On Maximizing Lifetime of Multicast Trees in Wireless Ad hoc Networks

Bin Wang* and Sandeep K. S. Gupta*

Abstract

This paper presents a distributed algorithm called L-REMiT for extending the lifetime of a source-based multicast tree in wireless ad hoc networks (WANET). The lifetime of a multicast tree is the duration from the formation of the tree to the time when the first node fails due to battery energy exhaustion. L-REMiT assumes that the energy consumed to forward a packet is proportional to the forwarding distance and that WANET nodes can dynamically adjust their transmission power. The task of extending the lifetime of a multicast tree is formulated as the task of extending the lifetime of bottleneck nodes in the tree. The number of multicast packets which a bottleneck node can forward, as determined by its residual battery energy and the distance of its farthest child node, is minimum over all the nodes in the multicast tree. Lifetime of a bottleneck node is improved by reassigning its farthest children to other nodes in the tree with the goal of improving the lifetime of the multicast tree. Nodes only require information from their neighbors for refining the tree in a distributed manner. Simulation results show that L-REMiT has low overhead and performs better than BIP/MIP and EWMA algorithms.

1 Introduction

Multicasting enables a single node in the network to communicate efficiently with multiple nodes in the network. A multicast service is needed for many distributed applications such as distributed resource allocation and replicated file systems. In wireless ad hoc networks (WANETs), including sensor networks, all nodes in the network cooperate to provide networking facilities to various distributed tasks. In such networks, nodes are usually powered by limited source of energy. As opposed to a wired network, availability of limited energy at nodes of a WANET has an impact on the design of multicast protocols. Specifically, the set of network links and their capacities in WANETs is not pre-determined but depends on factors such as distance between nodes, transmission power, hardware implementation and environmental noise. This is one of the basic differences between wireless and wired networks.

*Department of Computer Science and Engineering, Arizona State University,Tempe, AZ 85287. Email:{Bin.Wang,Sandeep.Gupta}@asu.edu.

The energy-efficient broadcasting/multicasting tree problem was presented in [10]. Wieselthier et al. [10] proposed a "node-based" elastic model for wireless multicast and the concept of *wireless multicast advantage*. Since the problem of constructing the optimal energy-efficient broadcast/multicast tree is NP-hard [2], several heuristic algorithms for building a source based energy-efficient broadcast/multicast tree have been developed. Wieselthier et al. have proposed two centralized algorithms - BIP/MIP [10] and two distributed version of BIP algorithm - Dist-BIP-A,Dist-BIP-G [11] to build source-based broadcast/multicast trees. But these two distributed algorithms have slightly worse performance than its centralized version. Cagalj et al. [1] have presented an Embedded Wireless Multicast Advantage (EWMA) algorithm to reduce energy consumption of source-based broadcast tree. They also described a distributed version of EWMA algorithm.

As all of the multicast data traffic must go through the intermediate nodes, the above algorithms can result in rapid depletion of energy at intermediate nodes; possibly leading to network getting partitioned and interruption to the multicast service. New approaches are therefore needed to extend the network lifetime. The problem of maximizing network lifetime was studied in [7]. Both Wieselthier et al. [12] and Kang et al. [5] extended BIP/MIP by using residual battery energy in their energy metric to extend the broadcast/multicast tree lifetime. Throughout this paper, we will use MIP and L-MIP to denote MIP without considering residual battery energy and MIP considering residual battery energy , respectively.

