizing Network Lifetime of Broadcasting Over Wireless Stationary Ad Hoc Networks*

INTAE KANG and RADHA POOVENDRAN
Department of Electrical Engineering, University of Washington, Seattle, WA

Published online: 24 October 2005

Abstract. We investigate the problem of extending the network lifetime of a single broadcast session over wireless stationary ad hoc networks where the hosts are not mobile. We define the network lifetime as the time from network initialization to the first node failure due to battery depletion. We provide through graph theoretic approaches a polynomial-time globally optimal solution, a variant of the minimum spanning tree (MST), to the problem of maximizing the static network lifetime. We make use of this solution to develop a periodic tree update strategy for effective load balancing and show that a significant gain in network lifetime over the optimal static network lifetime can be achieved. We provide extensive comparative simulation studies on parameters such as update interval and control overhead and investigate their impact on the network lifetime. The simulation results are also compared with an upper bound to the network lifetime.

Keywords: broadcast, energy efficient, routing protocols, ad hoc network, network topology, graphs and networks, algorithm/protocol design and analysis

1. Introduction

One of the important applications of wireless stationary ad hoc networks is wireless sensor networks [2,4,6,13]. The constraint of limited battery energy is one of the most salient feature of sensor networks [6,8,13,24]. Therefore, it is essential to develop efficient networking algorithms and protocols that are optimized for energy consumption. In this paper, we consider the wireless stationary ad hoc network strictly constrained by limited battery energy resource. We investigate the network lifetime maximization problem of a session-oriented single broadcast traffic. Note that session-oriented broadcast is markedly different from "regular" broadcast which is used to maintain network state information by flooding control packets where the reliability is of utmost importance. However, the session-oriented broadcast is a more suitable model, for instance, for continuous multimedia traffic where energy-efficiency is more valued than reliability. In the context of sensor networks, energy efficient broadcast tree construction is important for distribution of updates to sensor nodes, reliable sensor data delivery, among others. The results presented in this paper hold for arbitrary wireless stationary ad hoc networks and hopefully will provide valuable insights into designing link cost metrics to prolong network lifetime.

A similar network lifetime maximization problem in unicast routing has been pursued in Chang et al. [10]. However, broadcasting in wireless environment is fundamentally different from unicasting in many aspects:

(1) Due to the broadcast nature of the wireless medium using omnidirectional antennas, messages sent to a receiver reach every node within the transmission range for "free," and this property is called "*wireless broadcast advantage*" [5], which adds significant complexity to broadcast problems in wireless environments.
(2) For unicast routing, energy-efficiency can be roughly achieved by routing traffic through a path where nodes have sufficient residual battery energy and by avoiding the inclusion of nodes with scarce energy in the path [7,10,19]. However, broadcasting requires that every node in the network be involved either as a receiver or as a relay node. Hence, it is important to design algorithms and link cost metrics so that they can adaptively assign either very small or no transmit power to the nodes with scarce battery energies.
(3) Although elegant formulations in unicasting [10] are possible using the network flow theory [41], this is mainly due to the fact that the flow conservation property is satisfied in unicasting. To solve broadcast problems, we can not apply the network flow theory without modification, since the flow conservation property simply does not hold in broadcasting. Instead we take graph theoretic and heuristic approaches to solve the broadcast problem.

Conceptually similar work in other fields includes the maximization of battery lifetime of multi-celled batteries. In [31,32], the authors seek maximization of battery usage by comparing optimization criteria such as the minimization

*A preliminary version of this paper appeared in IEEE ICC 2003 [35]. This research was funded in part by NSF grant ANI-0093187, ONR award #: N00014-04-1-0479 and Collaborative Technology Alliance (CTA) from ARL under DAAD19-01-2-0011. All statements and opinions are that of the authors and do not represent any position of the U.S government.

of the *total* (MINTOTAL) cost or *maximum* (MINMAX) cost at each instance of time. Although the underlying assumptions are different from ours, they demonstrated that a MINMAX strategy can be a more effective load-balancing technique and achieve better performance than MINTOTAL strategy.

In this paper, we first explore the static (or fixed) network case such that the network is not self-configurable, but the initial setup of a routing structure is used throughout the session. We pursue the similar questions as those raised in [31] in a broadcast routing setup [5,6,8,22] and try to answer which optimization criteria such as MINTOTAL transmit power or MINMAX transmit power lead to longer extended network lifetime. We observe that current research in energy-efficient broadcast routing is heavily biased to MINTOTAL criteria. Since the seminal work of Wieselthier et al. [5], there has been much progress in finding solutions for MINTOTAL transmit power. Theoretically, it has been proven that the construction of a broadcast routing tree with MINTOTAL transmit power is NP-complete [21,25,26,34]. On the other hand, several suboptimal, yet efficient, heuristic algorithms such as *Broadcast Incremental Power* (BIP) [5], *Embedded Wireless Multicast Advantage* (EWMA) [34] and *Largest Expanding Sweep Search* (LESS) [36][1] have been proposed that use greedy approaches to construct a power-efficient tree. A characteristic common to all of these algorithms is that they effectively utilize the wireless broadcast advantage property. However, it is unclear whether MINTOTAL strategy can provide the longest network lifetime. In this paper, we investigate energy-efficient broadcast routing mainly from the viewpoint of MINMAX optimization criteria to maximize the network lifetime. An optimal solution for the static network case is provided using a graph theoretic approach.

Subsequently, the extension of the solution obtained for static networks to the dynamic (self-configuring) networks is also presented, where the routing structure is allowed to change adaptively. Once again, our work focuses on evaluating which optimization criteria to choose at each update interval and how to design link cost metric to get better network lifetime performance. We consider various cost and optimization metrics that extend the network lifetime and provide comprehensive simulation results. Our work is distinguishable from other work [12,37] in that we present detailed analyses of the impact on network lifetime of other crucial parameters such as tree update interval, network density, initial energy distribution and control overhead, which have not been addressed before. One notable finding is that MINMAX cost in conjunction with control overhead can provide longer lifetime, and constitute a viable (if not better) approach to MINTOTAL strategy. There exists another very interesting approach called the *cell-based energy conservation techniques* proposed by Blough et al. [38]. While theoretically interesting, it assumes an underlying "perfect load-balancer" and hence is limited in practical usages. In contrast, our schemes try to reach the upper bound on network lifetime with a practical on-line tree update strategy.

The remainder of this paper is organized as follows: In the next section, we provide background and define the terms used in this paper. In Section 3, we define static network lifetime and the maximization problem of static network lifetime is presented in Section 4. In Section 5, heuristics to extend dynamic network lifetime with periodic tree update strategies are provided. Sections 6 and 7 summarize our simulation results for static and dynamic network, respectively. Section 8 presents our conclusions.

2. Preliminaries

In this section, we present the background information and define terms that are used throughout this paper. Notations and some of the definitions are summarized in Table 1.[2]

2.1. Network model

We assume that each node (host) in a wireless ad hoc network is equipped with an omnidirectional antenna. We assume each node acquires its location information either using GPS or other localization techniques [42]. The broadcast routing trees rooted at the source node are constructed using algorithms according to the different cost and optimization metrics that will be discussed in the subsequent sections. We assume that a broadcast session initiates at time $t = 0$ and carries a constant bit rate (CBR) traffic.

When the Euclidean distance between node i and j is d_{ij}, the received power at a node varies as $d_{ij}^{-\alpha}$ where α is the path loss (attenuation) factor that usually satisfies $2 \leq \alpha \leq 4$. The required *pairwise transmit power* $\mathbf{P}_{ij}$ to maintain a link (i, j) from node i to j is $\mathbf{P}_{ij} = d_{ij}^{\alpha}$ assuming the proportionality constant related to the receiver sensitivity threshold is set to 1 (0 dB) for notational simplicity. Clearly, the matrix $[\mathbf{P}_{ij}]$ is a constant matrix that is invariant over time if nodes' location do not change.

We represent the amount of power consumption at node i at time t as $P_i(t)$, and its corresponding transmission range as $R_i(t)$. To reach node j from node i, we assume the required RF transmit power of node i is $P_i^{RF}(t) = \mathbf{P}_{ij}$. Other signal processing powers for transmission, reception, and other computational processing denoted p_i^T, p_i^R and p_i^C, respectively, contribute to the battery energy drain [11,12]. (Refer Table 1 for details.) Assuming identical transceivers are used, we set $p_i^T = p^T$, $p_i^R = p^R$ and $p_i^C = p^C$ for all i. Then, the general

[1]To the best knowledge of authors, our newly developed LESS algorithm constitutes currently the best performing heuristic algorithm in terms of total transmit power cost.

[2]We note that all notations defined in Table 1 (except $\mathbf{P}_{ij}$, $\mathcal{L}^*$, and $\mathcal{L}^\circ$) are time-dependent variables in general. However, whenever the clarity of presentation is favored, the time dependence in the expression will be omitted.

Table 1
Notations and definitions.

(i, j)	a directed edge (or link) in a directed graph G
$\mathbf{W}_{ij}$	the weight of an edge (i, j) for a weighted directed graph G
T	a directed spanning tree (arborescence) rooted at source node S
d_{ij}	Euclidean distance between node i and j
α	path loss factor satisfying $2 \le \alpha \le 4$
$\mathbf{P}_{ij}$	pairwise transmit power, $\mathbf{P}_{ij} = d_{ij}^{\alpha}$
$P_i^{RF}(t)$	RF transmit power of node i at time t due to power amplifier circuitry (PLL, VCO, etc.) within an antenna
$R_i(t)$	transmission range of node i at time t
p_i^T	transmit signal processing power of node i due to modulation, encoding and encryption, etc.
p_i^R	receive signal processing power of node i due to equalization, demodulation, decoding and decryption, etc.
p_i^C	other information processing power of node i
$\mathcal{P}_{TX}(T)$	total transmit power of nodes to maintain a spaning tree T
$\aleph_i$	physical neighbor of node i, i.e., $$\aleph_i := \{k \mid 0 < d_{ik} \le R_i\} \quad (1)$$
$\Re_i$	logical neighbor of node i, i.e., adjacent nodes of node i in a spanning tree T such that $$\Re_i := \{k \mid (i,k) \in T\} \quad (2)$$
$\mu(T)$	maximum edge weight of a spanning tree T such that $$\mu(T) := \max_{(i,j)\in T} \{\mathbf{W}_{ij}\} \quad (3)$$
$E_i(t)$	residual battery energy level of node i at time t
$\mathbf{L}_{ij}(t)$	residual link longevity of link (i, j) at time t such that $$\mathbf{L}_{ij}(t) := E_i(t)/\mathbf{P}_{ij} \quad (4)$$
$\ell_i(t)$	residual node longevity of a node i at time t such that $$\ell_i(t) := \frac{E_i(t)}{P_i(t)} = \frac{E_i(t)}{\max_{j\in\Re_i}\{\mathbf{P}_{ij}\}} = \min_{j\in\Re_i}\{\mathbf{L}_{ij}(t)\} \quad (5)$$
$\mathcal{L}(T)$	static network lifetime (SNL) of a broadcast network given by a spanning tree T
$\mathcal{L}^*$	optimal (maximum) static network lifetime over all possible spanning tree T such that $$\mathcal{L}^* := \max\{\mathcal{L}(T) \mid T \subset G\} \quad (6)$$
$\mathcal{L}^\circ$	optimal (maximum) dynamic network lifetime
Δt	update interval in dynamic network

form of $P_i(t)$ is

$$P_i(t) = P_i^{RF}(t) + p^T I\left\{P_i^{RF}(t) > 0\right\} + \sum_{j\in N\setminus\{i\}} p^R I\{d_{ij} \le R_j(t)\} + p^C \quad (7)$$

where

$$I\left\{P_i^{RF}(t) > 0\right\} = \begin{cases} 1, & \text{if } i \text{ is transmitting at time } t \\ 0, & \text{otherwise} \end{cases}$$

$$I\{d_{ij} \le R_j(t)\} = \begin{cases} 1, & \text{if } d_{ij} \le R_j(t) \text{ at time } t \\ 0, & \text{otherwise.} \end{cases}$$

Note that $I\{P_i^{RF}(t) > 0\}$ is an indicator function which is 1 only when $P_i^{RF}(t) > 0$, that is, transmission with nonzero RF power always incurs transmit signal processing power p^T. Similarly, $I\{d_{ij} \le R_j(t)\}$ means that receive signal processing power p^R is required for node i, if it is within the transmission range of node j.

