Multicast Communications in *Ad Hoc* Networks Using Directional Antennas: A Lifetime-Centric Approach

Y. Thomas Hou, *Senior Member, IEEE*, Yi Shi, *Student Member, IEEE*, Hanif D. Sherali, and Jeffrey E. Wieselthier, *Fellow, IEEE*

***Abstract*—We consider a wireless *ad hoc* network where each node employs a single-beam directional antenna and is provisioned with limited energy. We are interested in an online routing algorithm for successive multicast communication requests with the aim of maximizing the network lifetime. The beam-forming property, which is associated with single-beam directional antennas, introduces some unique problems that do not exist for omnidirectional antennas and, therefore, significantly increases the design space for routing algorithms. The contributions of this paper are twofold. First, we provide some important theoretical understanding on various multicast problems and deduce that even an offline version of this problem is NP-hard. Second, we develop a highly competitive online routing algorithm that takes the network lifetime consideration directly into iterative calculations and show that an algorithm that is designed under this methodology provides consistently better performance than the current state-of-the-art algorithm that only considers remaining energy. The theoretical results and routing algorithm in this paper offer some important insights on algorithm design for energy-constrained wireless *ad hoc* networks with directional antennas.**

***Index Terms*—*Ad hoc* networks, directional antenna, energy constraint, multicast, network lifetime, online algorithm, optimization, wireless communications.**

I. Introduction

IN RECENT years, there has been a tremendous interest in energy efficiency and lifetime problems associated with wireless *ad hoc* networks. For an *ad hoc* network where each node is provisioned with limited energy (also called an energy-constrained network), it is now well understood that an energy efficient routing usually cannot provide the best result for network lifetime performance [26]. This important result has led to the evolution of two lines of research: One focuses on minimizing the total energy required to maintain a tree (broadcast/multicast) [2], [4], [5], [13], [24], [25], and the other focuses on how to perform routing so that the network lifetime can be prolonged as much as possible [14], [26].

In parallel to algorithmic and protocol research in energy-constrained *ad hoc* networks, recent use of directional antennas in wireless communication has further enabled new approaches to energy saving for energy-constrained wireless networks. Indeed, the use of directional antennas allows a concentration of the beam toward the intended destination without wasting energy in unwanted directions. Further, because the beam is generated only toward a certain direction, it creates less interference to other nodes that are outside the beam, which enables greater information capacity in the network. Finally, since nodes outside the beam coverage cannot receive the source's signal, security concerns that are associated with omnidirectional broadcast can be somewhat alleviated. As a result, it is expected that the use of directional antennas has a great potential in wireless *ad hoc* networks. From a theoretical perspective, the use of directional antennas has also introduced some unique difficulties in algorithm design, particularly when each node is assumed to generate a single directional beam.[1] This is because a single directional beam provides partial broadcast to those nodes that are within the beam coverage. Unlike the case of omnidirectional antennas, where the design space depends solely on the radius (i.e., communication range), the algorithm design space for directional antennas now encompasses three components: beam radius, beamwidth, and beam orientation. Thus, a directional antenna-based routing problem needs to address the assignment of these three parameters on each node in the network.

In this paper, we consider the important problem of multicast routing with the objective of maximizing the network lifetime for energy-constrained wireless *ad hoc* networks employing directional antennas. The significance of this problem lies in that not only is it a general problem that encompasses unicast or broadcast, but it is also a generalized problem for omnidirectional antennas, which can be considered as a special case of directional antenna with 360° beamwidth. Therefore, advances along this investigation will yield significant intellectual

Manuscript received January 16, 2006; revised April 17, 2006. The work of Y. T. Hou and Y. Shi was supported in part by the National Science Foundation (NSF) under Grant CNS-0347390 and in part by the Office of Naval Research (ONR) under Grant N00014-03-1-0521. The work of H. D. Sherali was supported in part by the NSF under Grant DMI-0552676. The work of J. E. Wieselthier was supported in part by the ONR. The review of this paper was coordinated by Prof. M. Guizani.

Y. T. Hou and Y. Shi are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: thou@vt.edu; yshi@vt.edu).

H. D. Sherali is with the Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: hanifs@vt.edu).

J. E. Wieselthier is with the Information Technology Division, Naval Research Laboratory, Washington, DC 20375 USA (e-mail: wieselthier@itd.nrl.navy.mil).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2007.895478

[1]Although multiple beams can be formed by directional antenna arrays, the hardware complexity and energy consumption are much higher than those for single-beam directional antenna [22].

merits. We assume that the directional antenna at each node can only form a single beam where the beam radius, beamwidth, and beam orientation are adjustable. Instead of looking for an optimal routing solution for a single multicast session, we are interested in an online algorithm for the problem, where multicast requests arrive over time and there is no knowledge of the future request arrival pattern.

The contributions of this paper include both theoretical understanding and algorithm design for the multicast routing problem. From the theoretical perspective, we show that an offline version of this multicast routing problem is NP-hard. By "offline," we mean that we have complete knowledge of multicast requests over time. This result builds upon several intermediate results, each of which has its own significance and offers important understanding on closely related problems. In an important intermediate result (Theorem 1), we show that for the directional antenna case, the static maximum-lifetime tree problem for a single multicast (or broadcast) session is NP-complete. The proof of this result gives insights on how a directional antenna can increase the computational complexity in algorithm design.

Since even the offline multicast routing problem is NP-hard, for an online algorithm, only a heuristic approach is feasible. In the second half of this paper, we aim to develop an online multicast routing algorithm to maximize the network lifetime. In [26], Wieselthier *et al.* made a major step in the systematic study of the online multicast routing problem. In particular, they designed the directional multicast incremental power (D-MIP) algorithm that incorporates nodal residual energy into the local cost metric for routing. Although this algorithm offers good performance, there is a very subtle detail in the algorithm design of D-MIP that motivates us to further investigate this important problem. Nodal residual energy is indeed closely related to node lifetime (and, thus, network lifetime), but it may be more important to take lifetime consideration directly into the algorithm design. Consequently, we make the following conjecture in our investigation: if we incorporate lifetime consideration explicitly into the design of an online multicast routing algorithm, we should expect to have an algorithm that outperforms the D-MIP algorithm. To prove this conjecture, we design a new algorithm called Maximum Lifetime Routing for Multicast with Directional antennas (MLR-MD). The design experience for MLR-MD is quite interesting and offers understanding on beam-forming behavior under single-beam directional antennas, particularly the relationship between physical one-hop neighbor and logical one-hop neighbor concepts. Through simulation results, we conclusively demonstrate that the MLR-MD offers consistent performance improvement over the D-MIP algorithm, which confirms our initial conjecture.

The remainder of this paper is organized as follows. In Section II, we describe the network system model and state the online multicast routing problem under investigation. Section III provides a detailed discussion on theoretical aspects of the multicast routing via closely related problems, thereby setting up the theoretical background for the problem in this paper. In Section IV, we design a lifetime-centric online algorithm for the multicast routing problem. In Section V, we use simulation results to demonstrate the efficacy of the proposed algorithm. Section VII concludes this paper.

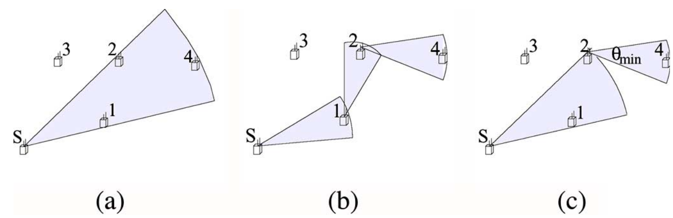

Fig. 1. Three different multicast routing solutions for the same multicast session.

II. System Model and Problem Description

A. System Model

We consider a wireless *ad hoc* network consisting of N nodes located over a region. For wireless communication, we assume that each node is equipped with a directional antenna for transmission and an omnidirectional antenna for reception.[2] Similar to [26], we assume that each node's transmitter has a power control capability. That is, by adjusting the transmission power level, the sender could reach destination nodes located at different distances. Further, we assume that each node could control the beamwidth and beam orientation of its directional antenna [21]. Then, a node's transmission coverage area can be effectively controlled by adjusting the power level, beamwidth, and beam orientation of the directional antenna. Fig. 1(a) illustrates this concept, wherein a sending node S could transmit to nodes 1, 2, and 4 simultaneously by controlling the power level, beamwidth, and beam orientation at node S without causing an interference at node 3.