In this paper, we focus on source initiated multicasting of data in WANETs. Our main objective is to extend the lifetime of a *source-based multicast tree*. A source-based multicast tree is rooted at a multicast source node and covers all the other multicast group members who are receivers. We define **lifetime** of a multicast tree as the time duration starting from beginning of multicast service until the first node in the multicast tree fails due to battery energy exhaustion. We propose a distributed protocol called L-REMiT which is a part of a suite of protocols called REMiT (Refining Energy efficiency of Multicast Trees) which we are designing to achieve various energy-efficiency goals related to multicasting in WANETs. REMiT protocols are distributed protocols which refine the energy-efficiency of a pre-existing multicast tree using local knowledge at each node. The

Proceedings of the 2003 International Conference on Parallel Processing (ICPP'03)
0190-3918/03 $ 17.00 © 2003 IEEE

REMiT protocols can be categorized along energy-metric dimension (minimizing energy-consumption or maximizing lifetime) and multicast-tree type dimension (source based or group-shared tree). For example, we presented G-REMiT [8] and S-REMiT [9] which minimize energy-consumption for group-shared trees and source-based trees, respectively. In this paper, we present L-REMiT which uses *minimum-weight spanning tree* (MST) as initial tree and improves its lifetime by switching (reassigning) children of a "bottleneck" node to another node in the tree. A bottleneck node is one which currently has minimum energy level among all the multicast tree nodes. Each such switching step is called a "refinement". A multicast tree is obtained from the "refined" MST (after all possible refinements have been performed) by pruning the tree to reach only multicast group nodes. L-REMiT algorithm is distributed in the sense that each node has only got local view of the tree and each node can independently switch its parent as long as the multicast tree remains connected. Our simulation results show that L-REMiT outperforms the most prominent proposals in the literature: BIP/MIP and EWMA.

2 System Model and Assumptions

We assume each node in the WANET with N nodes has an unique identifier i, $1 \leq i \leq N$. Each node has only local view of the network and knows the distance between itself and its neighbor nodes using some distance estimation method [6]. The connectivity in the network depends on the transmission power of the nodes. Each node can dynamically change its transmission power level. A node may use a different power level for each multicast tree in which it participates. For simplicity, we assume that all data packets are of the same size. Let $E_{i,j}$ be the minimum energy needed for link between nodes i and j for a data packet transmission. We assume the following model for $E_{i,j}$ [3]:

$$E_{i,j} = E_{elec} + K(r_{i,j})^{\alpha}, \qquad (1)$$

where $r_{i,j}$ is the Euclidean distance between nodes i and j, E_{elec} is a distant-independent constant that accounts for overheads of electronics and digital processing, K is a constant dependent upon the properties of the antenna and α, called the propagation loss exponent, is a constant which is dependent on the propagation losses in the medium. For **long range radios**, $E_{elec} << K(r_{i,j})^{\alpha}$, so $E_{i,j} \approx K(r_{i,j})^{\alpha}$. On the other hand, for **short range radios**, E_{elec} is not negligible, since E_{elec} can substantially exceed the maximum value of the $K(r_{i,j})^{\alpha}$ [3].

Compared to wired networks, WANETs have "wireless multicast advantage" [10] which means that all nodes within communication range of a transmitting node can receive a multicast message with only one transmission if they all use omni-directional antennas. We assume the same in our model. Further, every node (say node i) has two coverage areas. **Control coveRage area (CR_i)** and **Data coveRage area (DR_i)**, such that $DR_i \subseteq CR_i$. These coverage areas depend upon the transmission power selected by node i to transmit its control and data packets, respectively. For example, in Figure 1, radius of CR_{10} is 3.2, i.e., node 10's control message may reach nodes 6, 7, and 9, but if $DR_{10} = 2.75$, its data message may only reach only nodes 6 and 9, but not node 7.

Neighbors of node i are the nodes within CR_i. We use V_i, $V_i \subseteq CR_i$, to denote the set of **tree neighbors** of node i, i.e, those neighbors of node i which also belong to the multicast tree T. A **connected tree neighbor** j of a node i is a tree neighbor of node i which is connected to the node by a *branch*, i.e., link $(i,j) \in T$. A **non-connected tree neighbor** j of a node i is a tree neighbor of node i which is connected to the node i by more than one branch in T, i.e. the length of the unique path between i and j in T is greater than 1. We denote the set of connected and non-connected tree neighbors of node i as CTN_i and $NCTN_i$, respectively. Note that $NCTN_i = V_i - CTN_i$.