A network is represented as a weighted directed graph $G = (N, A)$ with a set N of $n = |N|$ nodes and a set A of $m = |A|$ directed edges (links). An edge $(i, j) \in N^2$ exists if $d_{ij} \le R_{\max}$. This model is usually called the (unit) disk graph [25]. The weight of the edge (i, j) is denoted by $\mathbf{W}_{ij}$. We define a network is *connected*, if there exists a directed path from the source node S to every node $i \in N$. Given transmission ranges $\{R_i(t)\}_{i\in N}$, the *topology* τ *induced by* $\{R_i(t)\}$ 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 $A' = \{(i, j) \mid (i, j) \in A(G), d_{ij} \le R_i(t)\}$ [16–18].

We define a *static network* as a network in which the underlying routing structure is not self-reconfigurable or does not change over time. When a wireless network is self-reconfigurable or changes over time, we will call it a *dynamic network*. If the positions of hosts are allowed to change, we call it a *mobile network*; otherwise, we call it a *stationary network* to avoid confusion with the static network.

For a directed graph, a directed spanning tree rooted at a node is called an *arborescence* [28,29]. 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 should be a tree. Given a spanning tree T, the actual (node) *transmit power* assigned to the node i is $P_i = \max_{j\in\Re_i}\{\mathbf{P}_{ij}\}$ where $\Re_i$ denotes the *logical neighbor* of node i which is a set of adjacent (child) nodes in the directed tree T such that $\Re_i = \{k \mid (i, k) \in T\}$. In contrast, if node i is transmitting with power P_i, then the *physical neighbor* $\aleph_i$ of node i in a wireless network is a set of all the nodes within the communication boundary such that $\aleph_i = \{k \mid 0 < d_{ik} \le R_i\}$. The set of nodes within the maximum communication range $R_{\max}$ of node i is called the (1-hop) *neighbor* of node i, i.e., $\mathcal{N}_i = \{k \mid 0 < d_{ik} \le R_{\max}\}$. In general, the physical neighbor is determined by the network topology and transmit power, whereas the logical neighbor is determined by routing algorithms. Hence, they do not generally coincide and usually $\Re_i \subseteq \aleph_i$. Clearly, the *total transmit power* $\mathcal{P}_{TX}(T)$ corresponding to a spanning tree T is the sum of all node transmit power $\mathcal{P}_{TX}(T) = \sum_{i\in N} P_i$.

In a wireless network, depending on the transmit power assignment, 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 with maximum transmit power (uncontrolled topology) where a network is most reliable at the cost of minimal network lifetime [23].

When trying to extend the network lifetime, it is critical to define what we mean by the lifetime of a network. In this paper, we adopt the definition of the *network lifetime* as the time to the first node failure due to battery depletion at the node as defined in [10]. We assume that broadcast from the source node takes place at the beginning of the network initialization. The *static network lifetime* $\mathcal{L}(T)$ corresponding to a tree T refers to the lifetime, when the tree T does not change once the tree is setup at the initialization phase. The *dynamic network lifetime* refers to the case when the trees are updated based on an update policy (for example, either periodically or whenever there are changes in the network topology). We note that other definitions of network lifetime used in the literature include (i) fraction of surviving nodes in a network [6,12,17,20] and (ii) mean expiration time [9], but these definitions are not used in this paper.

2.2. Energy dissipation model

We will use a linear battery discharge model. We do not consider in our battery model the nonlinear behavior of voltage as a function of remaining capacity [7] or the battery charge recovery effect due to diffusion process [31,32]. We intend to study the effect of different battery discharge models in the future. Let $E_i(0)$ denote the initial battery energy level of node i at time $t = 0$. Considering all the power consumption components introduced in (7), the *residual battery energy* $E_i(t)$ at time t satisfies

$$E_i(t) = E_i(0) - \int_0^t P_i(\tau)d\tau \tag{8}$$

$$= E_i(0) - \int_0^t P_i^{RF}(\tau)d\tau - p^T \int_0^t I\left\{P_i^{RF}(\tau) > 0\right\} d\tau - p^R \sum_{j\in N\setminus\{i\}} \int_0^t I\{d_{ij} \le R_j(\tau)\}d\tau - p^C t \tag{9}$$

where $\int_0^t I\{P_i^{RF}(\tau) > 0\}d\tau$ corresponds to the length of "on" period of node i up to time t and $\int_0^t I\{d_{ij} \le R_j(\tau)\}d\tau$ is the length of the period when node i was within the transmission range of node j.

To avoid undue complication, we set $p^T = p^R = 0$ and investigate the energy consumption by only RF transmit power. However, we do consider p^C by accounting for the control overhead to setup a routing tree in later sections. Hence, we will use a simplified version of (8) such that

$$E_i(t) = E_i(0) - \int_0^t P_i^{RF}(\tau)d\tau - p^C t. \tag{10}$$

The sum of all node energies $\sum_{i\in N} E_i(t)$ at a given time t is referred to as the *energy pool* of the network.

2.3. Single vs. multiple broadcast session

In this paper, we will concentrate on the lifetime maximization problem of a single broadcast session, not multiple sessions. That is because considering multiple broadcast session requires scheduling; otherwise, none of the sessions can succeed assuming all broadcasts are performed in the same channel. We will shortly show that even very naive scheduling can yield the desired load-balancing effect. It makes hard to appreciate the true effect of link cost metric design and the choice of optimization criteria, which are the main focus of this paper. Considering only a single broadcast session makes it possible to decouple each effect from the scheduling problem. All the definitions introduced in later sections are tailored to a single broadcast session. Let us explain this point with a simple example. Let us consider a linear topology where every node lies on a straight line segment as in figure 1, where all three subfigures have the same node distribution. Each figure corresponds to an independent broadcast session carrying different traffic.

We assume the distances between adjacent nodes are one unit and every node has the same initial battery energy $E_i(0) = \mathcal{E}$. We consider the nodes do not move around (stationary network) and routing trees do not change over time (static network). In figure 1(a) and (b), node A is the source of broadcast and in figure 1(c), node B is the source. As will be presented later in Corollary 1, the minimum spanning tree (MST) solution shown in figure 1(a) is the optimal solution for maximum static network lifetime in case of equally distributed energy network (EDEN). Since the transmit power of each transmitting node is $P_i = 1$ power unit, the network can survive up to $\mathcal{L} = \mathcal{E}/P_i = \mathcal{E}$ time unit(s). Now let's assume figure 1(b) and (c) represent broadcast trees given by some

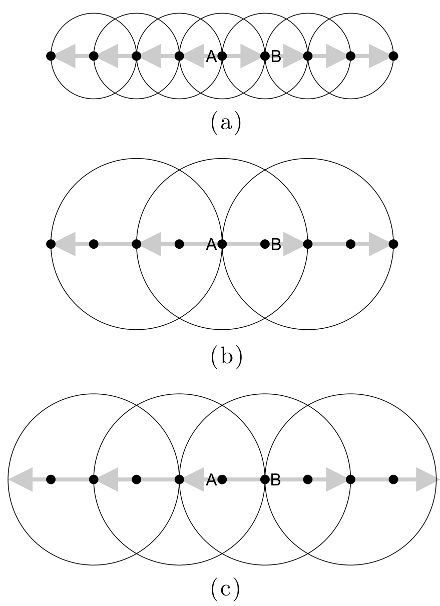


Figure 1. An illustration of three independent broadcast session traffic for demonstrating the load-balancing effect of multiple broadcast sessions. Arrows indicate the broadcast flow directions. (a) Node A is the source. (b) Node A is the source. (c) Node B is the source.

suboptimal algorithms. If we consider these two broadcast session separately, assuming the path loss factor is $\alpha = 2$, the network can survive only up to $\mathcal{L} = \mathcal{E}/4$, which is four times smaller than the case in figure 1(a). In contrast, from now on, let's consider scheduling these two broadcast sessions given in figure 1(b) and (c) so that two routing trees from different sources A and B are alternating in a round-robin fashion with an equal length duty cycle (1 second for example). It is clear that in this scenario the network can survive up to $\mathcal{L} = \mathcal{E}/2$ time units. Compared to previous case, the overall network lifetime becomes doubled. This is due to scheduling and its inherent load-balancing effect, when we consider multiple broadcast sessions from different sources. Because of this inherent load-balancing effect for multiple broadcast sessions, it is really hard to evaluate the "true" performance gain for a specific broadcast session obtained by a specific optimization criteria such as MINTOTAL or MINMAX and cost metrics. This is the most important reason we considered only a single session from a fixed source. Once we consider only a single session, any realistic traffic pattern such as Poisson arrival process becomes non-material, because energy is consumed only when there is transmission, ignoring idle time energy consumption. Therefore, choosing a simple traffic model such as continuous constant bit rate (CBR) traffic can be justified.

3. Static network lifetime

As mentioned earlier, the static network refers to the case when the routing structure does not change over time. Without loss of generality, we assume a broadcast session initializes at time $t = 0$ by constructing a broadcast routing tree T after exchanging control packets, and lasts until one node dies due to battery depletion. Since the node transmit power determined at the initialization stage is not a function of time, we will drop the time dependence in our notation and simply use $P_i(0) = P_i$. Since the control overhead for one-time initialization of the broadcast tree is negligible, $p^C = 0$ in (7), the residual battery energy level of node i at time t is $E_i(t) = E_i(0) - \int_0^t P_i(\tau)d\tau = E_i(0) - P_i \cdot t$.

Given $E_i(t)$, if node i transmits to node j using transmit power $\mathbf{P}_{ij}$, the link (i, j) can be supported up to $E_i(t)/\mathbf{P}_{ij}$ units of time. We call this as the *(residual) link longevity* $\mathbf{L}_{ij}(t) := E_i(t)/\mathbf{P}_{ij}$ of a link (i,j). Similarly, node i constantly transmitting with power P_i can survive for $\ell_i(t) = E_i(t)/P_i$ units of time. Therefore we call this as the *(residual) node longevity* $\ell_i(t)$. The link and node longevity of node i is related to each other as $\ell_i(t) := \min_{j\in\Re_i}\{\mathbf{L}_{ij}(t)\}$ as shown in (5). These notations are summarized in Table 1. Note that the variable dimension of both quantities is a time unit. If 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.

Given a routing tree T determined by some algorithm at time $t = 0$, the corresponding transmit power levels $\{P_i\}$, and the initial battery energy levels $\{E_i(0)\}$, the *static network lifetime* (SNL) $\mathcal{L}(T)$ of the tree T is defined as the time to first node failure, when this tree is constantly used for broadcasting. It is easy to see that this corresponds to minimum node longevity $\mathcal{L}(T) := \min_{i\in N}\{\ell_i(0)\} = \min_{i\in N}\{E_i(0)/P_i\}$. Hence, the network lifetime of a tree T is determined by a node with the minimum node longevity, which in turn determined by the link with minimum link longevity as follows:

$$\mathcal{L}(T) := \min_{i\in N}\{\ell_i(0)\} = \min_{i\in N}\left\{\frac{E_i(0)}{P_i}\right\} \tag{11a}$$

$$= \min_{i\in N}\left\{\min_{j\in\Re_i}\mathbf{L}_{ij}(0)\right\} = \min_{(i,j)\in T}\left\{\mathbf{L}_{ij}(0)\right\}. \tag{11b}$$

where, by definition of the logical neighbor, $T = \{(i, j) \mid i \in N, j \in \Re_i\}$. We now present a few interesting cases as examples.

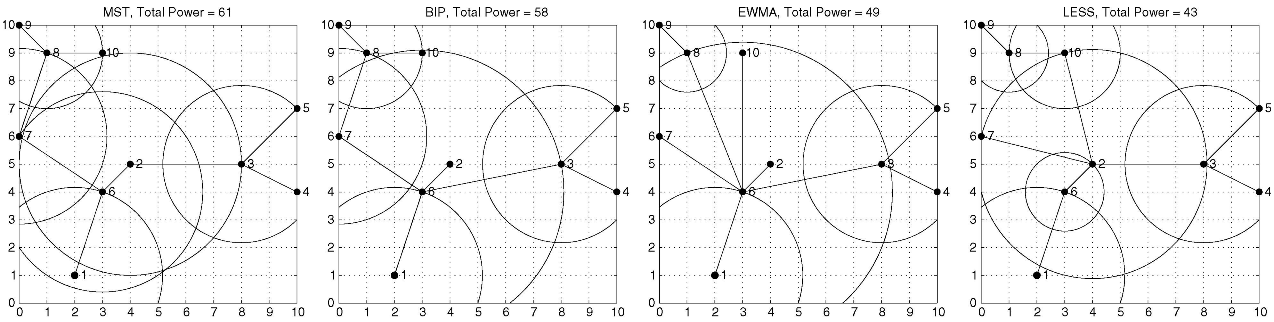
3.1. Some examples

Now, let us examine a few simple examples on how these definitions can be applied to different scenarios.