Depending on the specific wireless environment and a node's hardware and software implementation, each node's energy consumption behavior may be different. In our theoretical development (Section III) and algorithm design (Section IV), we model the transmission energy at a node u as a function of ρ, θ, and ω, where ω is the beam orientation, ρ is the reachable distance along this orientation, and θ is the beamwidth. Denote $p_u^{\mathrm{T}}(\rho, \theta, \omega)$ as the beam transmission cost function, which is node dependent. Without loss of generality, we assume that, in wireless communication environment, this function is an increasing function of ρ and θ, i.e.,

$$p_u^{\mathrm{T}}(\rho_1, \theta, \omega) < p_u^{\mathrm{T}}(\rho_2, \theta, \omega), \qquad \text{if } \rho_1 < \rho_2 \tag{1}$$

$$p_u^{\mathrm{T}}(\rho, \theta_1, \omega) < p_u^{\mathrm{T}}(\rho, \theta_2, \omega), \qquad \text{if } \theta_1 < \theta_2. \tag{2}$$

Further, to better model the wireless environment in practice, we do not assume a uniform path loss in all directions (ω). Instead, we let $p_u^{\mathrm{T}}(\rho, \theta, \omega)$ not only depend on ρ and θ but also be a function of beam orientation ω. Therefore, it is possible that $p_u^{\mathrm{T}}(\rho, \theta, \omega_1) \neq p_u^{\mathrm{T}}(\rho, \theta, \omega_2)$ if $\omega_1 \neq \omega_2$. Due to the nonuniform path loss along different directions, the beam coverage may not be a uniform sector; although for ease of

[2]It is possible to use a directional antenna for reception as well, although its energy saving may not be as significant as that for transmission, particularly for large-sized networks.

illustration, we use a uniform sector (e.g., in Fig. 1) to represent the coverage of a directional beam in all figures.

Since energy is also consumed for other nodal processing functions and reception, for each node u, we define p_u^{P} as the transmission processing energy and p_u^{R} as the reception energy for each unit data. Then, the total energy consumed at a node u for one unit data with beam (ρ, θ, ω) is

$$C_u(\rho, \theta, \omega) = p_u^{\mathrm{T}}(\rho, \theta, \omega) + p_u^{\mathrm{P}} + p_u^{\mathrm{R}}. \quad (3)$$

Note that depending on the role of node u (i.e., sender, receiver, or both), the term $p_u^{\mathrm{T}}(\rho, \theta, \omega)$ or p_u^{R} may not exist.

B. Multicast Problem

In the most general form, any node in the network may need to transmit to a subset of all other nodes in the network. Clearly, this multicast communication includes both unicast and broadcast communications. A source node could generate a single beam to reach all nodes in this subset in a single hop [e.g., Fig. 1(a)]. Although simple, this approach is not energy efficient, particularly for large-sized networks, due to the power consumption behavior in (1) and (2), i.e., the energy consumption increases when the distance ρ and beamwidth θ increase. As a result, it is important to explore a multihop relaying approach to extend the network lifetime [e.g., Fig. 1(b) and (c)].

There are various definitions for network lifetime [3]. For simplicity, we define the network lifetime as the time instance when the network can no longer support a multicast communication session. This happens when either the source node or any multicast receiving node runs out of energy during a multicast communication session. Clearly, the idle periods where there are no multicast sessions in the network should not be considered as part of the network lifetime since there is no energy expenditure during these periods. That is, the network lifetime under consideration only consists of the time intervals where there are active multicast communication sessions in the network. In the simple case where there is no time overlap between consecutive multicast communication sessions in the network, the accounting for network lifetime is the sum of successive time intervals for multicast communication sessions. In the case where there are multiple concurrent multicast communication sessions in the network, special care must be taken in the accounting of network lifetime. We will further elaborate this point in Section V.

We use an example to illustrate the multicast communication problem at a particular time instance. In Fig. 1, suppose that the multicast communication request arrives at node S (source node) and wishes to transmit to nodes 1, 2, and 4. Depending on the relay topology, there are various transmission behaviors that can be employed. For example, in Fig. 1(a), node S transmits to nodes 1, 2, and 4 directly; in Fig. 1(b), node S transmits to node 1, node 1 transmits to node 2, and node 2 transmits to node 4; and in Fig. 1(c), node S transmits to nodes 1 and 2, and node 2 transmits to node 4. Clearly, the topology and energy consumption behavior for each case are different, leading to different network lifetime performance. Note that in practice, there is a minimum beamwidth requirement $\theta_{\min}$ for a beam, even in the case where the transmitting node may have only one downstream neighbor.[3] Further, we assume that there is a maximum beamwidth requirement $\theta_{\max}$ for a beam.

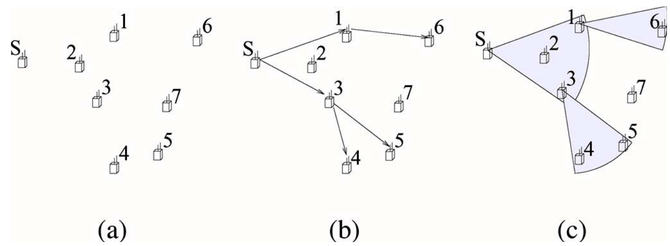

Fig. 2. Logical multicast routing tree and a physical beam forming behavior for a multicast session. (a) Network topology, (b) logical routing tree, and (c) beam forming pattern.

An important concept in designing routing algorithms for wireless networks is the distinction between physical one-hop neighbor and logical one-hop neighbor [14]. To illustrate these two concepts in the context of multicast routing with directional antennas, we use the example in Fig. 2. In Fig. 2(a), we have an *ad hoc* network, and a multicast request is initiated by node S, with multicast destination nodes being 4, 5 and 6. Fig. 2(b) shows a particular logical multicast routing topology that can support this multicast communication session, where nodes 1 and 3 are used as relay nodes in the multicast tree. Fig. 2(c) shows the corresponding physical beam-forming behavior at each node of the multicast tree. Note that in this example, a single beam from the source node S can cover not only nodes 1 and 3 but node 2 as well. Although node 2 is not a logical one-hop neighbor to node S on the multicast tree, it is a physical one-hop neighbor to node S. An important application of physical one-hop neighbors is that, should it become necessary to reconfigure a new multicast tree, these physical one-hop neighbor nodes can be added (attached) to the logical multicast tree without changing the current beam forming at any node.

In practice, multicast communication requests arrive at different nodes in the network over time, and the corresponding multicast groups (destination nodes) also change. For a given source node, the multicast group can change from request to request. For each request, there is an amount of data (also varies from multicast session to multicast session) that needs to be sent to the respective multicast group. Our objective is to pursue an optimal transmission behavior (assignment of beam radius, beamwidth, and beam orientation at each node) so that the network lifetime is maximized.

III. Theoretical Understanding of the Multicast Routing Problem

In this section, we explore some theoretical understanding of the multicast routing problem with energy constraint. Our investigation builds upon several intermediate results, each of which summarizes the level of computational complexity of some closely related problems.

[3]Typically, the smaller the minimum beamwidth requirement, the more complex and costly the directional antenna [22].

A. Minimum Power Routing (MPR)

In [6], Das *et al.* studied a single-session minimum-power broadcast tree problem with omnidirectional antennas. Although the authors proposed three mixed integer linear programming (MILP) formulations, no explicit solutions were given. It is well known that an MILP problem is NP-hard in general [10]. In [11], Guo and Yang studied the single-session minimum-power multicast tree problem in the context of directional antennas (with fixed beamwidth). They also formulated the problem into an MILP. Again, there is no explicit analytic solution (due to NP-hardness). Although there is software available to solve MILP problems, a solution is obtainable only for small-sized problems.

It is important to realize that the minimum-power broadcast tree problem cannot be translated into a spanning-tree problem, which can be solved in polynomial time. The spanning-tree problem addresses a connected graph with predefined edges and associated costs. The objective is to select edges with the minimum total cost that connects all vertices in the graph, where the total cost is the sum of costs of selected edges, but in wireless networks that are broadcast in nature (or partially broadcast in the case of directional antennas), the total cost at a node is not a simple summation of the cost of its outgoing links.

In [4], Cagalj *et al.* proved that for omnidirectional antennas, the minimum-power broadcast tree problem is NP-complete. Since broadcast is a special case of multicast and the omnidirectional antenna is a special case of the directional antenna, we conclude that the minimum-power multicast tree problem is NP-hard under either omnidirectional or directional antennas, which we summarize in the following lemma.

Lemma 1: For either directional or omnidirectional antenna, the problem of finding a static minimum-power multicast tree is NP-hard.