We assume that node s is the source node of the multicast tree whose lifetime is being maximized. A message to be multicasted to the group members is forwarded along the branches starting from source node s: every node on the tree which receives a new multicast message from its parent node forwards the message to all of its children nodes. Figure 1 gives a source-based multicast tree example, node 10 is the source node, nodes 1, 2, 3, 4, 5, 7, 8, 10 and 11 are multicast group members, nodes 6 and 9 are non-group nodes which serve as forwarding nodes in the multicast tree.

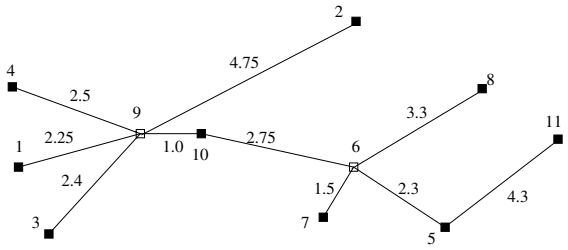

Figure 1. A Multicast Tree. (Only branches are shown for clarity and since L-REMiT ignores other links. Branch labels denote the Euclidean distance between their endpoints.)

3 Problem Definition

In this section, we define the problem of maximizing the lifetime of a multicast tree. Before we do so, we define the notion of energy cost of a node in a multicast tree and the lifetime of a multicast tree.

3.1 Energy Cost of a Node in a Multicast Tree

In a source based multicast tree, the energy consumption at every tree node is determined by the distance of the children nodes. For example, consider node 10's source-based multicast tree shown in Figure 1. Node 10 will send each

multicast message along the branch to nodes 6 and 9. Node 9 will forward them to nodes 1, 2, 3 and 4. Similarly, node 6 will forward them to nodes 5, 7 and 8, and so on. The energy consumed at node 9 for each multicast message from node 10, using the source-based multicast tree in Figure 1, is $max(E_{9,1}, E_{9,2}, E_{9,3}, E_{9,4}) = E_{9,2}$.

Let d_i be i's maximum length between i and it's farthest children. The **energy cost** of node i in a multicast tree T, $E(T,i)$ is:

$$E(T,i) = \begin{cases} E_{elec} + Kd_i^{\alpha} & \text{if } i \text{ is the source node;} \\ E_{elec} + Kd_i^{\alpha} + E_{recv} & \text{if } i \text{ is neither the source nor a leaf node in } T; \\ E_{recv} & \text{if } i \text{ is a leaf node in } T, \end{cases} \quad (2)$$

where E_{recv} denotes the energy cost to receive a data packet. We assume that E_{recv} is the same for every node.

3.2 Multicast Lifetime Metric

The lifetime of a node in a multicast tree, given its current battery energy level, is *the maximum number of multicast packets that may be transmitted by the node*, assuming that the node does not participate in any other packet transmission. If the residual battery energy at node i is R_i, the maximum number of packets that node i can transmit is $R_i/E(T,i)$. Hence, for a node i in Tree T, we define the node i's **multicast lifetime** as:

$$LT(T,i) = \frac{R_i}{E(T,i)}. \quad (3)$$

The lifetime of a multicast tree is *the maximum number of packets that may be transmitted over the multicast tree*, assuming that all nodes belonging to the multicast tree do not participate in any other packet transmissions. Thus, the lifetime of a multicast tree T is the minimum lifetime of any node in T:

$$LT(T) = \min_{\forall i \in T} LT(T,i) = \min_{\forall i \in T} \frac{R_i}{E(T,i)}. \quad (4)$$

We call the node with minimum multicast lifetime in a multicast tree to be its **bottleneck node**.

So the *problem of maximizing the lifetime of a multicast tree* becomes the problem of maximizing the lifetime of the tree's bottleneck node.