Example 1. Figure 2 shows some examples of known power-efficient trees constructed using MST, BIP, EWMA and LESS algorithms on $n = 10$ nodes distributed within 10×10 deploy region. In this example, we will show that the MINTOTAL transmit power does not necessarily imply maximum lifetime. We assume the path loss factor is $\alpha = 2$. Node 1 is picked as the source node. Although the direction of each link is not indicated, the direction of every link should be interpreted as directed away from the source. For instance, let us analyze the BIP tree T_{BIP} shown in figure 2(b). In this tree, the logical neighbor of node 6 is $\Re_6 = \{2, 3, 7\}$ which corresponds to the child nodes of node 6 in the tree. The physical neighbor of node 6 is $\aleph_6 = \{1, 2, 3, 7, 10\}$. For reader's convenience, the pairwise transmit power matrix matrix $[\mathbf{P}_{ij}]$ is presented:

$$[\mathbf{P}_{ij}] = \begin{bmatrix} 0 & 20 & 52 & 73 & 100 & 10 & 29 & 65 & 85 & 65 \\ 20 & 0 & 16 & 37 & 40 & 2 & 17 & 25 & 41 & 17 \\ 52 & 16 & 0 & 5 & 8 & 26 & 65 & 65 & 89 & 41 \\ 73 & 37 & 5 & 0 & 9 & 49 & 104 & 106 & 136 & 74 \\ 100 & 40 & 8 & 9 & 0 & 58 & 101 & 85 & 109 & 53 \\ 10 & 2 & 26 & 49 & 58 & 0 & 13 & 29 & 45 & 25 \\ 29 & 17 & 65 & 104 & 101 & 13 & 0 & 10 & 16 & 18 \\ 65 & 25 & 65 & 106 & 85 & 29 & 10 & 0 & 2 & 4 \\ 85 & 41 & 89 & 136 & 109 & 45 & 16 & 2 & 0 & 10 \\ 65 & 17 & 41 & 74 & 53 & 25 & 18 & 4 & 10 & 0 \end{bmatrix}$$

The transmit power of node 6 is $P_6 = \max_{j\in\Re_6}\{\mathbf{P}_{6j}\} = \mathbf{P}_{63} = 26$ power units. Likewise, node 1, 3, 6, 7, and 8 are transmitting nodes with transmit power $P_1 = 10$, $P_3 = 8$, $P_6 = 26$, $P_7 = 10$, $P_8 = 4$. The transmit power of other leaf nodes are 0. Thus, the total transmit power of T_{BIP} is $\mathcal{P}_{TX}(T_{BIP}) = \sum_{i\in N} P_i = 58$. Similarly, for other trees, $\mathcal{P}_{TX}(T_{MST}) = 61$, $\mathcal{P}_{TX}(T_{EWMA}) = 49$. and $\mathcal{P}_{TX}(T_{LESS}) = 43$.



(a) MST (b) BIP (c) EWMA (d) LESS

Figure 2. Broadcast routing trees corresponding to power-efficient algorithms including (a) minimum spanning tree (MST), (b) broadcast incremental power (BIP) [5], (c) embedded wireless multicast advantage (EWMA) [34], and (d) largest expanding sweep search (LESS) [36] for a sample network configuration of n=10 nodes: the node with ID=1 is the source node. The path loss factor α=2 is used. For details of each algorithm, readers are referred to the original references.

As seen in this example, our LESS algorithm performs better than other algorithms.

Again let's consider T_{BIP}. Let's assume $[E_i(0)] = [900, 200, 900, 400, 700, 500, 900, 900, 100, 300]$. The link longevity from node 6 to its logical neighbor is $\mathbf{L}_{62} = \frac{500}{2}, \mathbf{L}_{63} = \frac{500}{26}, \mathbf{L}_{67} = \frac{500}{13}$. Hence the node longevity of node 6 is $\ell_6 = \min_{j \in \Re_6}\{\mathbf{L}_{6j}\} = \mathbf{L}_{63} = 19.23$. Similarly, $\ell_1 = \frac{900}{10}, \ell_3 = \frac{900}{8}, \ell_7 = \frac{900}{10}, \ell_8 = \frac{900}{4}$, and the leaf nodes 2, 4, 5, 9, and 10 have infinite node longevity. Thus, the static network lifetime is $\mathcal{L}(T_{BIP}) = \ell_6 = 19.23$ which is determined by the edge (6, 3). Likewise, $\mathcal{L}(T_{MST}) = \ell_2 = \frac{200}{16} = 12.5, \mathcal{L}\,(T_{EWMA}) = \ell_6 = \frac{500}{29} = 17.24, \mathcal{L}\,(T_{LESS}) = \ell_2 = \frac{200}{17} = 11.76$. These values are summarized in the following Table 2.

Thus, although LESS consumes the smallest total transmit power, it has also the shortest network lifetime. This is not surprising, since the lifetime is a function of both battery energy and node transmit power, whereas MINTOTAL transmit power optimization criterion, for which these algorithms are optimized, does not incorporate the battery energy and hence can not guarantee the longest lifetime. In the next section, we will provide a systematic solution to obtain a tree with the maximum static network lifetime.

Example 2 (Equally Distributed Energy Network (EDEN))**.** As a special case, when all nodes in a network have identical initial energy levels (i.e., $E_1(0) = \ldots = E_n(0) = \mathcal{E}$), we will refer to the network as an *equally distributed energy network* (EDEN). The lifetime of EDEN can be expressed as $\mathcal{L}_{EDEN}(T) = \mathcal{E}/\max_{(i,j)\in T}\{\mathbf{P}_{ij}\}$.

Table 2
Summary of important values of the trees shown in figure 2.

Algorithm	$\sum_{i \in N} P_i$	$\max_{i \in N}\{P_i\}$	$\mathcal{L}(T)$
MST	61	16	$\ell_6 = 12.5$
BIP	58	26	$\ell_2 = 19.23$
EWMA	49	29	$\ell_6 = 17.24$
LESS	43	17	$\ell_2 = 11.76$

Example 3 (Flooding)**.** Another common example is flooding [23]. Although we concentrate on the network lifetime for a tree, the definition as a minimum node longevity (11a) is applicable to any general topology. In flooding, each node caches the messages it has previously received. If a message currently received is in the cached list, it simply drops the message. Otherwise, each node retransmits the message with the same maximum power $P_{\max}$. Hence, each node transmits each message exactly once even though it can receive the same message multiple times. In this case, the network lifetime for flooding is $\mathcal{L}(flooding) = \min\{\frac{E_1(0)}{P_{\max}}, \ldots, \frac{E_n(0)}{P_{\max}}\} = \frac{\min_{i\in N}\{E_i(0)\}}{P_{\max}}$.

4. Maximizing static network lifetime

4.1. Problem formulation

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 broadcast routing tree is used for the whole remaining time. We want to find a static routing tree T^* which is not necessarily unique, but gives the maximum network lifetime. This problem can be formulated as:

$$\text{maximize } \mathcal{L}(T) = \min_{(i,j)\in T}\{\mathbf{L}_{ij}\} \text{ over } \forall\, T \subset G. \qquad (12)$$

The global optimal solution to this problem is referred to as the *optimal static network lifetime* $\mathcal{L}^* := \max_{T\subset G}\{\mathcal{L}(T)\} = \max_{T\subset G}\min_{(i,j)\in T}\{\mathbf{L}_{ij}\}$. This problem can also be considered as a range assignment problem [25], but to obtain a tree, we additionally need to determine the connectivity matrix.

Note the analysis presented in Section 3 is a *network analysis* problem: given T, we can derive all information we want. On the other hand, this MAXMIN (or bottleneck) optimization problem (12) is a *network design* problem in which we need to find an optimal spanning tree T, or equivalently, the optimal power assignment $\{P_i\}$ to each node. We will show in the following subsections that we can in fact find a (global) optimum solution to this problem by a polynomial-time greedy algorithm. The presentation in this section is a quick summary of our earlier work [35]. Due to space limitation, the proofs of the lemmas and corollaries presented later will be omitted. We refer the reader to our original paper [35] for the proofs.

4.2. A special case (EDEN)—Undirected graph

We initially consider a special case (Example 2) when all the nodes have identical battery energy levels $E_i(0) = \mathcal{E}, \forall i$. Although this constraint is possibly too stringent in real situations, we solve this problem first because it provides insights into the more general case of unequal battery energies. Due to the equal energy assumption, the graph can be considered as undirected, because $\mathbf{L}_{ij} = \mathcal{E}/\mathbf{P}_{ij} = \mathcal{E}/\mathbf{P}_{ji} = \mathbf{L}_{ji}$. We define the edge weight as $\mathbf{W}_{ij} = \mathbf{P}_{ij}$, if $d_{ij} \leq R_{\max}$, and $\mathbf{W}_{ij} = \infty$, if $d_{ij} > R_{\max}$.

Let $\mu(T) := \max_{(i,j)\in T}\{\mathbf{W}_{ij}\}$ be the maximum edge weight of a tree T and we will refer to the edge satisfying this condition as the *bottleneck edge* of the tree T. We also define $\mu^* := \min_{T\subset G}\{\mu(T)\}$. A *bottleneck spanning tree* (BST) denoted T_{BST} is a spanning tree which has the minimum bottleneck edge weight among all spanning trees such that $T_{BST} := \arg\min_{T\subset G}\{\mu(T)\}$. Let $\mathcal{T}_{BST} = \{T \mid \mu(T) = \mu^*, T \subset G\}$ be the set of all BST's. The following lemma shows that the minimum weight spanning tree (MST) is a bottleneck spanning tree.

Lemma 1. (MST has minimum bottleneck edge weight.) Let T_{MST} denote a minimum spanning tree over an undirected weighted graph G. Then the bottleneck edge weight of the MST is the smallest among all spanning trees $T \subset G$, i.e.,

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

The problem can be rephrased as showing whether $T_{MST} \in \mathcal{T}_{BST}$ holds. This is a standard problem in the graph theory and the proof is omitted. Interested readers are referred to [15,35] for details. We note that for a given specific network topology, when $\mathbf{W}_{ij} = \mathbf{P}_{ij}$, the bottleneck edge weight μ^* corresponds to the "critical power" discussed in [39].

Corollary 1. If $E_i(0) = constant, \forall i \in N$ (EDEN), the minimum spanning tree is a globally optimal solution to the static network lifetime maximization problem.

Lemma 1 shows that the MST is a sufficient condition to be a bottleneck spanning tree. However, it is not a necessary condition in general [15]. That is, MST is a globally optimal solution but *not* unique in general. In fact, any tree $T \in \mathcal{T}_{BST}$ with the same bottleneck weight edge serves our purpose equally well. We also emphasize that MST is *not* a tree with minimum total node transmit power and does not exploit the wireless broadcast advantage property during its construction as in [5]. However, after the tree is constructed, the node transmit power is calculated considering the wireless broadcast advantage using $P_i = \max_{j\in\Re_i}\{\mathbf{P}_{ij}\}$.

4.3. General case—Directed graph

When we consider the general distribution of the initial battery energy levels (i.e. there exist $E_i(0) \neq E_j(0)$), the graph is no longer undirected. Note that although $\mathbf{P}_{ij} = \mathbf{P}_{ji}$, $\mathbf{W}_{ij} = E_i(0)/\mathbf{P}_{ij} \neq E_j(0)/\mathbf{P}_{ji} = \mathbf{W}_{ji}$. In Section 4.2-B, we found that a global optimal solution for EDEN is MST and then obtained an important result: the *minimization of the total cost leads to minimization of the maximum cost for an undirected graph (but not vice-versa)*. It is natural to ask the question whether this analogy carries over to a directed graph as well. Unfortunately, the answer is negative.

Lemma 2. (see [35].) Minimum total weight arborescence rooted at the source node does not necessarily minimize the maximum weight of a directed graph.