B. Maximum Lifetime Routing

From the lifetime performance perspective, it has been recognized that the minimum-power routing usually cannot provide good network lifetime performance. Consequently, there have been recent efforts on exploring multicast routing, with an objective of maximizing the network lifetime, for energy-constrained *ad hoc* networks. The problems along this line of research can be classified into three problems:

- *Problem 1)* maximizing the lifetime of a single static multicast tree;
- *Problem 2)* maximizing the lifetime of a single multicast tree with dynamic topology updates;
- *Problem 3)* maximizing the lifetime for a sequence of requests, each of which will generate a multicast tree, with dynamic topology updates for each multicast tree.

The third problem is the focus of this paper. Note that the first and second problems can be considered special cases under the third problem. We now discuss the theoretical aspect of each problem as follows.

Problem 1: The first problem addresses the network lifetime of a single static multicast tree (without dynamic topology updates). There are polynomial time algorithms [7], [14] to solve this problem for the broadcast case with omnidirectional antennas. In [9], Floreen *et al.* proved that this problem can be solved in polynomial time for omnidirectional antennas. All these prior results are obtained under the assumption that omnidirectional antennas have uniform path loss behavior in all directions (i.e., a node's coverage is a disc). We now extend the proof in [9] for the general case where path loss may be nonuniform (see discussion in Section II-A). Since broadcast is a special case of multicast, we only need to show the result for the multicast case in the following lemma.

Lemma 2: For omnidirectional antennas, the problem of finding a static maximum-lifetime tree for a single multicast (or broadcast) session can be solved in polynomial time.

Proof: Suppose we have N nodes in the network. For each node in a multicast tree, the energy consumed on p_u^{P} term (source node) or $p_u^{\mathrm{P}} + p_u^{\mathrm{R}}$ term (nonsource node) is deterministic. Now, we consider the p_u^{T} term. For the case of omnidirectional antennas, given the value of p_u^{T}, the set of covered nodes is unique. To be energy efficient, we only need to consider $O(N)$ values of p_u^{T} term at each node, which correspond to the number of power levels to cover i neighbors $(0 \leq i \leq N-1)$, where $i = 0$ represents the special case that the node does not transmit data to any node. Thus, there are $O(N)$ different total power consumption levels at each node, which correspond to $O(N)$ different node lifetimes. Since we have a total of N nodes, there exists a maximum of $O(N^2)$ different lifetime values. We only need to check which value among these $O(N^2)$ lifetime values yields a maximum feasible lifetime solution for this multicast.

We now check if a given lifetime value t is feasible. If this t is feasible, then there exists a multicast tree such that each node has a lifetime of at least t. For each node, we first assume that it is on the multicast tree and subtract the energy consumed on p_u^{P} term (for source node) or $p_u^{\mathrm{P}} + p_u^{\mathrm{R}}$ term (for nonsource node) over t. For each node that has negative remaining energy after this subtraction, we check to see if it is the source node or a destination node in the multicast session. If yes, we can declare immediately that this t is infeasible. Otherwise, this node cannot be a node in the multicast tree and is thus removed from further consideration. For the remaining nodes, we first compute the maximum transmission powers based on t and their remaining energy. Then, we can compute their transmission coverage. Based on the coverage of each node [$O(N^2)$ complexity], we can quickly determine if a multicast tree exists (e.g., via depth first search) in $O(N)$ time. The complexity for this feasibility check is therefore $O(N^2)$.

Since we have $O(N^2)$ different lifetime values, we can sort them in $O(N^2 \log N)$ complexity. Then, we use binary search [$O(\log N)$ times] to find the maximum-lifetime tree. The overall complexity, i.e., $O(N^2 \log N) + O(\log N) \cdot O(N^2)$, is $O(N^2 \log N)$. ■

For the case of directional antennas, Problem 1 becomes much harder. Its complexity is addressed in the following theorem.

Theorem 1: For directional antennas, the problem of finding a static maximum-lifetime tree for a single multicast (or broadcast) session is NP-complete.

Proof: Instead of proving that the maximum-lifetime tree problem is NP-complete, it is sufficient to prove that the lifetime feasibility problem is NP-complete. This is because if we can find the maximum lifetime t^* in polynomial time, then for any given t, the lifetime feasibility problem can be solved by comparing t and t^*. On the other hand, if we can determine the feasibility of any given t in polynomial time, with a similar analysis on the possible maximum network lifetime as that in Lemma 2, the maximum lifetime t^* can be obtained by a binary search in polynomial time.

We begin with the directed Hamilton path problem with a given starting vertex. That is, for a given directed graph $G(V, E)$, with vertex set V and directed edge set E, and a designated vertex S, we want to find if there exists a directed Hamilton path with starting vertex S to each of the other vertices in V. It is well known that the directed Hamilton path problem (with any starting vertex) is NP-complete [19]. Note that by creating a dummy starting node and directed edges from this node to all other nodes, we can show that the directed Hamilton path problem with a given starting vertex is NP-hard. It is easy to show that this problem is also in NP. Thus, the directed Hamilton path problem with a given starting vertex is NP-complete.

Now, given any instance of the directed Hamilton path problem with starting vertex S, we show how to reduce it to an instance of the lifetime feasibility problem in polynomial time. In the lifetime feasibility problem, we consider broadcast, which is a special case of multicast, where a node S needs to transmit data to all other nodes. We let $\theta_{\min} = \theta_{\max}$ and denote it as θ_{f}, i.e., the beamwidth is fixed. For the given directed graph G, denote N_u as the set of vertices that are "outgoing" neighbors of vertex u, such that for any $q \in N_u$, there is a directed edge from u to q, i.e., $u \rightarrow q$.

Now, we assign the parameter values of an N-node network for the feasibility problem. First, we arrange the N nodes such that there are no more than two nodes on the same line. As a result, we can set a value for θ_{f} such that a single beam from any node (with any ρ and ω) can cover at most one node. Further, we can arrange an energy consumption function $C_u(\rho, \theta, \omega)$ with such a property that for each node u, the energy cost to cover any node $q \in N_u$ is smaller than the energy cost to cover any node $z \notin N_u$. Now, for an arbitrarily given lifetime $t > 0$, we can always initialize the energy of each node u such that it can transmit to any node $q \in N_u$ over time t, while it is unable to transmit to any node $z \notin N_u$ for the entire duration of t. As an example of how to define $C_u(\rho, \theta, \omega)$ and set initial energy for each node u, we can let $C_u(\rho_{uq}, \theta_{\mathrm{f}}, \omega_{uq}) = a_u$ for every node $q \in N_u$, and $C_u(\rho_{uz}, \theta_{\mathrm{f}}, \omega_{uz}) = b_u$ for every node $z \notin N_u$, where a_u and b_u are constants and $b_u > a_u > 0$; ρ_{uq} and ρ_{uz} are distances from node u to q and z; ω_{uq} and ω_{uz} are beam orientations from node u to q and z, respectively. The value of $C_u(\rho, \theta, \omega)$ at other locations is not of our concern and thus can be defined arbitrarily. The only requirement that $C_u(\rho, \theta, \omega)$ should have is that it is an increasing function of ρ for any fixed ω, as we discussed in (1) (note that θ is already fixed as θ_{f} earlier). Also, as we discussed in Section II-A, due to potential nonuniform path loss along different beam orientations ω in the practical wireless environment, $C_u(\rho, \theta, \omega)$ also depends on beam orientation ω. In particular, even if $\rho_1 > \rho_2$, it is possible that $C_u(\rho_1, \theta_{\mathrm{f}}, \omega_1) < C_u(\rho_2, \theta_{\mathrm{f}}, \omega_2)$ if $\omega_1 \neq \omega_2$. Now, we can set the initial energy at each node u in the network to be $t \cdot r \cdot a_u$, where r is the transmission data rate. Thus, node u can transmit to any node $q \in N_u$ over time t, while it is unable to transmit to any node $z \notin N_u$ for the entire duration of t since $a_u < b_u$. This completes the example.

Under the above setting, it follows that the lifetime t is feasible if and only if G has a directed Hamilton path with starting vertex S to all other vertices. Therefore, any instance of the directed Hamilton path problem with starting vertex S can be reduced to an instance of the lifetime feasibility problem. Thus, the lifetime feasibility problem is NP-hard. It is easy to show that the lifetime feasibility problem is in NP. Thus, the lifetime feasibility problem is NP-complete. As a result, for directional antenna case, our static maximum-lifetime tree problem for multicast is also NP-complete. This completes the proof. ■

Problem 2: We now move on to the discussion of the second problem, which addresses how to maximize the lifetime of one multicast tree under dynamic routing (i.e., routing topology may change over time for this single multicast). For omnidirectional antennas, Floreen *et al.* [9] claimed that this problem is NP-hard. Based on this claim, since the directional antenna is a general case of the omnidirectional antenna, the problem of maximizing the lifetime for one multicast tree under dynamic routing is also NP-hard.