4 L-REMiT Algorithm

L-REMiT tries to improve the lifetime of bottleneck nodes in the initial multicast tree by changing bottleneck node's children node so that the tree's lifetime is higher. It uses MST as the initial tree since MST is useful for various purpose, such as broadcast. Further, MST performs quite well for our problem based on our experimental results. We use $Change_i^{x,j}$ to refer to the refinement step in which (bottleneck) node x's child node i switches its parent from node x to node j. Let T be a multicast tree, and T' be the resulting graph after refinement $Change_i^{x,j}$ is applied to T. The following lemmas, presented here without proof, guarantees that T' is a tree and identify which nodes lifetime change due to refinement:

Lemma 1 *If node j is not a descendant of node i in tree T, then the tree remains connected after $Change_i^{x,j}$.*

Lemma 2 *Nodes j and x are the only nodes in the tree whose multicast lifetime may be affected by $Change_i^{x,j}$.*

4.1 Refinement Criterion

The lifetime of a tree may change as a result of performing a refinement. We call the change in the tree's lifetime due to refinement $Change_i^{x,j}$ as $gain$ in the tree's lifetime, i.e. $gain = LT(T') - LT(T)$. L-REMiT uses $gain$ as the criterion for changing the parent of a node: the refinement $Change_i^{x,j}$ is performed only if it is expected that $gain > 0$.

For example, consider the multicast tree in Figure 1 which is node 10's source-based multicast tree. If node 9 is the bottleneck node of the tree, we show how can node 9's children change their parent to increase the lifetime of the multicast tree. In this example, we consider how node 2 decides to change its parent from node 9, to node 6. We refer to this change event as $Change_2^{9,6}$. To simplify the following explanation, we assume that $K = 1, \alpha = 2, E_{elec} = 0, E_{recv} = 1$, and $R_i = 100$ unit, $i = 1, 2, \ldots, 11$:. Using Formula (3), node 2 will estimate the change in the lifetime at node 2, 9 and 6 if it makes $Change_2^{9,6}$. We use T and T' to denote the multicast tree before and after $Change_2^{9,6}$. First, node 2 will estimate the current lifetime at node 2, 6 and 9: $LT(T,2) = 100/1 = 100$; $LT(T,6) = 100/(r_{6,8}^2 + 1) = 8.41$; $LT(T,9) = 100/(r_{9,2}^2 + 1) = 4.24$. Similarly, node 2 can estimate the new lifetime at node 2, 9, and 6 after $Change_2^{9,6}$, $LT(T',2) = 100/1 = 100$; $LT(T',6) = 100/(r_{6,2}^2+1) = 7.16$; $LT(T',9) = 100/(r_{9,3}^2+1) = 6.17$. After $Change_2^{9,6}$, the *expected gain* ($g_2^{9,6}$) of the tree obtained by switching at node 2 from node 9 to node 6 is:

$$g_2^{9,6} = \min\{LT(T',2), LT(T',9), LT(T',6)\} - LT(T,9) = 6.17 - 4.24 = 1.93.$$

Likewise node 2 can compute the gain in lifetime if it switches to node 10 and node 8: $g_2^{9,10} = 1.64$ and $g_2^{9,8} = 2.92$, respectively.

By comparing the gains, node 2 selects a node with the highest positive lifetime gain as the new parent. Thus node 8 is selected as the new parent of node 2. Node 2 will select node 8 as its parent node in the multicast tree and disconnect from node 9. So in Figure 1, branch between nodes 2 and 9 will be deleted, and branch between nodes 2 and 8 will be added to the multicast tree. Because DR_9 does not need to cover node 2 any more, radius of DR_9 will decrease to $r_{9,4}$. DR_8 should be changed to cover node 2, hence radius of DR_8 will increase from 0 to $r_{8,2}$.