Thus, the results for an undirected graph in Lemma 1 cannot be directly applied to a directed graph. In [35], we presented a two step algorithm in which we first find by binary search techniques the sparse topology consisting of edges whose edge weights are bounded by the bottleneck edge weight. However, we will show in the following lemma that the first step of the algorithm in [35] is redundant and only applying Prim's algorithm [15] on a directed graph from the source node by inspecting only outgoing edges at each step is enough to produce an arborescence with MinMax edge weight. Since the original Prim's algorithm is for a minimum spanning tree on an undirected graph, we will conveniently call this algorithm as the *Directed Minimum Spanning Tree* (DMST). Note that DMST does not necessarily produce an minimum total weight arborescence rooted at the source node in general. Since the time complexity of Prim's algorithm is $O(m + n \log n)$ using Fibonacci heap [15], the overall time complexity of the DMST algorithm for a fully connected graph is $O(n^2)$. This is a slight enhancement over the algorithm in [35] since running time was $\Theta((n + m) \log m)$.

Lemma 3. Let T_{DMST} denote a directed tree obtained by the DMST algorithm. For any directed weighted graph, the bottleneck edge weight $\mu(T_{DMST})$ of the DMST tree is minimum among all arborescences, i.e.,

$$\mu(T_{DMST}) = \min_{T\subset G}\{\mu(T)\}.$$

Proof: Let (i^*, j^*) denote the bottleneck edge of the DMST tree T_{DMST}. Removing (i^*, j^*) from T_{DMST} partitions the tree into two subtrees T_1 and T_2 where T_1 includes the source node S. For any directed edge $(i, j) \in T_1 \times T_2$, $\mathbf{W}_{i^*j^*} \leq \mathbf{W}_{ij}$, because DMST chooses an edge with the smallest cost among outgoing edges from the nodes in T_1. On the other hand, for any arbitrary tree $T\subset G$ to be connected, an edge $(i', j') \in \{(i, j) \mid i \in T_1, j \in T_2\}$ should be chosen. Hence $\mu(T) \geq \mathbf{W}_{i'j'} \geq \mathbf{W}_{i^*j^*} = \mu(T_{DMST}), \forall T \subset G$, and the result immediately follows. □

Corollary 2. Let $\mathbf{W}_{ij} = \mathbf{L}_{ij}^{-1} = \mathbf{P}_{ij}/E_i(0)$ be the inverse of link longevity (or normalized pairwise transmit power), then DMST is a (globally) optimal broadcast routing tree solution for static network lifetime maximization problem.

Note that distributed algorithms for minimum weight spanning tree over both undirected and directed graphs are presented in [14] and [27], respectively.

To illustrate this point, figure 3 shows the topology $\tau = \{(i, j) \mid \mathbf{L}_{ij}(0) \geq \mathbf{L}_{63}(0), i \in N, j \in \aleph_i\}$ which consists of all links such that the link longevity is larger than $\mathbf{L}_{63}(0)$. In this example, the link (6,3) is the critical link, without which node 3, 4, and 5 can not be reached from the source node 1. A link is drawn with a solid line, if it is a bidirectional link, i.e., $(i, j) \in \tau$ and $(j, i) \in \tau$; otherwise, it is drawn with a dotted line and the arrow in the middle of the link indicates the direction of each edge. It is clear that the graph is directed: node 1, 3, 6, 7 and 8 which have relatively large battery energy can reach node 2, but node 2 can reach only node 6.

Consequently, we will call DMST algorithm in conjunction with edge weight $\mathbf{W}_{ij} = \mathbf{P}_{ij}/E_i(0)$ as *Maximum Static Network Lifetime* (MSNL) algorithm. Note that MST (or BST) with edge weight $\mathbf{W}_{ij} = \mathbf{P}_{ij}$ provides the MinMax transmit power solution. This is optimal only for undirected graphs and hence is equivalent to MSNL only for EDEN. In general, the solution obtained by MST is not optimal. Note that while MSNL algorithm achieves the maximum static network lifetime, it is not the best solution in terms of total transmit power with the same bottleneck edge constraint. To get smaller total transmit power with the same optimal static network lifetime, we may apply LESS algorithm on the topology obtained by MSNL. However, this approach is not taken in favor of the computational efficiency of the Prim's algorithm.

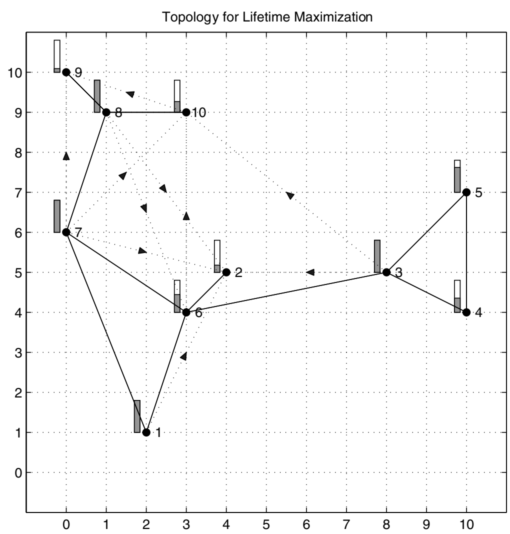


Figure 3. The underlying topology of Example 1. This figure shows the topology $\tau = \{(i, j) \mid \mathbf{L}_{ij}(0) \geq \mathbf{L}_{63}(0), i \in N, j \in \aleph_i\}$. A link is drawn with a solid line, if it is a bidirectional link, i.e., $(i, j) \in \tau$ and $(j, i) \in \tau$; otherwise, it is drawn with a dotted line and the arrow indicates the direction of flow.

5. Extending dynamic network lifetime

From now on, we look at the problem of maximizing dynamic network lifetime (DNL). The dynamic (or self-reconfigurable) network refers to the case when the routing structure is allowed to change adaptively over time. Intuitively, the optimal dynamic network lifetime $\mathcal{L}^\circ$ will always be longer than or equal to the optimal static network lifetime $\mathcal{L}^*$. We will consider $\{E_i(t)\}$ in the cost metric and hence the network is naturally modeled as a directed graph. We note that neither MinMax nor MinTotal transmit power is a valid figure of merit for dynamic networks, since both approaches depend solely on the location of nodes (or on the time-invariant matrix $[\mathbf{P}_{ij}]$). Thus the current residual battery energy has no effect on the construction of trees as demonstrated in Example 1.

If we allow the updates in routing tree, the optimality of MSNL algorithm discussed in the prior section no longer holds. For instance, given $\{E_i(t)\}$, suppose a time-dependent spanning tree $T(t)$ is constructed at time t by MSNL algorithm and is used as a broadcast routing tree for a period Δt. The energy distribution at time $t + \Delta t$ changes to $\{E_i(t + \Delta t)\} = \{E_i(t) - \Delta t \cdot P_i(t)\}$ without considering control overhead. Since the edge weights change to $\mathbf{P}_{ij}/E_i(t + \Delta t)$ accordingly, the bottleneck edge weight $\mu(T(t + \Delta t))$ may change and hence the optimality of static network lifetime does not hold true anymore. However, temporal optimality in lifetime can be reclaimed with the following strategies:

- Triggered update: whenever the bottleneck edge changes, the tree is updated. Constantly monitoring the change requires excessive control overhead.
- Periodic update: instead, we can take a proactive approach such that the tree is updated at every specified update interval Δt.

In this paper, we will exclusively use the periodic update strategy. Trees are updated at time $t = k \cdot \Delta t$ $(k = 0, 1, \ldots)$, and the amount of energy consumed during the time period $(\Delta t \cdot P_i(k\Delta t))$ is subtracted from the corresponding residual energy level $E_i(t)$ using the linear battery discharge model, i.e., $E_i((k+1)\Delta t) = E_i(k\Delta t) - \Delta t \cdot P_i(k\Delta t) - E_{\text{control}}$ where E_{control} denotes the energy consumption due to control overhead. Note that if the update interval is larger than the static network lifetime of the initial tree $T(0)$, i.e, $\Delta t \geq \mathcal{L}(T(0))$, the DNL is equivalent to the SNL. In extending the dynamic network lifetime, we do not claim the global optimality as in the previous section and will investigate which optimization criteria such as MinMax and MinTotal cost provide better performance, the choice of better cost metrics, the proper length of update interval and the trade-offs with control overhead.

Incorporating and utilizing the residual battery energy into cost metric results in randomizing the routing structure over time even for a stationary network, which effectively performs *load-balancing* [7,12] by distributing energy dissipation evenly among the nodes throughout the network operation time and extend network lifetime. Load-balancing of battery energy makes the network lifetime less sensitive to a specific initial energy distribution and what becomes more important is the whole amount of energy, energy pool, in the network.

5.1. A case study

Before we proceed further, it is instructive to revisit the sample example discussed in Example 1. This time, the network is dynamic. The initial energy is distributed as $[E_i(0)] = [900\,200\,900\,400\,700\,500\,900\,900\,100\,300]$ as before. The battery energy level of each node is displayed with a rectangle where the full capacity is assumed 900.

Following the previous discussion, suppose we apply MSNL algorithm every 10 second ($\Delta t = 10$). Incidentally, the MSNL tree $T_{MSNL}(t)$ at the first interval $0 \leq t < 10$ is same as BIP tree T_{BIP} shown in figure 2(b). Note that in this example the optimal SNL was $\mathcal{L}^* = 19.23$ seconds. Remarkably, by updating the tree only 4 times which are drawn in figure 4(a)–(d), the network lifetime can be extended to 45.53 seconds which is 237% of the original static case. In figure 2(b), node 6 is transmitting to reach node 3. In the next time interval $10 \leq t < 20$ shown in figure 4(a), since node 2 and 6 have small battery energy, node 1 decides to transmit with larger transmit power. Next, node 8 in figure 4(b) takes charge to transmit to node 3. Repeating this way, these figures demonstrate the load-balancing effect of battery energy of MSNL algorithm, and show that the network lifetime can be significantly extended.

Figure 4(e) shows the time evolutionary behavior in terms of total transmit power for $\Delta t = 1$. In general, the total transmit power tends to increase as time progresses. We can also observe that there are oscillations. If we perform an off-line optimization, we can reduce the oscillations by grouping the same trees over the period of network lifetime as shown in figure 4(f) and obtain the same network lifetime. The area under the curve represents the amount of energy consumed by all nodes in the network before the first node dies. The dynamic network lifetime for various values of updated interval is plotted in figure 4(g), which shows that reducing the update interval does not provide further improvement. As noted earlier, if $\Delta t \geq \mathcal{L}(T_{MSNL}(0)) = \mathcal{L}^*$, the DNL is equivalent to the optimal SNL, which is indicated by curves corresponding to $y = x$ in figure 4(g).

5.2. Problem formulation

For dynamic network, the transmit power $P_i(t)$ is a time dependent function. The node longevity ℓ_i of node i for dynamic network is the time at which the relation $E_i(\ell_i) = 0$ holds, or equivalently, rewriting (10) as $\int_0^t P_i(\tau)d\tau = E_i(0) - E_i(t)$ and setting $t = \ell_i$, the node longevity ℓ_i is the time at which the following relation holds:

$$\int_0^{\ell_i} P_i(\tau)\, d\tau = E_i(0). \tag{13}$$

The problem of maximizing the dynamic network lifetime (DNL) is equivalent to maximizing the minimum node longevity satisfying (13) by finding a set of time-dependent transmit power $\{P_i(\tau)\}$ which gives a connected topology at any time. Then the problem can be formulated as:

$$\text{maximize } \min_{i \in N} \{\ell_i\} \text{ over } \{P_i(t)\} \tag{14}$$

subject to:

$$\begin{bmatrix} \{P_i(t)\} \text{ induces a connected topology} \\ \text{at each instance of time.} \end{bmatrix} \tag{15}$$

We refer to the *optimal dynamic network lifetime* as

$$\mathcal{L}^\circ := \max_{\{P_i(t)\}} \min_{i \in N} \{\ell_i\}.$$

Considering that the formulation (13) is a functional of $P_i(t)$ and (14) is an optimization of a system of functionals, we may need to approach this problem using the calculus of variation techniques [40]. However, since the constraint (15) requiring the network connectivity is a graph-theoretical concept, it is very challenging applying the conventional calculus of variation techniques. Note that finding an optimal solution to this problem is currently unknown. We do not claim the global optimality as in previous static network case. Instead we will rely on heuristics to extend the dynamic network lifetime.

5.3. Heuristics to extend dynamic network lifetime

In this section, we will discuss two heuristics and its variations to extend the dynamic network lifetime. The heuristic approach adopted in this section is not new. Similar ideas can be found in [7,9,12] including our earlier work [37]. Also some recent related developments can be found in [43]. However, the key differences between the earlier works and our current work are as follows: (i) At each update interval, we have a clear optimization principle, e.g., maximizing the static network lifetime; (ii) Unlike the prior work, we investigate the effect of control overhead in route updates; (iii) The link cost metric considered is different from that of [12,37] and leads to up to 25% longer network lifetime than [12], which will be presented in the simulation section; (iv) We explain the reason for such gain in lifetime in Section 5; (v) We also provide comparison with an upper bound to dynamic network lifetime. Because EWMA and LESS heavily rely on geometric concepts, it is difficult to utilize these algorithms in a similar manner.