Problem 3: The third lifetime problem addresses how to perform multicast routing when successive multicast requests arrive to the network. This problem is substantially more difficult than the multicast routing for a single request (e.g., [4], [6], [9], [11]) in that we are not interested in the maximum-lifetime tree for one request, but rather, we are interested in the network lifetime performance when successive multicast session requests (generated at different nodes in the network and with different multicast groups) arrive and depart over time. That is, we are looking for an "online" algorithm without any knowledge of future request arrivals with the aim of maximizing network lifetime.[4] This is in contrast to the "offline" optimization for maximum network lifetime problem, which assumes that the future multicast requests are known *a priori*. In [17], Li *et al.* proposed an online routing algorithm to maximize the network lifetime for unicast case. They showed that an online algorithm does not have a constant competitive ratio[5] to the offline optimum. Since unicast is a special case of multicast, we conclude that online algorithms for multicast routing do not have a constant competitive ratio.

As the online optimization cannot be solved analytically, one might ask whether it is possible to pursue an "offline" optimization algorithm. By "offline," we mean that we first record the successive multicast request arrivals to the network over time. Then, assuming that we can "go back" in time with the knowledge of all these future arrivals, we attempt to pursue

[4]Recall that network lifetime is defined as the first time instance when a multicast communication fails, either due to energy depletion at the sender or any receiver of the multicast group.

[5]The competitive ratio of an online algorithm is the ratio between the performance of this online algorithm and an optimal offline algorithm.

routing optimally for each successive request such that the network lifetime is maximized. Note that in the extreme case, when all multicast requests have the same source node and destination nodes, this problem reduces to Problem 2 that we discussed earlier, which is NP-hard. Therefore, we conclude that the offline multicast routing for successive multicast requests is also NP-hard.

Theorem 2: For both omnidirectional and directional antennas, an offline problem of optimal routing to maximize the network lifetime for a wireless *ad hoc* network with successive multicast requests is NP-hard.

Since the offline problem for multicast routing is NP-hard, for an online problem, only the heuristic approach is feasible. This will be our effort in the next section.

IV. Lifetime-Centric Design for Online Multicast Routing Algorithm

A. Background and Motivation

In [26], Wieselthier *et al.* made a major step in the systematic study of online multicast routing for energy-constrained *ad hoc* networks. In particular, they examined the source-initiated session-based multicast problem for successive requests and proposed an online heuristic algorithm (D-MIP) that was shown to have good performance in terms of network lifetime and traffic volume. In particular, in the design of D-MIP, the authors explicitly incorporated nodal residual energy into the local routing cost metric. Then, they used a spanning-tree-like technique to obtain a broadcast tree, which they called broadcast incremental power (BIP) algorithm. A multicast tree can be obtained by pruning the unnecessary links. The algorithm for directional antenna case was called D-MIP algorithm.

Although the D-MIP algorithm is currently the state-of-the-art online algorithm to the multicast routing problem, there is a very subtle detail in its design that motivates us to further investigate this important problem. Specifically, although nodal residual energy indeed is closely related to node lifetime (and, thus, network lifetime), it still may not be as effective as if we take the network lifetime metric directly into iterative calculations. Consequently, we make the following conjecture in our investigation. If we incorporate lifetime consideration directly into the iterative calculation of the online multicast routing algorithm, we should have an algorithm that outperforms an algorithm based on the nodal residual energy (e.g., D-MIP). In this section, we develop an online multicast routing algorithm along this approach, and in Section V, we use simulation results to demonstrate the performance improvement. We name our algorithm MLR-MD, which is intended to contrast with the traditional MPR [24], [25] or variants of minimum cost routing [26]. It is worth pointing out that problems either addressing broadcast or considering omnidirectional antennas can be considered as special cases under multicast or directional antennas (with $\theta = 360$), respectively.

B. Basic Idea

For a given multicast request, the basic idea of the MLR-MD algorithm is to start with a multicast routing solution first (e.g., a single beam from the source covering all multicast destination nodes) and then iteratively improve the lifetime performance of the current solution by identifying the node with the smallest lifetime[6] and revising the routing topology, as well as corresponding beam-forming behavior for an increased network lifetime. For directional antennas with power control capability, a node's lifetime can be increased via two techniques: narrowing beamwidth θ and reducing beam radius ρ. A direct consequence of such operation is that some nodes in the multicast tree that are covered by the original beam could be exposed (uncovered) under the new beam with reduced beamwidth or beam radius. The MLR-MD algorithm has several approaches to "re-attach" these exposed nodes back onto the multicast tree. Since a reattachment operation would decrease some other node's lifetime, a decision must be taken on whether a reattachment operation is feasible. Naturally, a reattachment operation is feasible only if the new network lifetime is increased. For the next iteration, we repeat the same process, i.e., identifying the node among all the nodes in the network with the minimum lifetime and attempting to revise the routing topology and beam-forming behavior to increase the network lifetime.

When nothing can be done to further improve this minimum lifetime, we move on to consider the node with the second smallest lifetime and attempt to increase its lifetime, under the condition that the lifetime for the first node (with minimum lifetime) will not decrease. In particular, MLR-MD does not increase the lifetimes of downstream nodes of the first node. The motivation for attempting to reconfigure the node with the second smallest lifetime is the following. Although the increase of this second smallest node lifetime may not increase the minimum node lifetime, it will enable the multicast routing topology to evolve to a better structure, thereby creating a new optimization space for the first node (with minimum node lifetime) in the next iteration.

If nothing can be done for the node with the second smallest node lifetime, the MLR-MD algorithm will continue to try the node with the third smallest node lifetime, and so forth. The algorithm terminates after it has tried all the nodes (on the order of nondecreasing node lifetime) and cannot increase the lifetime of any of the nodes. The pseudocode of this basic idea is shown in Figs. 3 and 4, which are further elaborated upon as follows.

C. Algorithm Details

We now consider some details in the MLR-MD algorithm. As described earlier, upon identifying a minimum-lifetime node at an iteration, we will attempt to reduce either its beamwidth or beam radius in order to increase its lifetime. The immediate consequence of this operation is that some nodes along the border of the original beam are being pushed out of the new beam's coverage. We use an example to illustrate this point. Fig. 5(a) shows the logical one-hop links on node 1, while Fig. 5(b) shows the beam-forming behavior on node 1. Suppose we wish to extend node 1's lifetime by reducing either its

[6]Node lifetime is calculated by assuming that the last routing topology holds forever.

```
Source node generates one beam to cover all destination nodes;
Sort all nodes in non-decreasing lifetime order and arrange the sorted list
with a stack L (with the top node having the smallest lifetime);
set Removed[i][j] = 0 for 1 ≤ i, j ≤ N;
while (L != ∅){
    i = pop(L);
    Identify the logical downstream one-hop neighbors of node i that are on
    the border on node i's beam;
    Sort such nodes in non-increasing improvement order and arrange the
    sorted list with a stack L_i (with the top node contributing to the largest
    lifetime increase on node i if removed);
    Improved = 0;
    while (L_i != ∅){
        j = pop(L_i);
        if (RemoveLink(i, j)==1){
            Removed[i][j] = 1;
            Improved = 1;
            break;}}
    if network lifetime increases, set Removed[i][j] = 0 for 1 ≤ i, j ≤ N;
    if (Improved==1)
        Sort all nodes in non-decreasing lifetime order and arrange the
        sorted list with a stack L (with the top node having the smallest
        lifetime);}
```