A node uses the locally computable expected gain (instead of $gain$) as a criterion for the tree refinement. In general, the expected gain $g_i^{x,j} = \min\{LT(T',i), LT(T',x), LT(T',j)\} - LT(T,x)$. Note, that $g_i^{x,j} > 0$ does not necessarily imply $gain > 0$, since there may be multiple bottleneck nodes in T. L-REMiT performs refinement steps until the expected gain from the refinements continues to be positive.

4.2 Performing Refinement $Change_i^{x,j}$

Following are the steps involved in tree refinement $Change_i^{x,j}$. First, find the bottleneck node, say node x. Second, identify the farthest child of node x, say node i (based on Formulas (2) and (3), lifetime of node x is determined by its farthest child). Third, compute the set S_i, which is a subset of $NCTN_i$, such that $S_i = \{k | k \in NCTN_i \bigcap k \notin \text{subtree of } i\}$. Selection of the new parent of node i from nodes in S_i guarantees that no cycle is formed or equivalently the tree is not fragmented as a result of $Change_i^{x,j}$. Fourth, select a node j from set S_i with the highest positive gain, $g_i^{x,j}$. Finally, node i changes its parent to be node j instead of node x.

4.3 Local Data Structure and Messages Types

Before describing a node's local data structure and message types used by our distributed protocol, we introduce the following notation. Let d_i' be the second maximum length of link between i and its children. We denote the two-tuple (d_i, d_i'), as l_i. Further, let node j be a neighbor of i, $j \in V_i$. We will use the notation $Data_k$ to denote the data associated with node k:

- $LT(T,k)$: multicast lifetime of k in the tree T;
- B_k: a list of bottleneck nodes in k's sub-tree (there may exists several tree nodes with the same minimum multicast lifetime), also we use b_k to denote one of the node in B_k;
- $CTNT_k$: a list of records of the type $(i, l_i, b_i), \forall i$ is k's child;
- $NCTNT_k$: a list of records of they type $(i, l_i), \forall i \in NCTN_k$.

L-REMiT uses the following message types:

- $TOKEN(x, i, b_k, LT(T', b_k), flag)$: sent to the bottleneck node x and returned to the source node s along the tree branches. $flag$ is a boolean value to represent the refinement was successful or not. This message is important and used throughout the second phase of L-REMiT. So it needs reliable passing between nodes.
- $JOIN\_REQ(i,j)$: sent by node i to node j requesting j to become its parent. This message is used in Step II.4 by node i to make $Change_i{}^{x,j}$.
- $JOIN\_REP(i,j)$: sent by j to reply node i's $JOIN\_REQ(i,j)$. This message is used in Step II.4 by node j to make $Change_i{}^{x,j}$.
- $LEAVE(i,x)$: sent by node i to leave parent node x. This message is used in Step II.4 by node i to make $Change_i{}^{x,j}$ and in Step II.7 by node i to leave the tree when i is a leaf node and non-group node.
- $ELECTION\_REQ(s,i)$: sent by node s to node i requesting election from node i. This message is used in Step I.2. by node s to request all the leaf nodes for bottleneck node election. This message is also used in Step II.6. by node s to request node i for bottleneck node election.
- $ELECTION(b_i, LT(T', b_i))$: bottleneck node election result sent by node i to its parent node. This message is used in Step I.2. by node i to submit election result of i's sub-tree.
- $NEIGHBOR\_UPDATE(i,x,j)$: sent by node i to nodes in V_i notifying $Change_i^{x,j}$. This message is used in Step II.4 by node i.

4.4 Distributed Protocol

L-REMiT consists of two phases: 1) multicast tree construction and 2) lifetime refinement. The **first phase** includes the following two steps:

I.1. **Building initial multicast tree:** All nodes run a distributed algorithm proposed by Gallager et al. [4] to build a MST of the wireless network, which is used as the initial multicast tree T. We require that after building T, each node in the multicast tree know its parent and children nodes with respect to the source node s. Further, each node i, $i \in T$, has all local information l_k ($\forall k \in V_i$).