(1) *MST + Inverse Link Longevity = WMST:* This metric corresponds to the best effort (greedy) approach to extend the lifetime by applying MSNL algorithm at each update interval Δt. A snapshot of tree $T_{MSNL}(t)$ by the MSNL is made as already explained in detail using the case study in Section 5.1. Note that MSNL is equivalent to MINMAX inverse link longevity:

$$\begin{aligned} \mathit{MSNL} &= \mathit{DMST} + \textit{inverse link longevity} \\ &= \text{Min Max } \textit{inverse link longevity}, \end{aligned}$$

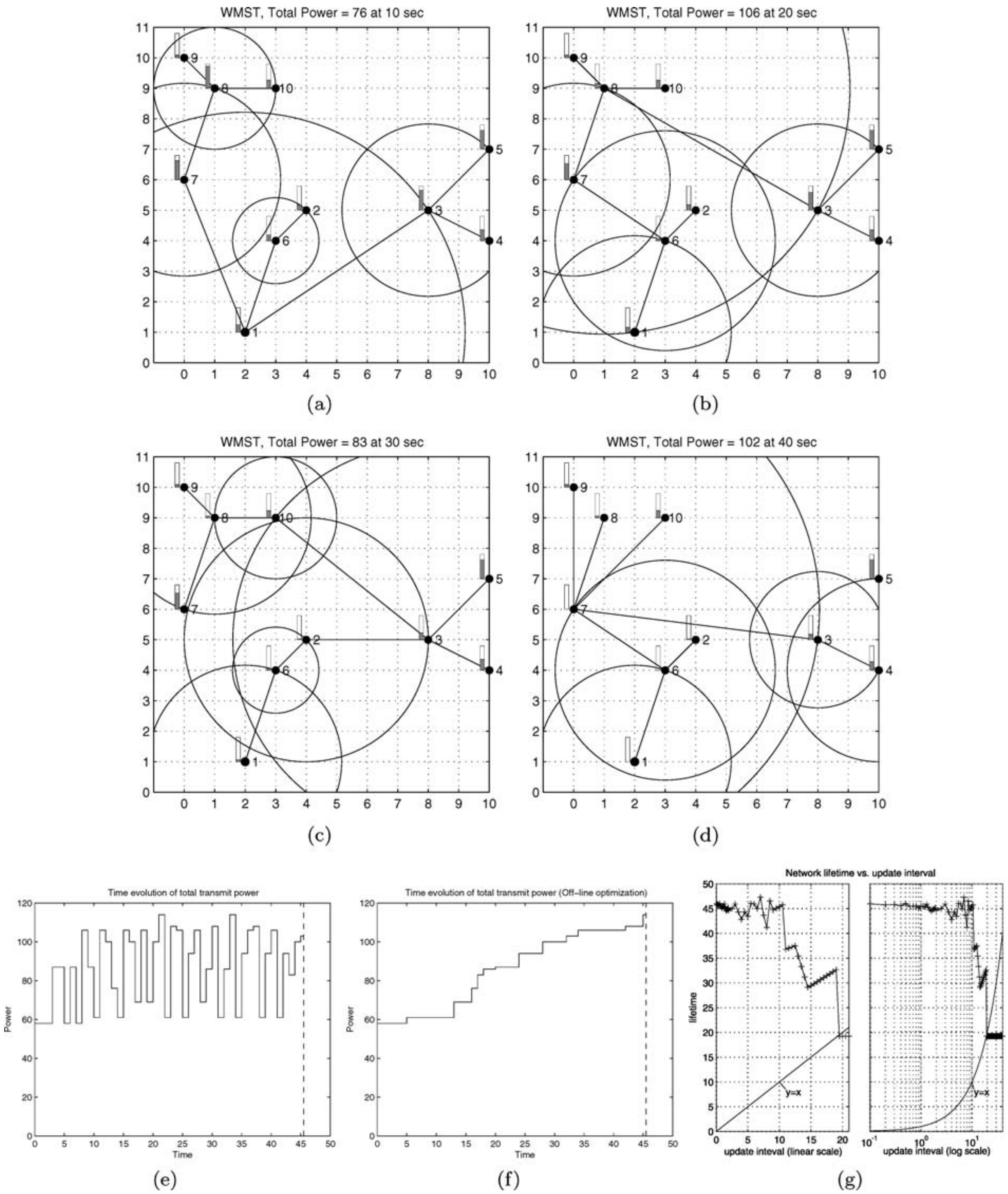


Figure 4. A case study for the topology in figure 2. $\alpha = 2$ and $\Delta t = 10$, (a) $T_{MSNL}(t)$ for $10 \leq t < 20$, (b) $T_{MSNL}(t)$ for $20 \leq t < 30$, (c) $T_{MSNL}(t)$ for $30 \leq t < 40$, (d) $T_{MSNL}(t)$ for $40 \leq t < 45.53$, (e) total power vs. time, (f) oscillation reduction when off-line optimization, and (g) network lifetime vs. update interval in linear and log scales.

where the edge weight is given as the inverse link longevity $\mathbf{W}_{ij}(t) = \mathbf{L}_{ij}^{-1}(t) = \mathbf{P}_{ij}/E_i(t)$. In fact, this algorithm covers a class of link cost metrics $(\mathbf{P}_{ij}/E_i(t))^{\mu}$ where $\mu > 0$, because x^{μ} for $\mu > 0$ is a monotonic increasing function of x and DMST algorithm only depends on the ordering of link weights. The obtained trees will be identical regardless of the value μ. Since we will consider the general link cost metric, we will call the class of algorithms with these cost metrics as Weighted MST (WMST). Indeed, the optimization criteria chosen in this heuristic correspond to MINMAX cost.

(2) *BIP + Inverse Link Longevity = WBIP*: Another heuristic adopted here is a variant of BIP algorithm. The same link cost metric $\mathbf{W}_{ij}(t) = \mathbf{P}_{ij}/E_i(t)$ is used as above. However,

the decision made by BIP at each step of the algorithm is the minimum incremental inverse link longevity $\mathbf{L}_{ik}^{-1}(t) - \mathbf{L}_{ij}^{-1}(t) = (\mathbf{P}_{ik} - \mathbf{P}_{ij})/E_i(t)$ instead of the minimum incremental power $(\mathbf{P}_{ik} - \mathbf{P}_{ij})$. We refer to the algorithm as Weighted BIP (WBIP). A time-dependent (dynamic) tree at each instance of time is found. Considering this cost can be interpreted as a normalized transmit power, the computational complexity at each update is the same as BIP algorithm $O(n^3)$. As the BIP algorithm is for minimizing the total transmit power, we can interpret that the BIP heuristic "loosely" corresponds to MINTOTAL normalized transmit power.

If all nodes have almost the same initial battery energy, the tree selected at the beginning is very close to BIP tree. However, over time, the optimization problem evolves and deviates from the original MINTOTAL transmit power. This, in turn, leads to an increase in the total transmit power to reach every node in the network. To improve the lifetime performance, we also apply post sweeping [5] at each update interval of the dynamic routing trees and call these as WBIP with sweep (WBIPSW) and WMST with sweep (WMSTSW), respectively.

5.4. Control overhead vs. update interval

Regardless of whether an on-demand or a proactive scheme is used, to make use of tree-based algorithms such as MSNL, WMST or WBIP, every node needs to keep track of up-to-date global state information such as the location of all other nodes and their current residual battery energies. Maintaining such information by exchanging control packets incurs control overhead. The algorithms for dynamic networks considered in this paper are deterministic and centralized. Hence, at least in theory, the knowledge of initial global information makes possible for every node to calculate the future routing trees and to estimate other nodes' battery level subsequently without demanding further update communication overhead. However, in practice, such a scheme is very limited in that, if there are other causes of energy consumption such as simultaneous unicast or multicast traffics in different channels, the estimation will not be correct and algorithms will not work properly. The periodic update scheme adopted in this paper does not have such shortcomings, since the information is used only over one update period.

Suppose the bit rate over the wireless link is denoted R_b and the necessary state information is maintained by flooding periodic beacon signals of short control packets of length L bits. We assume that these control packets consist of the broadcast session ID, node ID, current battery energy level, x and y axis coordinates, and parity checksum. Let's assume the control packets are transmitted at a fixed transmit power level P_{control}. Each node needs to flood n control packets at every update interval and hence overall $O(n^2)$ control packets are required for the whole network. The amount of energy consumption by the control packets E_{control} per node per update interval is $E_{\text{control}} = P_{\text{control}} \frac{Ln}{R_b}$. Note that energy consumption by control overhead depends on the size of networks and hence $E_i((k+1)\Delta t) = E_i(k\Delta t) - \Delta t \cdot P_i(k\Delta t) - P_{\text{control}} \frac{Ln}{R_b}$.

Possessing the most up-to-date information provides better chance to build a more efficient routing solution. However, there is a trade-off with the update interval: if broadcast tree is updated too often, the benefit of having current state information diminishes and most of the energy will be consumed exchanging control packets instead of payload broadcast session traffic.

5.5. Impact of initial energy distribution on network lifetime

The general form of the edge weight function $\mathbf{W}_{ij}(t)$ including the initial energy $E_i(0)$, the residual energy $E_i(t)$, and the pairwise transmit power $\mathbf{P}_{ij}$ of each node i first appeared in Chang et al. [10] as

$$\mathbf{W}_{ij}(t) = \mathbf{P}_{ij}^{\lambda} E_i(0)^{\mu} E_i(t)^{-\nu} \tag{16}$$

with non-negative weighting factors λ, μ, ν. When $\lambda = 1$ and $\mu = \nu$ (normalized residual energy), the average and the worst case performance of *unicast* routing is shown to be the best by simulations [10]. Recent papers (see for example [12]) have also adopted the same following cost metric for the simulation of *broadcast* routing:

$$\mathbf{W}_{ij}(t) = \mathbf{P}_{ij} \left(\frac{E_i(0)}{E_i(t)} \right)^{\mu}. \tag{17}$$

We now discuss some implications of including the initial energy $E_i(0)$ in the general edge weight function (16).

(1) If $\mu \neq \nu$, in general, placing the initial energy $E_i(0)$ in the numerator is counter-intuitive, because the edges incident to a node with larger initial energy are assigned larger costs and hence the node with larger initial energy tends to be assigned with smaller transmit power or no power, which is not desirable.

(2) If $\mu = \nu$, it may appear that $E_i(0)/E_i(t)$ does have a meaningful interpretation as an inverse of normalized residual energy. However, when lifetime is considered, the following situation illustrates some problem of this case: suppose nodes i and j start with initial energy $\mathcal{E}$ and $\mathcal{E}/100$, respectively and after transmitting for some time both nodes have 10% of the initial energy. The ratio is the same $E_i(0)/E_i(t) = E_j(0)E_j(t)$ but the absolute residual energy $E_i(t)$ of node i is 100 times larger than that of node j. Although it is better to assign large transmit power to node i, the metric can not differentiate this case. Therefore, it is still not a good strategy because normalization does not clearly reveal the absolute residual energy.

(3) Previous literatures using the results of [10] have ignored the fact that [10] performs *off-line optimization*. The main results in [10] imply that, to maximize the network lifetime for unicast, the rate of the flow in each link should be assigned a certain rate to satisfy their optimization criteria. This approach is hard to translate to an actual routing protocol (e.g., routing table), because the unpredictable future traffic (flow) in the real situation cannot be assigned with an optimal flow rate. In the off-line optimization as in [10], the knowledge of the initial energy distribution

$\{E_i(0)\}$ can be fully utilized and hence it does have a positive impact on the aggregated network lifetime. However, for an *on-line optimization* discussed in the previous sections, the inclusion of initial energy distribution is discouraged, because it has a detrimental effect on network lifetime as will be shown by simulation in Section 7.3. What really counts at the current update interval is the (absolute) residual energy level $E_i(t)$. Therefore, except for some special occasions such as EDEN, it is safer in real situations to assume the initial energy as a random variable and not to include it in the deterministic cost metric design. In the case of EDEN, $\{E_i(0)\}$ has no impact and is redundant. Simulation results with and without initial energy will be given in the following section.