Fig. 3. Main program.

```
int RemoveLink(int i, int j){
    Remove link i → j;
    if (Case1B(i, j)==1) return 1;
    if (Case1A(i, j)==1) return 1;
    if (Case2A(i, j)==1) return 1;
    if (Case2B(i, j)==1) return 1;
    Add link i → j back, recover node i's beam;
    return 0;}
int Case1A(int i, int j){
    Identify a node v (v ≠ i) so that
    1) Removed[v][j] = 0;
    2) v is in the multicast tree or ∃k, Removed[k][v] = 0, v is covered
    by node k's beam, and k's lifetime>node i' old lifetime (in this case,
    add link k → v); and
    3) after adding link v → j, v's new lifetime> i's old lifetime;
    if such node exists, then choose the best node (having the largest new
    lifetime) and return 1; else return 0;}
int Case1B(int i, int j){
    Identify a node v in the multicast tree (v ≠ i) so that Removed[v][j]
    = 0, j is covered by node v's beam, and v's lifetime>node i's
    old lifetime;
    if such node exists, then choose any node and return 1; else return 0;}
int Case2A(int i, int j){
    Identify a pair of nodes (u, v) (u ≠ i) so that
    1) Removed[u][v] = 0 and Removed[v][j] = 0;
    2) u is in the multicast tree or ∃k, Removed[k][u] = 0, u is covered
    by node k's beam, and k's lifetime>node i's old lifetime (in this case,
    add link k → u); and
    3) v is not covered by any beam of the multicast tree; and
    4) after adding links u → v → j, the pair lifetime (the smaller new
    lifetime of nodes u and v)>node i's old lifetime;
    if such pair exists, then choose the best pair (having the largest pair
    lifetime) and return 1; else return 0;}
int Case2B(int i, int j){
    Identify a node v so that
    1) Removed[i][v] = 0 and Removed[v][j] = 0;
    2) v is not covered by any beam of the multicast tree; and
    3) after adding links i → v → j, the pair lifetime (the smaller new
    lifetime of nodes i and v)>node i's old lifetime;
    if such node exists, then choose the best node (having the largest pair
    lifetime) and return 1; else return 0;}
```

Fig. 4. Auxiliary functions.

beamwidth or beam radius. Under either technique, it is only necessary to consider the three border nodes 2, 4, and 5. In the case of beam radius reduction, we can consider to expose node 5 and let the beam cover only nodes 2, 3, and 4 [see Fig. 5(d)]. This will result in a new downstream logical topology for node 1 in Fig. 5(c), where the previous logical link between nodes 1 and 5 is removed. Since node 5 (or one node in node 3's subtree) may belong to the multicast group, it has to be reattached back to the multicast tree through another link by means of a procedure that we will describe shortly. In the case

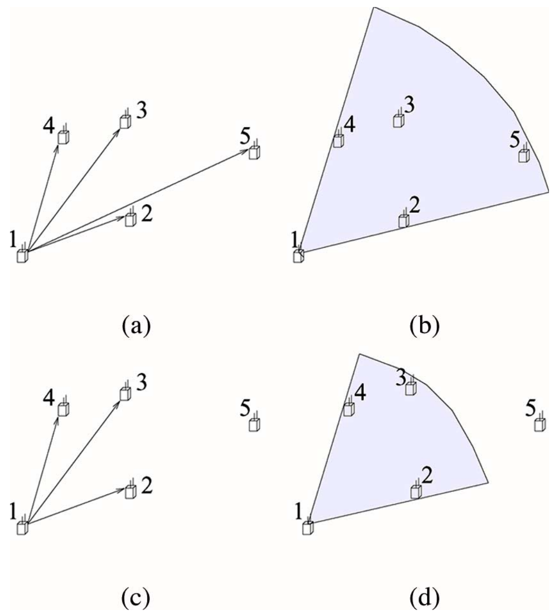

Fig. 5. Example of reducing node 1's beam coverage. (a) Logical one-hop links. (b) Beam from node. (c) New logical one-hop links. (d) New beam from node 1.

of beamwidth reduction, we can consider to remove node 2 or 4 following the same approach. As either beam radius reduction (i.e., remove coverage for node 5) or beamwidth reduction (i.e., remove coverage for node 2 or 4) will increase node 1's lifetime, a decision must be made as which node we should remove (2, 4, or 5). In our implementation (Fig. 3), we rank the order of these three possibilities (nodes 2, 4, and 5) in terms of how much improvement each will bring to node 1's lifetime. We will first try to remove the node that yields the largest increase in nodes 1's lifetime. If the reattachment of this node (node 5 in example) is feasible, then we are done. Otherwise, we declare by removing this node as a failure, and we will consider removing the node that will yield the second largest increase in node 1's lifetime, and so forth. From the perspective of logical one-hop neighbor, any of these node removal operations, either due to beamwidth reduction or due to beam radius reduction, is equivalent to breaking a logical link to one of the logical one-hop neighbors. This observation is important in coding and implementation in the sense that a link removal subroutine [RemoveLink() in Fig. 4] can be used by either the beam radius reduction operation or the beamwidth reduction operation.

We now discuss another important property associated with nodes that are not on the logical multicast tree but are within the coverage of one of the directional beams associated with the multicast tree (i.e., physical one-hop neighbor). Referring to Fig. 6, suppose node S is the source node and nodes 3, 4, 5, 6, and 8 are the multicast destination nodes. Fig. 6(a) shows a multicast tree topology for a particular routing solution, and Fig. 6(b) shows the areas that are being covered by the beams of the multicast routing solution. For those nodes that are not on this multicast tree but are within the coverage of these beams (e.g., nodes 7 and 9), we claim that there exists a path from the source of the multicast to each of them. For example, a path for node 7 is $S \rightarrow 2 \rightarrow 5 \rightarrow 7$, where logical link $5 \rightarrow 7$

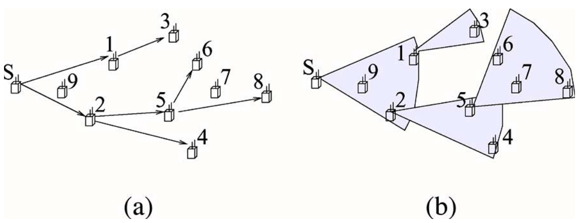

Fig. 6. Concept of logical one-hop neighbor and physical one-hop neighbor. (a) Logical multicast tree. (b) Physical beamforming behavior at each node.

can be added under node 5's current beam since node 7 is the physical one-hop neighbor of node 5. That is, if there is a need to add one of these nodes onto the multicast tree, all we need to do is to add one logical link in the multicast tree without any change to existing physical beams. We formally state this important property associated with directional antenna-based multicast routing as follows.

Property 1 (Multicast Beam Coverage): Consider a node that is not in the logical multicast tree but is a physical one-hop neighbor of a node within the multicast routing tree. This node can be attached to the logical multicast tree by adding one logical link, without any change to the existing beam-forming structure in the network.

We now discuss how the MLR-MD algorithm handles the "re-attachment" operation, i.e., the reconnection of an exposed node back to the multicast routing tree. This operation requires the reconfiguration of existing beam-forming structures in the network and can be classified into two cases: 1) without the use of intermediate relay nodes (Case I) and 2) with the use of intermediate relay nodes (Case II). An intermediate relay node is a node that is currently not within the coverage of any beam and is chosen as a relay for reattachment.

Case I—Reattachment Without Intermediate Relay Nodes: This case is best explained with an example. Suppose that we have a logical multicast tree in Fig. 7(a) with a physical beam-forming solution in Fig. 7(b), where node S is the source, and nodes 3, 4, 5, and 6 are multicast destination nodes. Now, we want to increase node 1's lifetime by pushing out node 5 from its beam. Consequently, a new beam can be formed to cover nodes 3 and 4 only, and node 5 is exposed. It is necessary to reattach node 5 back to the multicast tree. Under Case I, no intermediate relay nodes are used. We only consider to adjust the beam at one of nodes S, 2, 3, 4, or 6 to cover 5, i.e., nodes in the multicast tree or nodes covered by a node in the multicast tree, excluding node 1, and the new lifetime of the corresponding node (with a modified beam) will decrease. The reattachment operation is considered a success only if this node's new lifetime (with modified beam) is larger than node 1's lifetime before pushing out node 5. In the case when there are multiple successful reattachments, we will choose the reattachment that yields the longest node lifetime. For example, in Fig. 7(c), suppose that node 2's new lifetime is the largest among others, then MLR-MD will choose node 2 to connect node 5, with a new beam-forming shown in Fig. 7(d).

Fig. 8 shows the special case that one can take advantage of when one node (i.e., node 5) already falls within the beam coverage of another node (i.e., node 2). In this case (recall our discussion for Property 1), there is no need to generate new beams or update beams in order to reattach node 5 into the multicast tree. Instead, it is only necessary to update the logical multicast tree [see Fig. 8(c)] and mark node 5 to be a downstream node of node 2.

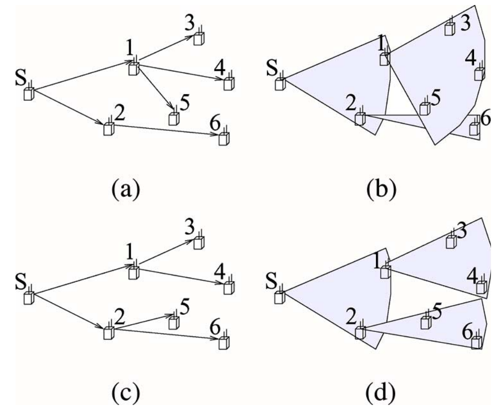

Fig. 7. Case I—Reattachment without intermediate relay nodes. (a) Multicast tree. (b) Beam on each node. (c) New Multicast tree. (d) New beam on each node.