I.2. **Bottleneck node election:** The source node s requests all of the nodes in T to elect the minimum multicast lifetime node in a bottom up manner from leaf nodes to the source node s. If i is a node on the tree, i needs to first find out b_i, a bottleneck node in i's sub-tree. Then node i informs its parent node the tuple (b_i, $LT(T, b_i)$) as the election result of node i's sub-tree. If node i is a leaf node of T, $b_i = i$. An intermediate tree node delays the computation until it obtains election results from all its children, before sending its election result to its parent node. Also node i records each of its child sub-tree's bottleneck node information in $Data_i$. This information is used in Step II.2 for selection of the new bottleneck node after a refinement. Note that B_i obtained by this election may not include all of the bottleneck nodes in node i's sub-tree. But the B_i obtained in the election is good enough for L-REMiT to proceed, and does not affect the results of L-REMiT algorithm.

The **second phase** proceeds in rounds coordinated by the source node s. Based on the bottleneck node election results, node s selects a bottleneck node x from B_s. The node s passes L-REMiT token to node x, then node x lets its farthest child, say node i, switch its parent to increase x's multicast lifetime. We use node j to denote the new parent of node i. After refinement $Change_i^{x,j}$, x passes the token

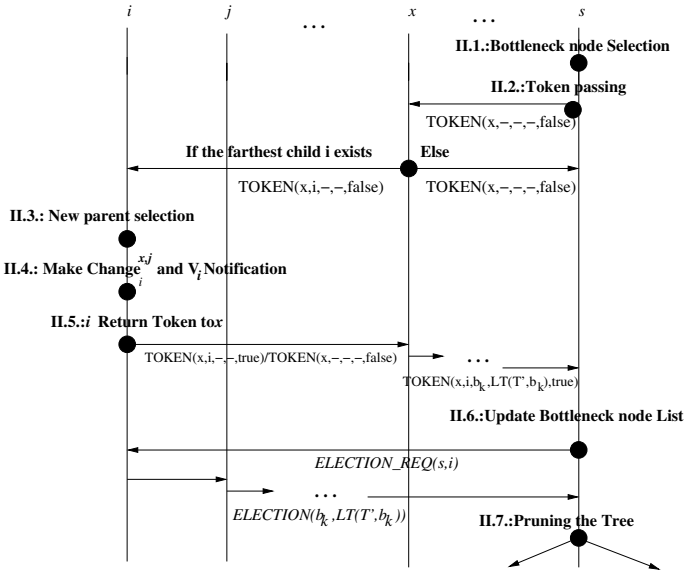

Figure 2. Overview of 2^{nd} Phase of L-REMiT

back to node s along the tree path from node x to node s. Similar to Step I.2, node k (k is ancestor of x) will re-elect bottleneck node b_k of Tree T' (T' is the multicast tree after $Change_i^{x,j}$) when node k is forwarding the token to its parent. After node s gets back the token, it requests node i to re-elect the new bottleneck node b_s from node i in a bottom up manner in Tree T'. After $Change_i^{x,j}$, b_x and b_j may be changed in Tree T', bottleneck node needs to be re-elected at all of node x and i's ancestors. So that node s can locate new b_s in Tree T' for the next round of refinement. The node s terminates L-REMiT algorithm when there is no lifetime gains by the current refinement step.

Following are the steps to improve lifetime of the multicast tree in the second phase (see Figure 2 for illustrations of these steps):

II.1. **Bottleneck node selection:** Node s selects a node x from B_s.

II.2. **Token passing and farthest children selection:** Node s gives $TOKEN(x,-,-,-,false)$ to node x. The node x selects a node i from the list of x's farthest children list (x may have several farthest children nodes at the same time). If node i does not exist, x return $TOKEN(x,-,-,-,false)$ to node s along the tree path from x to s, and goes to Step II.6; otherwise node x forwards $TOKEN(x,i,-,-,false)$ to node i, goes to next step.