5.6. Upper bound to optimal dynamic network lifetime

Because of dynamic nature of this problem, conceptually, what we want to achieve is to fully and efficiently utilize the energy pool with minimum possible power just enough for network connectivity. In order to provide an absolute measure of comparison among different optimization metrics, we propose a straightforward upper bound to the optimal dynamic network lifetime as follows.

Lemma 4. $\mathcal{L}_U^\circ := \frac{\sum_{i\in N} E_i(0)}{\min_{T\subset G}\{\sum_{i\in N} P_i\}}$ is an upper bound to the optimal dynamic network lifetime $\mathcal{L}^\circ$, i.e.,

$$\mathcal{L}_U^\circ := \frac{\sum_{i\in N} E_i(0)}{\min\limits_{T\subset G}\left\{\sum_{i\in N} P_i\right\}}. \tag{18}$$

Proof: Assuming recharging the battery is not an option, we cannot spend more energy than the available initial energy pool $\sum_{i\in N} E_i(0)$. Also, to form a routing (spanning) tree, no matter what kind of tree we use at each update interval, at least the minimum amount of power $\min_{T\subset G}\{\sum_{i\in N} P_i\}$ should be spent to satisfy the network connectivity. Hence, the dynamic network lifetime cannot exceed this upper bound $\mathcal{L}_U^\circ$. □

Lemma 5. The optimal static network lifetime $\mathcal{L}^*$ is strictly upper bounded by $\mathcal{L}_U^\circ$, i.e.,

$$\mathcal{L}^* = \max_{T\subset G}\left\{\min_{i\in N}\left\{\frac{E_i(0)}{P_i}\right\}\right\} < \mathcal{L}_U^\circ. \tag{19}$$

Proof: First, we show that left-hand side of (19) is less than $\sum_{i\in N} E_i(0)/\min_{T\subset G}\{\sum_{i\in N} P_i\}$. Let $a = (a_1, \ldots, a_n)$ be a sequence of positive numbers and $b = (b_1, \ldots, b_n)$. be a sequence of nonnegative numbers where there is at least one nonzero element such that $b_j \neq 0$ for $1 \leq j \leq n$ and also there is at least one zero element such that $b_k = 0$ for $1 \leq k \leq n$. If $m = \min_k\{\frac{a_k}{b_k}\} = \min_{k,b_k\neq 0}\{\frac{a_k}{b_k}\}$, then we have successively for all $1 \leq k \leq n$, $0 < mb_k \leq a_k$, if $b_k \neq 0$, and $0 = mb_k < a_k$, if $b_k = 0$, and $m\sum_{k=1}^n b_k < \sum_{k=1}^n a_k$. Since there is at least one nonzero element in b, $\sum_{k=1}^n b_k \neq 0$. Hence $\min_{1\leq k\leq n}\{\frac{a_k}{b_k}\} < (\sum_{k=1}^n a_k)/(\sum_{k=1}^n b_k)$.

Given a routing tree T in a directed graph G, there exists at least one leaf node (with zero transmit power). $\{P_i\}$ can be calculated from the tree T. Also for the graph G to be connected, there is at least one node with nonzero transmit power. Let $a_i = E_i(0)$ and $b_i = P_i$, then these satisfy the condition for a and b. Because the inequality $\min_k\{\frac{a_k}{b_k}\} < (\sum_k a_k)/(\sum_k b_k)$ holds for arbitrary trees, (19) immediately follows. □

Because the upper bound $\mathcal{L}_U^\circ$ does not contain the notion of first node failure, it is presumably a quite loose upper bound to the optimal dynamic network lifetime. Since finding a MINTOTAL transmit power broadcast routing tree is NP-complete, the upper bound (18) can not be found with polynomial-time algorithms. Hence we will approximate the denominator in (18) with LESS algorithm, currently the best known approximation, in our simulation. We will demonstrate that using the heuristics presented in Section 5.3, we can achieve approximately half of this upper bound to dynamic network lifetime on average.

6. Simulation results for static network lifetime

In this section, we perform simulations using the network model discussed in Section 2.1. Within a 1000×1000 m^2 network deploy region, the network configurations (locations of nodes) are randomly generated according to uniform distribution. All the generated nodes participate in the group of a single broadcast session. The source node S is chosen arbitrarily among them. The initial battery energy distribution $\{E_i(0)\}$ is drawn according to uniform probability distribution $unif(\eta, \xi)$ ranging from the minimum value η to the maximum value ξ which denotes the full battery capacity. The simulation results are for stationary (non-mobile) network topologies as in wireless sensor networks. We consider only energy consumption by RF transmit power of transceivers. We do not consider control overhead to setup the routing tree, because energy consumption incurred by the control overhead at the initialization stage is negligible compared to energy consumption by session-oriented broadcast. Each point in figure 5(a) and (b) represents an average value of 100 different randomly generated network topologies ($\alpha = 2$). The same random seeds are used for valid comparison of each metric. The initial energy $\{E_i(0)\}$ is distributed according to three uniform probability distributions: (i) $unif(10^7, 10^7)$=constant, (ii) $unif(0.5\times 10^7, 10^7)$, and (iii) $unif(0, 10^7)$. The performance of MSNL, MST, BIP, EWMA, and LESS algorithms is compared in terms of the total transmit power and the static network lifetime.[3] Note that, except MSNL algorithm, all algorithms produce static trees that are independent of time.

In figure 5(a), the performance comparison of static trees in terms of total transmit power is presented. Because the total transmit power of MSNL depends on both network topology (locations of nodes) and initial energy distribution, one

[3]From now on, we will use the terms for the algorithms and the corresponding trees obtained by them interchangeably.

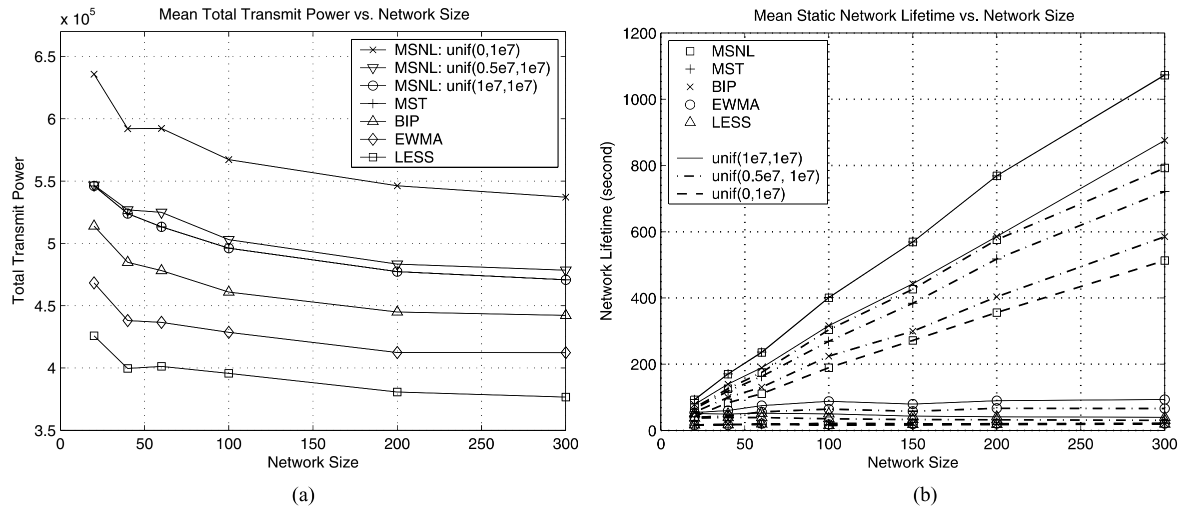


Figure 5. Performance comparison of various power-efficient algorithms including MSNL, MST, BIP, EWMA, and LESS ($\alpha = 2$) in terms of (a) mean total transmit power, and (b) mean static network lifetime.

curve for each energy distribution is shown. For other algorithms, since they only depend on locations, only one curve is shown regardless of initial energy distribution. In general, the total transmit power of all trees decreases as the number of nodes per unit area increases. Hence, per node average transmit power will decrease even more rapidly. From figure 5(a) and (b), one can infer that to prolong the static network lifetime, a large quantity of nodes (e.g., sensors) should be densely deployed in the target environment, because it will result in extended lifetime with small total transmit power. We can observe the superior performance of LESS in terms of total transmit power. Notice that, when all the nodes have identical energy of 10^7 units (EDEN), the curves in figure 5(a) and (b) by MST and MSNL overlap perfectly, which is consistent with the theoretical result given in Section 4.2.

Figure 5(b) summarizes the lifetime performance of static trees for various distributions of the initial battery energy $\{E_i(0)\}$ and the size of the networks n. In general, the static network lifetime increases linearly as a function of the network size per 1×1 km^2 region. If a network has a larger initial energy pool $\sum_{i \in N} E_i(0)$, then the lifetime also increases. When the initial energies are not equal ($unif(0,10^7)$ and $unif(0.5 \times 10^7, 10^7)$), MST is no longer optimal and MSNL always produces longer lifetime, as expected. The separation of MSNL from other algorithms becomes even more significant when $\{E_i(0)\}$ is uniformly distributed from 0 to 10^7 energy units. This is because the max-min lifetime is heavily dependent on the nodes with very scarce initial energy. On average, MST performs better than BIP, EWMA, and LESS. This is due to the fact that BIP, EWMA and LESS algorithms are optimized for a global network property (MINTOTAL transmit power), whereas the MST algorithm is optimized for a local node property (MINMAX transmit power).

Figure 5 shows a favorable adaptive property of MSNL. When every node has sufficient energy, a tree with relatively smaller total transmit power is chosen. On the other hand, when there exist nodes with scarce energy, more emphasis goes to extending the network lifetime guarding against premature death of nodes. Therefore, MSNL trade-offs between network lifetime and total transmit power depending on the current residual battery energy status. In summary, the simulation results support the hypothesis, to maximize the static network lifetime, we should incorporate the residual battery energy in the cost metric, and MSNL constitutes the optimal solution.

7. Simulation results for dynamic network lifetime

The simulation results and analysis on dynamic network lifetime are presented in this section.

7.1. Network lifetime vs. update interval without control overhead

In order to show the effect of update interval (Δt) on dynamic network lifetime (DNL), we do not consider control overhead at each update interval $E_{\text{control}} = 0$, and hence battery energy is updated as $E_i((k+1)\Delta t) = E_i(k\Delta t) - \Delta t \cdot P_i(k\Delta t)$. The choice of update interval is a crucial parameter when designing an energy-efficient routing protocol. Figure 6 summarizes the typical average behavior of the dynamic network lifetime of each algorithm (WMST, WMSTSW, WBIP, and WBIPSW) as a function of update interval for network sizes of $n = 20, 40$, and 60 per 1×1 km^2. The initial energies are distributed according to $unif(0, 10^7)$. Each point in the figures appearing after figure 6 represents an average value of 100 different randomly generated network topologies. Obtaining each point requires a significant number of computations. For instance, in WBIPSW case, for $\Delta t = 0.1$, the network lifetime is approximately 320 seconds on average. This translates to 3200 tree updates. Since each point is an average of 100 topologies, to compute the average values, about 320,000 applications of BIP algorithm along with sweep procedures are required.

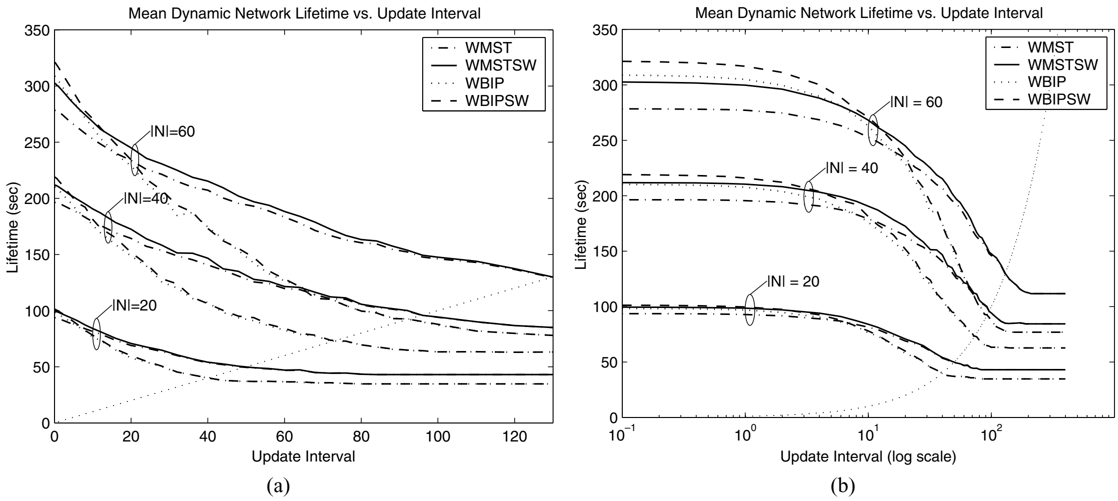


Figure 6. Mean dynamic network lifetime as a function of update interval Δt for $\alpha = 2$, $n = 20, 40, 60$ and $unif(0, 10^7)$ battery energy distribution. (a) Linear scale update interval (b) log scale update interval.