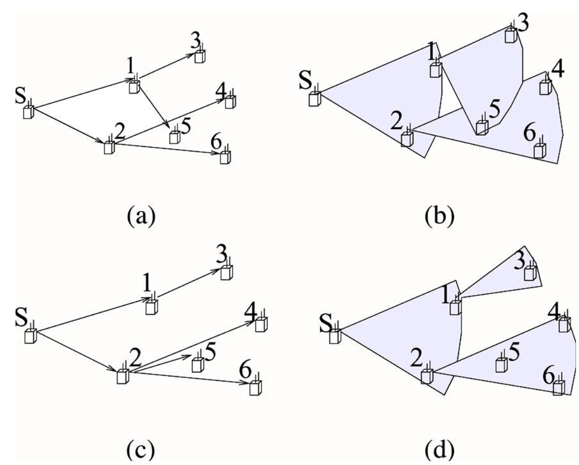

Fig. 8. Special case of Case I. (a) Multicast tree. (b) Beam on each node. (c) New Multicast tree. (d) New beam on each node.

The pseudocode of Case I is shown in Fig. 4 as Case1A(), and the special case of Case I is Case1B(). The algorithm tries Case1B() first because there is no lifetime decrease under this special case.

Case II—Reattachment With Intermediate Relay Nodes: Again, this case is best explained with an example. Suppose that we have a logical multicast tree in Fig. 9(a), where node S is the source and nodes 1, 2, 3 and 6 are multicast destination nodes. Fig. 9(b) shows a beam-forming solution of this multicast tree. Now, we want to increase node 1's lifetime by pushing out node 6 from its beam. Consequently, we regenerate a new beam from node 1 (with beamwidth $\theta_{\min}$) to just cover node 3. Since node 6 is now exposed, we need to reattach it back to the multicast tree. Under Case II, we will consider to employ one intermediate relay node (node 4 or 5) during the reattachment process. In particular, we will adjust the beam on one of nodes S, 2, and 3 to cover the intermediate relay node, i.e., nodes in the multicast tree or nodes covered by a node in the multicast tree, excluding node 1. For the pair of adjusted

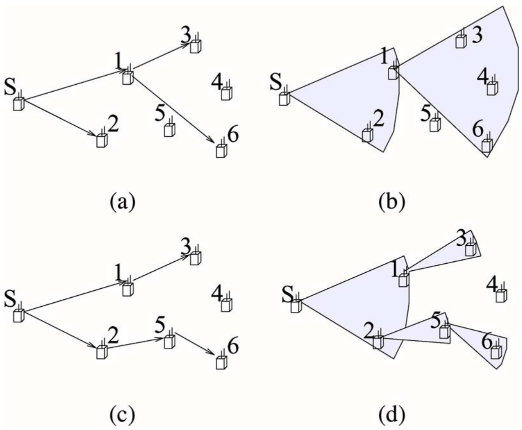

Fig. 9. Case II—Reattachment with intermediate relay nodes. (a) Multicast tree. (b) Beam on each node. (c) New Multicast tree. (d) New beam on each node.

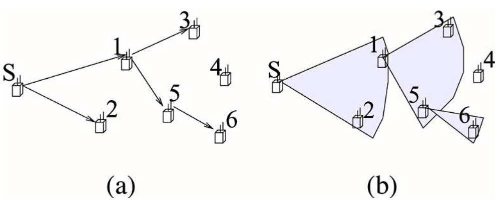

Fig. 10. Special case of Case II. (a) New Multicast tree. (b) New beam on each node.

node and intermediate relay node, define the pair lifetime as the smaller lifetime of their node lifetimes. The reattachment is successful only if the pair lifetime is larger than node 1's lifetime before pushing out node 6. If there are multiple successful reattachment options to choose from, we will choose the pair of nodes that yields the largest pair lifetime. For example, suppose that the node pair 2 and 5 yields the largest pair lifetime among all possible options; then, the MLR-MD algorithm will choose this pair of nodes and generate a new beam at node 2 to cover node 5 and another new beam at node 5 to cover node 6, respectively [see Fig. 9(d)]. The corresponding new logical multicast routing tree is shown in Fig. 9(c).

In the special case, suppose that we find the best option that is to choose the node pair 1 and 5. In this case, we need to readjust the beam on node 1 to cover the intermediate relay node (i.e., node 5). Since each node is allowed to generate one beam for the directional antenna under our investigation, we have to modify the existing beam from node 1 to cover both node 5 and node 3. In this case, node 1's beamwidth is increased (after first decreasing its beam radius to push out node 6). This solution is shown in Fig. 10(b), with corresponding logical multicast routing tree shown in Fig. 10(a).

The pseudocode of Case II is shown in Fig. 4 as Case2A(), and the special case of Case II is shown as Case2B(). In the special case, node i's lifetime is smaller than that in Case II; therefore, MLR-MD tries Case2B() last.

Between two network lifetime increases, it is possible that MLR-MD removes one logical link from a logical multicast tree, and after some iterations, MLR-MD tries to add this link back to an advanced logical multicast tree. We further impose a limit wherein once MLR-MD removes a logical link, it cannot use this link until the network lifetime is increased. It is easy to verify that the computational complexity of the MLR-MD algorithm is strictly polynomial.

V. Simulation Results

A. Simulation Settings

In this section, we use simulation results to illustrate the behavior and performance of the proposed MLR-MD algorithm and compare the algorithm to the D-MIP algorithm. For comparison, we also show results for multicast routing under the MPR paradigm, where a broadcast tree is obtained first by a spanning-tree-like technique and is then pruned to a multicast tree [25].

In our numerical investigation, we assume that the energy consumption in (3) is independent of ω. Further, we define $p_u^{\mathrm{T}}(\rho, \theta)$ as follows [26]:

$$p_u^{\mathrm{T}}(\rho, \theta) = \max\left\{\frac{\theta}{360}\rho^{\alpha},\ p_{\min}\right\} \quad (4)$$

where α is the path loss index and is typically within $2 \le \alpha \le 4$ [20], and $p_{\min}$ is the minimum power that is needed to generate a beam.

We consider networks of various sizes consisting of either 10, 20, 50, or 100 nodes. For 10-, 20-, and 50-node networks, we assume that the nodes are randomly deployed in a 5 by 5 unit square region, where the distance unit is consistent to that for ρ in (4). For 100-node networks, we assume the nodes are randomly deployed over a 15 by 15 unit square region. In all cases, we assume that each node starts with 200 units of energy, with the energy unit consistent with that in (4).

We are interested in an online operation where multicast requests arrive sequentially over time. The source of the multicast request is chosen at random, and the multicast group is also a random group of nodes in the network (excluding the source node). For each multicast request, the amount of data generated by the source node is uniformly chosen between [10, 100] units, and transmission rate at the source node is 10 units of data per time unit. In our simulation, we assume $\alpha = 4$ in (4), $p_u^{\mathrm{T}} \gg p_u^{\mathrm{R}}$, and $p_u^{\mathrm{T}} \gg p_u^{\mathrm{P}}$. That is, RF transmission energy is the dominant source of energy consumption. We also assume $p_{\min} = 0$ in (4). For the bounds of beamwidth for directional antennas, we set $\theta_{\min} = 30$ and $\theta_{\max} = 360$ (both in degrees).

For both MLR-MD and D-MIP algorithms, routing topology is dynamically changed every time unit, as discussed in Section V (if there is remaining data to send at the source node), where the time unit can be defined to reflect practical settings. For MPR, routing is only performed for each multicast request and remains static (fixed routing).