II.3. **New parent selection:** Once node i gets the L-REMiT token, i selects new parent node j from S_i[1] with the highest positive $gain$: $g_i^{x,j} := \min\{LT(T',i), LT(T',x), LT(T',j)\} - LT(T,x)$. If there is no such node j available, node i constructs token as $TOKEN(x,-,-,-,false)$, then goes to Step II.5; otherwise, goes to the next step.

II.4. **Make $Change_i^{x,j}$ and V_i notification:** Node i makes $Change_i^{x,j}$ and notifies nodes in V_i. Node i constructs token as $TOKEN(x,i,-,-,true)$.

II.5. **Return token to x:** The node i returns the token to node x. The node x will update B_x and return $TOKEN(x,i,x,LT(T',x),true)$ or $TOKEN(x,-,-,-,false)$ to node s along the tree path from x to node s, then goes to next step.

II.6. **Update bottleneck node list at s:** The node s gets back the L-REMiT token. If $flag = false$, then it goes to the next step; otherwise node s requests i to do bottleneck election which is similar to Step I.2. After node s gets back the election results, it updates B_s and goes to Step II.1.

II.7. **Pruning the Tree:** The node s will request all of the tree node to prune the redundant transmissions that are not needed to reach the members of the multicast group from the tree. Then node s terminates L-REMiT protocol.

Following are two examples to illustrate the second phase of L-REMiT algorithm: 1) bottleneck node election; and 2) single refinement at a node. In these two examples, we use multicast tree in Figure 1.

Example 1: This example illustrates how to elect the bottleneck node. Node 10 requests all the nodes to elect the bottleneck node. Because node 1,2,3,4,7,8, and 11 are leaf nodes, they will submit their own multicast lifetime to their respective parent nodes. Once node 9 obtains all of the submission from nodes 1,2,3, and 4, it will compare these lifetime values with its own multicast lifetime. It finds that the bottleneck node of its subtree is itself. Then node 9 will submit the identifier of itself and lifetime value to its parent node, node 10. Similarly, node 6 finds out that its sub-tree's bottleneck node is node 6. Then node 6 submits this election result to node 10 also. Now node 10 obtains the election results from all of its children node. So node 10 finds that node 9 is the only bottleneck node in the tree. □

Example 2: This example illustrates a single refinement step. Based on the Example 1, node 9 is the bottleneck node. Node 10 passes the L-REMiT token to node 9. In turn, node 9 passes the token to its farthest children node 2. Once node 2 gets the L-REMiT token, node 2 performs the following steps:

1. Node 2 calculates $gains$ as explained previously in the paper and finds out $g_2^{9,8}$ is the highest positive value.
2. Node 2 makes $Change_2^{9,8}$ and notifies the nodes in V_2.
3. Finally, node 2 will pass the L-REMiT token back to its previous parent node 9. Node 9 passes the $TOKEN(9,2,9,LT(T',9),true)$ back to node 10, Then node 10 requests node 2 to do bottleneck election. Similar to Example 1, the bottleneck election will

[1] $S_i = \{k | k \in NCTN_i \bigcap k$ is not on sub-tree of $i\}$. Node i needs request k to send message to k's ancestors. If the i gets the message, then i knows k is on subtree of i. Otherwise the message will come to s at last. If s gets the message, it will forward it back to i. If i gets the message from s, then i knows k is not on i's sub-tree.

go through nodes 2, 8 and 6. At last node 10 gets back the election result from node 6. So node 10 updates B_{10} and finds that node 6 is the only new bottleneck node of tree T'.