Due to the complexity, we provide the results up to network size $n = 60$.

Figure 6 provides the most useful insights into this problem. Two sub-figures corresponding to linear and log scales of update intervals are presented to clearly visualize its effect. As is clear from the linear scale plot, figure 6(a), the general tendency is DNL decreases as the update interval becomes larger. In other words, the network lifetime generally increases, if the routing tree is updated more frequently. As noted earlier, if $\Delta t \geq \mathcal{L}(T(0))$, DNL is essentially equivalent to SNL. Additionally, for WMST and WMSTSW, if $\Delta t \geq \mathcal{L}(T_{\text{MSNL}}(0)) = \mathcal{L}^*$, DNL corresponds to optimal SNL. This fact is emphasized in figure 6 with a dotted curve ($y = x$). Hence, if the lifetime curves intersect with this linear curve, they become flat after the intersection and the values correspond to the SNL of initial trees as already shown in figure 4(g). Because figure 6 is an average of 100 realizations, it does not exhibit such a sharp transition.

However, as can be observed from the log scale plot, figure 6(b), there exists an upper bound in the network lifetime we can achieve with the heuristic approaches. There is almost no gain, if the update interval is less than 1, $\Delta t \leq 1$, which is common to all metrics we considered. The post sweep procedure has a positive impact on network lifetime, but the gain is quite negligible considering the additional computational power. What is more important is the choice of optimization metrics (WMST or WBIP). Note that also there exists a threshold value τ: if $\Delta t < \tau$, WBIP is beneficial; otherwise, $\Delta t > \tau$, WMST shows better performance. This implies that when the residual battery energy is ample compared to the energy consumption at each interval $E_i(k\Delta t) \gg \Delta t \cdot P_i(k\Delta t)$, MinTotal optimization is preferable. Otherwise, MinMax principle gives better results. Figure 6 suggests that *no single optimization metric performs better than the other in every range of update intervals*. Since the results presented in this section do not account for the energy consumption due to control overhead, the network lifetime performance of any stateless routing scheme which doesn't require periodic maintenance of state information should be compared with this result.

7.2. Network lifetime vs. network size (density)

Figure 7(a) summarizes the dynamic network lifetime performance as a function of node density (number of nodes per 1×1 km^2 region), where different algorithms and link cost metrics are compared for an update interval of $\Delta t = 1$. Note that in figure 6 this value of update interval ($\Delta t = 1$) lies within the region in favor of WBIP over WMST. Hence, WBIPSW consistently outperforms WMSTSW on average, although not by a large margin ($\mathcal{L}_{\text{WBIPSW}} > \mathcal{L}_{\text{WMSTSW}} > \mathcal{L}_{\text{WBIP}} > \mathcal{L}_{\text{WMST}}$). The dynamic network lifetime increases almost linearly as the network density increases. Simulation results corresponding to $\alpha = 3$ and 4 is omitted due to limited space. Nevertheless, let us briefly summarize the results. The behavior is quite different from $\alpha = 2$ case. First, the lifetime curves seem to be superlinear (i.e., x^β, $\beta > 1$) instead of linear. Second, WMST performs better than WBIP on average ($\mathcal{L}_{\text{WMSTSW}} > \mathcal{L}_{\text{WMST}} > \mathcal{L}_{\text{WBIPSW}} > \mathcal{L}_{\text{WBIP}}$). This is because the penalty for using larger transmit power is much larger for $\alpha = 3$ and 4. According to figure 7(a), we can interpret that the increase in network lifetime as the number of node increases is largely due to increase in initial energy pool $\sum_{i \in N} E_i(0)$.

Figure 8(a) shows the average of the ratio of dynamic network lifetime by simulations vs. the upper bound to optimal dynamic lifetime ($\mathcal{L}_{\text{algorithm}}/\mathcal{L}_U^\circ$) as a function of network sizes when $\Delta t = 1$. On average, the periodic tree update scheme achieves roughly a half of the upper bound $\mathcal{L}_U^\circ$. Considering (18) is quite a loose upper bound, we suspect our scheme is quite close to the theoretically achievable limit.

Figure 8(b) shows the percentage of lifetime curve of WMST as a function of total number of updates over the

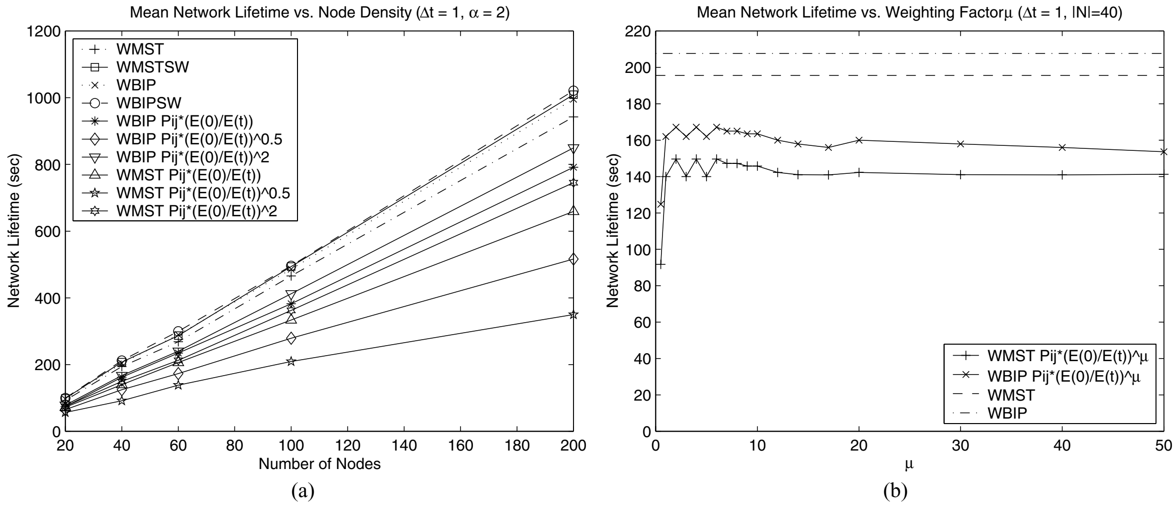


Figure 7. (a) Dynamic network lifetime vs. network size ($\Delta t = 1, \alpha = 2$). (b) Dynamic network lifetime as a function of weighting factor μ in $P_{ij}(E_i(0)/E_i(t))^{\mu}$ ($\Delta t = 1,\ \alpha = 2,\ n = 40$).

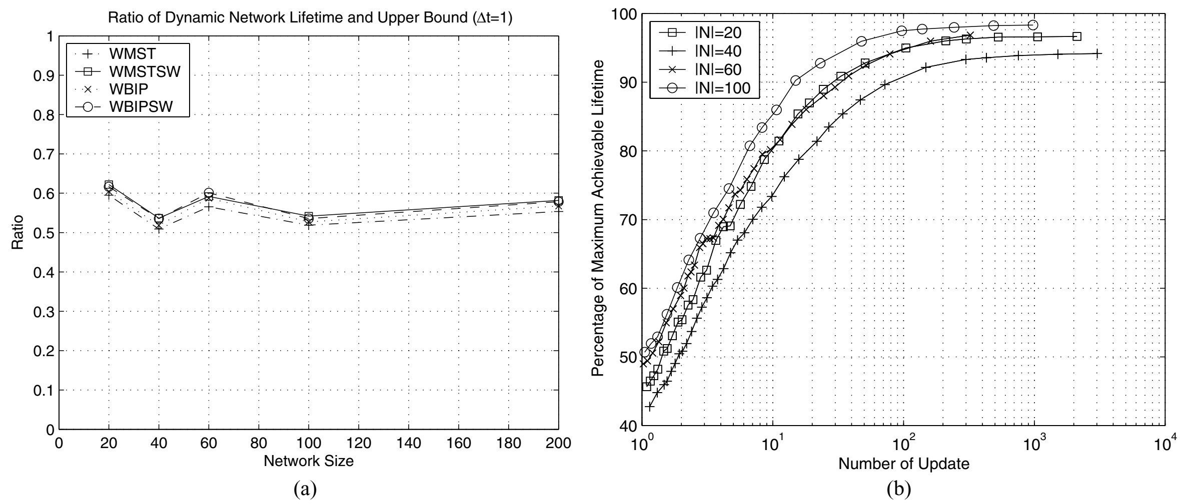


Figure 8. (a) Mean ratio of $\mathcal{L}_{\text{algorithm}}/\mathcal{L}_U^{\circ}$ (b) Percentage of maximum achievable lifetime.

whole lifetime. This figure shows that with a single tree update (two routing trees are used over the lifetime period), about 40∼50% of the maximum achievable lifetime (in our simulations, using WBIPSW updated at every $\Delta t = 0.1$ second). If the tree is updated 10 times over the whole lifetime, about 74∼85% can be achieved. To achieve 90% of the maximum achievable lifetime, the number of required updates is 13, 28, 72 and 33 for $n = 20, 40, 60, 100$, respectively. From this data, the reasonable number of updates seems to be around 10∼100 times.

7.3. *Impact of initial energy distribution*

Using the general cost metric (16), we investigated how the inclusion of initial energy factor $E_i(0)$ in the cost metric affects the dynamic network lifetime. Specifically, we ran experiments assuming $\lambda = 1$ and $\mu = v$, hence we use the form (17) in our simulations. Figure 7(a) compares the performance in terms of network lifetime with and without incorporating the initial energy in the cost metric for various network sizes. The weighting factor μ takes the values from $\{0.5,1,2\}$ and $\alpha = 2$ and $\Delta t = 1$ are used. For this update interval, it was shown in figure 6 that WBIP is generally better than WMST. We can observe that the link cost metrics of WMST or WBIP without initial energy (i.e., $\mathbf{P}_{ij}/E_i(t)$ or $\Delta\mathbf{P}_{ij}/E_i(t)$) produce longer network lifetime compared to $\mathbf{P}_{ij}(E_i(0)/E_i(t))^{\mu}$ or $\Delta\mathbf{P}_{ij}(E_i(0)/E_i(t))^{\mu}$ for every value of μ we considered. This justifies the arguments given in Section 5.5.

In figure 7(b), for fixed values of $n = 40$, $\alpha = 2$ and $\Delta t = 1$, the weighting factor μ was varied from 0.5 to 50 and the cost metrics $\mathbf{P}_{ij}(E_i(0)/E_i(t))^{\mu}$ and $\Delta\mathbf{P}_{ij}(E_i(0)/E_i(t))^{\mu}$ are compared. When $\mu = 2, 4, 6$, the lifetime is longer than other values. However, whichever value of weighting factor μ is used, the lifetime of WMST or WBIP without initial energy in the cost metric was longer by approximately 25%. Therefore, these results seem to suggest that the inclusion of

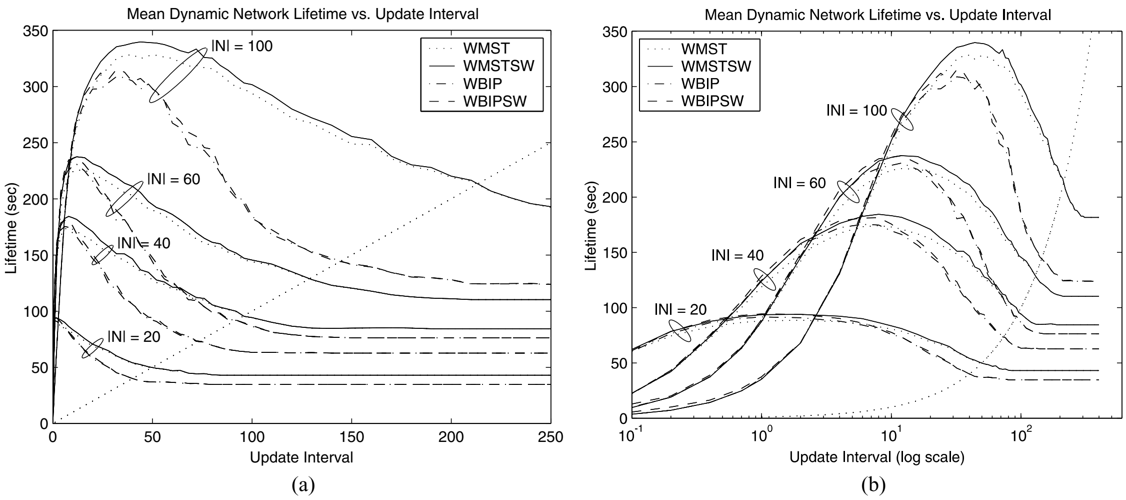


Figure 9. Dynamic network lifetime as a function of update interval Δt including control overhead ($n = 20, 40, 60, 100$, and $\alpha = 2$). (a) Linear scale update interval, (b) log scale update interval.

initial energy should be avoided in an on-line optimization problem.