Although multicast session requests arrive to the network sequentially, it is possible that a new multicast session request arrives (at a different source node) before the previous multicast session terminates. That is, we may have multiple multicast sessions in the network at the same time. Our online MLR-MD algorithm (so does D-MIP algorithm) still works since it will consider multicast routing for each session independent from other on-going sessions in the network. Although multiple

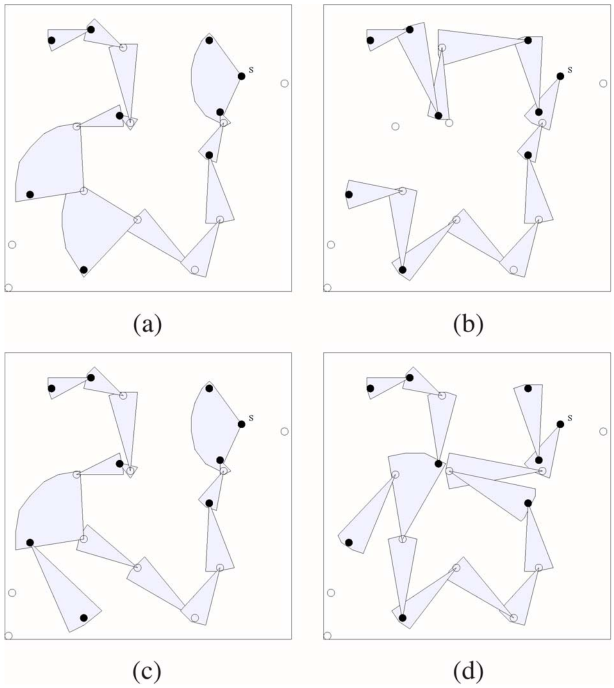

Fig. 11. Multicast tree under MPR, D-MIP, and MLR-MD. (a) MPR and D-MIP at time 0. (b) MLR-MD at time 0. (c) D-MIP at time 8. (d) MLR-MD at time 2.

concurrent sessions do not pose any difficulty to our algorithm, they do introduce a subtle issue in the accounting of total network lifetime. A logical approach to address this issue is to consider data volume that is being transmitted. As data transmission rate is common for all nodes (10 units of data/time unit), the total amount of data that has been transmitted successfully by each multicast source node should be directly related to network lifetime calculation. Following this reasoning, any time overlap of multiple multicast communication sessions should be counted multiple times, corresponding to the time overlap of multiple concurrent multicast sessions.

B. Results

For each network size (10, 20, 50, 100), we run simulations 100 times by generating 100 network topologies randomly and run the three algorithms on each topology. Before we show complete numerical results, we illustrate how the multicast routing looks like for each algorithm at some time instances under a particular topology. Fig. 11(a) and (b) shows a 20-node network and the multicast tree under MPR, D-MIP, and MLR-MD for the first multicast session request (at time 0). In this multicast, node S is the source node, and there are eight destination nodes (filled in black color). The total data volume that needs to be sent at the source is 90 units for the first multicast session. At time 0, the remaining energy at each node is its initial energy (200 units). Also at time 0, it happens that D-MIP has the same multicast tree as MPR [Fig. 11(a)], while MLR-MD has a different multicast tree topology [Fig. 11(b)].

Recall that both D-MIP and MLR-MD update routing topology dynamically, and the routing algorithm is executed every time unit. At time 2, MLR-MD updates its multicast tree as

TABLE I
STATISTICAL DATA OF NORMALIZED NETWORK LIFETIME PERFORMANCE OF MLR-MD AND D-MIP ALGORITHMS WITH RESPECT TO MPR

N	Algorithm	Average	Best	Worst	Standard Deviation
10	D-MIP	245.68	1466.67	100.00	63.69
	MLR-MD	336.54	1861.76	98.08	93.02
20	D-MIP	276.91	1210.11	100.00	77.79
	MLR-MD	359.72	1821.35	100.00	117.42
50	D-MIP	383.15	1203.24	154.57	138.07
	MLR-MD	504.38	1894.44	161.32	213.27
100	D-MIP	307.62	973.91	98.98	167.39
	MLR-MD	424.67	1132.61	105.26	242.34

shown in Fig. 11(d). Under the D-MIP algorithm, it turns out that multicast tree remains the same as that constructed at time 0 until time advances to eight, with a new topology shown in Fig. 11(c). Recall that for MPR, its multicast routing tree is fixed throughout this multicast, i.e., the multicast tree in Fig. 11(a) is used for MPR until the end of this multicast. Upon successive multicast requests over time (with source and destination nodes randomly chosen for each request), we calculate the network lifetime for MPR, D-MIP, and MLR-MD, respectively.

We now summarize our numerical results (by performing 100 simulations for each network size). Instead of showing the absolute network lifetime values, we find that it is more meaningful to show the normalized network lifetime for easy comparison. Define the normalized network lifetime as the network lifetime obtained by MLR-MD or D-MIP, divided by the network lifetime obtained by MPR. The average, best-case, worst-case, and standard deviation (all in percentage) are shown in Table I. For 100-node networks, D-MIP obtains 207% improvement, while MLR-MD is able to achieve a 324% improvement on average compared with MPR. For the best case, MLR-MD obtains 1032% improvement, while D-MIP obtains 873% improvement. Recall that D-MIP takes explicit consideration of a node's remaining energy in routing and is a cost-based algorithm. On the other hand, the proposed MLR-MD algorithm directly addresses the lifetime issue in algorithm design and, thus, is able to achieve better network lifetime performance over D-MIP on average. Simulations on 10-, 20-, and 50-node networks show similar results. To get a sense of what the actual network lifetimes look like under different multicast routing algorithm in real time unit, Table II shows the first 20 sets of results for the 50-node network under the MPR, D-MIP, and MLR-MD algorithms.

VI. RELATED WORK

The most significant theoretical work related to this paper (i.e., [4], [6], [9], [11], [14], and [17]) has been discussed in detail in Section III. In this section, we briefly review other relevant work that contributed to the background of our investigation.

There have been many recent papers addressing minimum energy routing for broadcast or multicast problem. Since this problem is NP-hard (see Lemma 1 in Section III), many heuristics have been proposed for broadcast (e.g., [2], [4], [5], [8], and [13]) and multicast (e.g., [24] and [25]). In [23], Wan *et al.* explored the performance of several heuristic algorithms by analyzing their competitive ratio (the heuristic result divided by

TABLE II
NETWORK LIFETIME OF MPR, D-MIP, AND MLR-MD FOR 50-NODE NETWORK

Index	MPR	D-MIP	MLR-MD	Index	MPR	D-MIP	MLR-MD
1	246.5	699.8	872.7	11	207.1	576.9	642.4
2	74.4	191.9	249.3	12	71.5	707.8	883.3
3	167.2	497.9	730.2	13	145.0	425.9	579.4
4	165.2	552.9	684.4	14	92.8	231.9	293.3
5	147.3	328.9	505.0	15	311.5	774.2	775.4
6	93.1	268.9	699.1	16	237.8	738.9	950.5
7	207.1	610.9	820.2	17	55.3	344.9	488.1
8	161.6	412.8	424.3	18	21.6	259.9	409.2
9	69.8	490.6	549.0	19	72.5	491.8	676.3
10	158.2	337.9	401.9	20	114.7	496.7	540.5

the optimal result). In particular, they analyzed the competitive ratios for the minimum spanning tree, shortest path tree, and BIP (see discussion in Section IV) and found that BIP offers the best performance.

There is a rich literature on online algorithms (see [12] and references therein) and online algorithms for network routing in particular (see [16] and references therein). In [18], Li *et al.* proposed an online algorithm for minimum energy routing. As discussed, the minimum energy routing may not provide a good performance in network lifetime. In [15], Kar *et al.* offered an online algorithm to maximize the capacity for unicast communications in energy-constrained *ad hoc* networks. In [1], Adamou and Sarkar proposed an online algorithm to maximize the node lifetime rather than the network lifetime.

The problem for maximizing the lifetime for a sequence of multicast requests, where the routing tree for each multicast can be updated over time, is shown to be NP-hard (see Theorem 2 in Section III). Currently, there are only two heuristic algorithms addressing this problem, i.e., the work of Kang and Poovendran in [14] for the broadcast problem (a special case of the multicast problem) and the work of Wieselthier *et al.* in [26], whose performance was shown in the numerical results in Section V.

VII. CONCLUSION

In this paper, we investigated the multicast routing problem for energy-constrained wireless *ad hoc* networks, where each node is equipped with a single-beam directional antenna. We are interested in an online routing algorithm for successive multicast communication requests with the aim of maximizing the network lifetime. The main contributions of this paper are 1) some important theoretical understandings on various multicast problems for energy-constrained wireless *ad hoc* networks and 2) the development of an online algorithm that takes the network lifetime consideration directly into iterative calculations. We showed that an algorithm that is designed under this methodology is able to provide consistent performance improvement over an algorithm that takes remaining energy into iterative calculations.

REFERENCES

[1] M. Adamou and S. Sarkar, "A framework for optimal battery management for wireless nodes," in *Proc. IEEE Infocom*, New York, Jun. 23–27, 2002, pp. 1783–1792.