7.4. Impact on energy consumption by control overhead

In the previous sections, we investigated the enhancement in network lifetime purely in terms of the cost and optimization metric design problem at an algorithm level to isolate the effect of each metric on lifetime. Also the simplified energy dissipation model (10) (energy consumption by RF transmit power) was considered. For algorithms to be translated to a protocol level description, the energy consumption by control overhead cannot be ignored. In this simulation, we suppose the bit rate is $R_b = 1$ Mbps and the length of control packet as $L = 100$ bits. Let's assume the control packets are transmitted with power $P_{\text{control}} = 500^\alpha$. Then, the amount energy consumption by the control packets E_{control} per node per update interval for $\alpha = 2$ is $E_{\text{control}} = P_{\text{control}} \frac{Ln}{R_b} = 25n$. While these values are chosen quite arbitrarily, we believe it is enough to demonstrate the overall effect of the control overhead.

In figure 9, the impact on network lifetime of the energy dissipation by control overhead is shown. If the update interval is very small (e.g., $\Delta t = 0.1$), as the network size becomes larger, most of node energies are consumed due to control overhead and lifetime becomes significantly reduced. Figure 9 suggests the following:

- Considering the energy consumption by control overhead, the update interval cannot be arbitrarily small.
- There exists a certain peak value of update intervals which achieves the maximum lifetime: e.g., $\Delta t = 2, 8, 12, 44$ for $n = 20, 40, 60, 100$, respectively. The corresponding total number of updates over the lifetime is 46, 23, 19, 7 for different values of network size.
- Because the required total number of updates to get maximum lifetime is relatively small, the MINMAX optimization criteria such as WMST and WMSTSW produces better results than WBIP and WBIPSW, which is loosely related to the MINTOTAL optimization criteria.

8. Conclusions

We addressed the problem of maximizing the network lifetime of a single broadcast session over wireless stationary ad hoc networks. We noticed that the prior related research is overly biased to minimizing the total transmit power. Based on this observation, among min-max and min-total strategies, we investigated which optimization criterion generally provides better network lifetime performance. For that purpose, we first categorized networks as static or dynamic networks, dependent upon whether or not routing structure is self-configurable, then separately solved the corresponding problems.

For a static network, we formulated the lifetime maximization problem as a min-max optimization problem and solved using graph-theoretic approaches. An optimal polynomial-time heuristic algorithm based on the minimum spanning tree (MST) was presented. For a dynamic network, we extended the solution obtained from the static network case and developed several cost metrics and heuristics that lead to prolonged network lifetime. Even in this case, we found the min-max strategy provides comparable to or better results than the min-total strategy in terms of network lifetime. We also analyzed the impact of various parameters such as control overhead and update interval on network lifetime and compared that with an upper bound to network lifetime.

Acknowledgment

We would like to thank the anonymous reviwers for their helpful comments.

References

[1] Special issue on challenges in mobile ad hoc networking, IEEE Communications Magazine 39(6) (2001).

[2] Special issue on energy-aware ad hoc wireless networks, IEEE Wireless Communications 9(4) (2002).

[3] Special issue on wireless ad hoc networks, IEEE Journal on Selected Areas in Communications 17(8) (1999).

[4] Special issue on collaborative signal and information processing in microsensor networks, IEEE Signal Processing Magazine 19(2) (2002).

[5] J.E. Wieselthier, G.D. Nguyen and A. Ephremides, On the construction of energy-efficient broadcast and multicast trees in wireless networks, in: *Proc. IEEE INFOCOM* (2000) pp. 586–594.

[6] A. Ephremides, Energy concerns in wireless networks, IEEE Wireless Communications 9(4) (2002) pp. 48–59.

[7] S. Singh, M. Woo and C.S. Raghavendra, Power-aware routing in mobile ad hoc networks, in: *Proc. ACM/IEEE MOBICOM*, (Dallas, TX, Oct. 1998) pp. 181–190.

[8] S.-J. Lee, W. Su, J. Hsu, M. Gerla and R. Bagrodia, A performance comparison study of ad hoc wireless multicast protocols, in: *Proc. of the IEEE Conference on Computer Communications (INFOCOM 2000)* (Tel Aviv, Israel, March 2000) pp. 565–574.

[9] C.-K. Toh, Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks, IEEE Communications Magazine (2001) pp. 138–147.

[10] J.H. Chang and L. Tassiulas, Energy conserving routing in wireless ad-hoc networks, in: *INFOCOM '2000* (2000).

[11] V. Rodoplu and T.H. Meng, Minimum energy mobile wireless networks, IEEE J. Selected Areas in Communications 17(8) (1999).

[12] J.E. Wieselthier, G.D. Nguyen and A. Ephremides, Resource management in energy-limited bandwidth-limited, transceiver-limited wireless networks for session-based multicasting, Computer Networks 39(2) (2002) pp. 113–131.

[13] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, Next century challenges: Scalable coordination in sensor networks, in: *Proc. of the ACM/ IEEE International Conference on Mobile Computing and Networking* (Seattle, Washington, USA, August 1999) pp. 263–270.

[14] 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 5(1) (1983) pp. 66–77.

[15] T.H. Cormen, C.E. Leiserson and R.L. Rivest, *Introduction to Algorithms*, 2nd edition (MIT Press, Cambridge, MA, 1990).

[16] R. Ramanathan and R. Rosales-Hain, Topology control of multihop wireless networks using transmit power adjustment, in: *Proc. IEEE INFOCOM 2000* (March 2000).

[17] R. Wattenhofer, L. Li, P. Bahl and Y. Wang, Distributed topology control for power efficient operation in multihop wireless ad hoc networks, in: *Proc. IEEE INFOCOM 2001*(2001).

[18] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, SPAN: An energy-efficient coordination algorithms for topology maintenance in ad hoc wireless networks, in: *Proc. 7th ACM MOBICOM* (Rome, Italy, July 2001).

[19] Q. Li, J. Aslam and D. Rus, Online power-aware routing in wireless ad-hoc network, in: *Proc. ACM/IEEE MOBICOM, 2001* (Rome, Italy, July 2001) pp. 97–107.

[20] Y. Xu, J. Heidemann and D. Estrin, Geography-informed energy conservation for ad hoc routing, ACM SIGMOBILE 7/01 Rome, Italy.

[21] P.J. Wan, G. Calinescu, X.Y. Li and O. Frieder, Minimum-energy broadcast routing in static ad hoc wireless networks, in: *Proc. IEEE INFOCOM* (2001).

[22] B.N. Levine and J.J. Garcia-Luna-Aceves, A comparison of known classes of reliable multicast protocols, in: *Proc. International Conference on Network Protocols (ICNP-96)* (Columbus, Ohio, Oct. 29–Nov 1, 1996).

[23] C. Ho, K. Obraczka, G. Tsudik and K. Viswanath, Flooding for reliable multicast in multi-hop ad hoc networks, in: *ACM Workshop on Discrete Algorithms for Mobility* (Aug. 1999).

[24] W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: *Proc. Hawaii International Conference on System Sciences* (Maui, Hawaii, 2000).

[25] A.E.F. Clementi, P. Crescenzi, P. Penna, G. Rossi and P. Vocca, A worst-case analysis of an MST-based heuristic to construct energy-efficient broadcast trees in wireless networks, Technical Report 010 of the Univ. of Rome "Tor Vergata " (2001).

[26] O. Egecioglu and T. Gonzalez, Minimum-energy broadcast in simple graphs with limited node power, in: *Proc. IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2001)*, Anaheim, CA (Aug. 2001) pp. 334–338.

[27] P.A. Humblet, A distributed algorithm for minimum weight directed spanning trees, IEEE Trans. on Communications COM-31(6) (1983).

[28] Y.J. Chu and T.H. Liu, On the shortest arborescence of a directed graph, Sci. Sinica 14(1965) pp. 1396–1400.

[29] R.E. Tarjan, Finding optimum branchings, Networks 7(1977) pp. 25–35.

[30] V.Tsiatsis, S. Zimbeck and M. Srivastava, Architectural strategies for energy efficient packet forwarding in wireless sensor networks, in: *Proc. ISLPED* (2001) pp. 92–95.

[31] M. Adamou and S. Sarkar, A framework for optimal battery management for wireless nodes, in: *Proc. of INFOCOM* (2002) pp. 1783–1792.

[32] C.F. Chiasserini and R.R. Rao, Energy efficient battery management in: *Proc. of INFOCOM 2000* (Tel Aviv, Israel, March 2000).

[33] V. Raghunathan, C. Schurgers, S. Park and M.B. Srivastava, Energy-aware wireless microsensor networks, IEEE Signal Processing Magazine 19(2) (2002) pp. 40–50.

[34] M. Cagalj, Jean-Pierre Hubaux and C. Enz, Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues, in: *MOBICOM'02* (Atlanta, Georgia, USA, Sept. 2002).

[35] I. Kang and R. Poovendran, Maximizing static network lifetime of wireless broadcast adhoc networks, in: *IEEE International Conference on Communications (ICC) 2003* (Anchorage, Alaska, May 11–15, 2003).

[36] I. Kang and R. Poovendran, Broadcast with heterogeneous node capability accepted to *IEEE Global Telecommunications Conference (Globecom)* (Dallas, Texas, Nov. 29–Dec. 03, 2004).

[37] I. Kang and R. Poovendran, On the lifetime extension of energy-constrained multihop broadcast networks, in: *IEEE International Joint Conference on Neural Networks (IJCNN) 2002* (Hawaii, USA, May 2002).

[38] D.M. Blough and P. Santi, Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks, in: *Proc. ACM MobiCom 2002* Atlanta pp. 183–192.

[39] P. Gupta and P.R. Kumar, Critical power for asymptotic connectivity in wireless networks, in: *Stochastic Analysis, Control, Optimization and Applications* (Birkhauser, Boston, 1998) pp. 547–566.

[40] R. Weinstock, *Calculus of Variations: With Applications to Physics and Engineering*, (Dover Publications, 1980).

[41] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows: Theory, Algorithms and Applications*, (Prentice Hall, Upper Saddle River, NJ, 1993).

[42] N. Bulusu, J. Heidemann and D. Estrin, GPS-less low cost outdoor localization for very small devices, IEEE Personal Communications Magazine 7(5) (2000) pp. 28–34.

[43] B. Bui-Xuan, A. Ferreira and A. Jarry, Computing shortest, fastest, and foremost journeys in dynamic networks, International Journal of Foundations of Computer Science 14(2 2003), pp. 267–285.

Intae Kang received his B.S. degree in physics from Seoul National University, Seoul, Korea and M.S. degree in electrical engineering from the Johns Hopkins University, Baltimore, MD. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering at the University of Washington, Seattle, WA.

His current research interests are in the area of ad hoc and sensor networks. In particular, he is interested in energy efficient routing, topology control, medium access control, mobility management, and modeling and performance analysis of network protocols using directional/smart antennas. E-mail: kangit@u.washington.edu

Radha Poovendran has been an assistant professor at the Electrical Engineering Department of the University of Washington at Seattle since September 2000. He received his Ph.D. in Electrical Engineering from the University of Maryland, College Park in 1999. His research interests are in the areas of applied cryptography for multiuser environment, wireless networking, and applications of Information Theory to security. He is a recipient of Faculty Early Career Award from the National Science Foundation (2001), Young Investigator Award from the Army Research Office (2002), Young Investigator Award from the Office of Naval Research (2004), and the 2004 Presidential Early Career Award for Scientists and Engineers, for his research contributions in the areas of wired and wireless multiuser security. He is also a co-recipient of the 2002 Outstanding Teaching as well as the Outstanding Advisor Awards from the Department of Electrical Engineering of the University of Washington.
E-mail: radha@ee.washington.edu