[2] M. Agarwal, J. H. Cho, L. X. Gao, and J. Wu, "Energy efficient broadcast in wireless *ad hoc* networks with hitch-hiking," in *Proc. IEEE Infocom*, Hong Kong, Mar. 7–11, 2004, pp. 2096–2107.

[3] D. Blough and S. Paolo, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary *ad hoc* networks," in *Proc. ACM Mobicom*, Atlanta, GA, Sep. 23–28, 2002, pp. 183–192.

[4] M. Cagalj, J. P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," in *Proc. ACM Mobicom*, Atlanta, GA, Sep. 23–26, 2002, pp. 172–182.

[5] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized minimum-energy broadcasting in ad-hoc networks," in *Proc. IEEE Infocom*, San Francisco, CA, Apr. 1–3, 2003, pp. 1001–1010.

[6] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: Integer programming formulations," in *Proc. IEEE Infocom*, San Francisco, CA, Apr. 1–3, 2003, pp. 2210–2217.

[7] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, "MDLT: A polynomial time optimal algorithm for maximization of time-to-first-failure in energy constrained broadcast wireless networks," in *Proc. IEEE Globecom*, San Francisco, CA, Dec. 1–5, 2003, pp. 362–366.

[8] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, "R-shrink: A heuristic for improving minimum power broadcast trees in wireless networks," in *Proc. IEEE Globecom*, San Francisco, CA, Dec. 1–5, 2003, pp. 523–527.

[9] P. Floreen, P. Kaski, J. Kohonen, and P. Orponen, "Multicast time maximization in energy constrained wireless networks," in *Proc. Joint Workshop Found. Mobile Comput.*, San Diego, CA, Sep. 2003, pp. 50–58.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.

[11] S. Guo and O. Yang, "Antenna orientation optimization for minimum-energy multicast tree construction in wireless *ad hoc* networks with directional antennas," in *Proc. ACM MobiHoc*, Tokyo, Japan, May 24–26, 2004, pp. 234–243.

[12] S. Irani and A. R. Karlin, "On online computation," in *Approximation Algorithms for NP-Hard Problems, ACM SIGACT News Archive*, D. Hochbaum, Ed. New York: ACM, Jun. 1997, vol. 28, pp. 521–564. issue 2.

[13] I. Kang and R. Poovendran, "A novel power-efficient broadcast routing algorithm exploiting broadcast efficiency," in *Proc. IEEE Veh. Technol. Conf.*, Orlando, FL, Oct. 6–9, 2003, pp. 2926–2930.

[14] I. Kang and R. Poovendran, "Maximizing network lifetime of broadcast over wireless stationary *ad hoc* networks," *ACM/Springer MONET*, vol. 10, no. 6, pp. 879–896, Dec. 2005.

[15] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas, "Routing for network capacity maximization in energy-constrained ad-hoc networks," in *Proc. IEEE Infocom*, San Francisco, CA, Apr. 1–3, 2003, pp. 673–681.

[16] S. Leonardi, "On-line network routing," in *Online Algorithms: The State of the Art*, vol. LNCS 1442, A. Fiat and G. Woeginger, Eds. New York: Springer-Verlag, 1998, pp. 242–267.

[17] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless *Ad-hoc* networks," in *Proc. ACM Mobicom*, Rome, Italy, Jul. 16–21, 2001, pp. 97–107.

[18] Q. Li, J. Aslam, and D. Rus, "Distributed energy-conserving routing protocols," in *Proc. 36th Annu. Hawaii Int. Conf. Syst. Sci.*, Big Island, HI, Jan. 6–9, 2003, p. 301b.

[19] C. H. Papadimitriou, *Computational Complexity*. Boston, MA: Addison-Wesley, 1994.

[20] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[21] T. S. Rappaport, *Smart Antennas: Adaptive Arrays, Algorithms, and Wireless Position Location*. Piscataway, NJ: IEEE, 1998.

[22] T. K. Sarkar, M. C. Wicks, M. Salazar-Palma, and R. J. Bonneau, *Smart Antennas*. Hoboken, NJ: Wiley, 1996.

[23] 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*, Anchorage, AK, Apr. 22–26, 2001, pp. 1162–1171.

[24] B. Wang and S. K. S. Gupta, “G-REMiT: An algorithm for building energy efficient multicast trees in wireless *ad hoc* networks,” in *Proc. IEEE Int. Symp. Netw. Comput. and Appl.*, Cambridge, MA, Apr. 16–18, 2003, pp. 265–272.

[25] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “Energy-efficient broadcast and multicast trees in wireless networks,” *ACM/Kluwer Mobile Netw. and Appl. (MONET)*, vol. 7, no. 6, pp. 481–492, Dec. 2002.

[26] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “Energy-aware wireless networking with directional antennas: The case of session-based broadcasting and multicasting,” *IEEE Trans. Mobile Comput.*, vol. 1, no. 3, pp. 176–191, Jun.–Sep. 2002.

Y. Thomas Hou (S’91–M’98–SM’04) received the B.E. degree from City College of New York in 1991, the M.S. degree from Columbia University, New York, in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 1998, all in electrical engineering.

From 1997 to 2002, he was a Researcher with Fujitsu Laboratories of America, IP Networking Research Department, Sunnyvale, CA (Silicon Valley). Since fall 2002, he has been an Assistant Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg. His research interests are in the algorithm design and optimization for network systems. His current research focuses on wireless *ad hoc* networks, sensor networks, and video over *ad hoc* networks. In recent years, he worked on scalable architectures, protocols, and implementations for differentiated services Internet; service overlay networking; multimedia streaming over the Internet; and network bandwidth allocation policies and distributed flow control algorithms. He has published over 100 journals and conference papers in the above areas.

Dr. Hou is a member of ACM.

Yi Shi (S’02) received the B.S. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1998, the M.S. degree from the Institute of Software, Chinese Academy of Science, Beijing, China, in 2001, and the second M.S. degree from Virginia Tech, Blacksburg, in 2003, all in computer science. He is currently working toward the Ph.D. degree in electrical and computer engineering at Virginia Tech.

His current research focuses on algorithms and optimization for wireless sensor networks, *ad hoc* networks, and UWB networks. His work has appeared in journals and highly selective international conferences such as ACM Mobicom, ACM Mobihoc, and IEEE Infocom.

Mr. Shi was a recipient of the Meritorious Award at the International Mathematical Contest in Modeling in 1997 and 1998.

Hanif D. Sherali received the B.S. degree from Bombay University, Mumbai, India, in 1975 and the M.S. and Ph.D. degrees from Georgia Institute of Technology, Atlanta, in 1977 and 1979, respectively.

He is a W. Thomas Rice Endowed Chaired Professor of engineering with the Industrial and Systems Engineering Department, Virginia Tech, Blacksburg. His area of research interest is in discrete and continuous optimization, with applications to location, transportation, and engineering design problems. He has published about 200 papers in operations research journals, has coauthored four books in this area, and serves on the editorial board of eight journals.

Prof. Sherali is a member of the U.S. National Academy of Engineering.

Jeffrey E. Wieselthier (S’67–M’69–SM’88–F’07) was born in Brooklyn, NY, in 1949. He received the S.B. degree from the Massachusetts Institute of Technology, Cambridge, in 1969, the M.S. degree from Johns Hopkins University, Baltimore, MD, in 1971, and the Ph.D. degree from University of Maryland, College Park, in 1979, all in electrical engineering.

From 1969 to 1979, he was with the Naval Surface Warfare Center, White Oak, Silver Spring, MD. Since 1979, he has been with the Information Technology Division, Naval Research Laboratory, Washington, DC, where he is a Senior Researcher and the Head of the Wireless Network Theory Section of the Networks and Communication Systems Branch. In addition, he is currently a Program Manager in communications and networking for the Office of Naval Research. He has studied a variety of communication networking problems, including multiple access and routing in spread-spectrum networks and the use of neural networks and other approaches for network performance evaluation, optimization, and control. His current research interests include wireless communication networks, with an emphasis on issues relating to energy-aware and cross-layer operation of *ad hoc* and sensor networks.

Dr. Wieselthier was the Lead Guest Editor of the two-part Special Issue on Wireless *Ad hoc* Networks, which was published in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in 2005. He was a Technical Program Cochair of the Third IEEE Symposium on Computers and Communications in Athens, Greece, in 1998 and Treasurer of the 1991 IEEE International Symposium on Information Theory in Budapest, Hungary. He won the IEEE Fred W. Ellersick Award for the Best Unclassified Paper at MILCOM 2000. He is on the Editorial Board of Elsevier’s journal *Ad hoc Networks*. He is a member of Eta Kappa Nu and Sigma Xi.