□

4.5 Worst Case Complexity Analysis

The message complexity of bottleneck node election is $O(N)$, where N is the number of nodes in the network. The message complexity for changing a node's parent is $O(1)$. The message complexity of a round in which a tree refinement is performed is $O(\delta_{max} + 2H)$, where δ_{max} is the maximum number of neighbor in any node's Control coveRage area(CR), and H is the diameter of the network. Hence the message complexity of L-REMiT $O(N + R(\delta_{max} + 2H))$, where R is the number of rounds performed. The computational complexity of one refinement is $O(\delta_{max})$. Therefore, the computational complexity of L-REMiT is $O(R\delta_{max})$. The space complexity of L-REMiT for each node is $O(\delta_{max})$ since the size of V is $O(\delta_{max})$.

5 Simulation Results

We used simulations to evaluate the performance of L-REMiT algorithm. We compare our algorithm with MIP, L-MIP, MST, and EWMA-Dist (Distributed version of EWMA algorithm). Because EWMA-Dist algorithm is used for building broadcast tree, we extend EWMA-Dist algorithm for multicasting by pruning the redundant transmissions that are not needed to reach the members of the multicast group from the broadcast tree produced by EWMA-Dist algorithm. The simulations were performed using networks of four different sizes: 10, 40, 70, and 100 nodes. The distribution of the nodes in the networks and the residual battery energy at nodes are randomly generated. Every node is within the maximum transmission range of at least one other node in the network, i.e., the network is connected. We use two different E_{elec} values to represent the long range radio and short range radio. Based on the experiment data in [3], we decided to use $E_{elec} = 0$ to represent long range radio and $E_{elec} = 4r^{\alpha}$ to represent short range radio. We ran 100 simulations for each simulation setup consisting of a network of a specified size to obtain average $LT(T)$ with 95% confidence, the propagation loss exponent α is varied from 2 to 4. For each simulation setup, we use *normalized Lifetime* as the performance metric.

$$normalized\ Lifetime = \frac{LT(T_{alg})}{LT(T_{best})},$$

where $LT(T_{best}) = \max\{LT(T_{alg})\}, alg \in A = \{L-REMiT, MST, MIP, L-MIP, EWMA-Dist\}$.

5.1 Short Range Radios

For short range radios, the performance is shown in Figures 3 and 5. We can see the average $normalized\ Lifetime$ (show on the vertical axis) achieved by the algorithms on networks of different sizes (the horizontal axis). The figures show that the solutions for multicast tree obtained by L-REMiT have, on the average, higher $normalized\ Lifetime$ than the solutions of L-MIP, MIP, EWMA-Dist, and MST, when 100% and 50% of the nodes are group members with different energy cost at the receivers (This is also true for $\alpha = 3$, which is not shown in the figure).

5.2 Long Range Radios

For long range radios, the performance is shown in Figures 4 and 6. In the figures, we can see that the multicast trees produced by L-REMiT algorithm have, on the average, higher $normalized\ Lifetime$ than those obtained by the L-MIP, MIP, EWMA-Dist, and MST, when 100% and 50% of the nodes are group members with propagation loss exponent of $\alpha = 2$ and 4. However, we can notice that for the propagation loss exponent of $\alpha = 4$, L-MIP and L-REMiT have very similar performance. Thus the figure also reveals that the difference in performance decreases as the propagation loss exponent increases when 100% nodes are group member. The main reason for such behavior is that by increasing the propagation loss exponent, lifetime of the trees which use longer links decreases. Consequently, L-MIP and L-REMiT select their transmitting nodes to transmit at lower power levels. Hence, L-REMiT and L-MIP's broadcast trees (because 100% of the nodes are group nodes) become similar when α increases (This is also true for different group sizes, which is not shown in the figure). Further our simulation results show that energy overhead of L-REMiT is always below 1% of the toal energy consumption of all nodes in the multicast tree within the lifetime of the tree.

Based on our simulation results, we find that L-REMiT has better performance than L-MIP, MIP, EWMA, and MST for various scenarios. Because Dist-BIP-A and Dist-BIP-G [11] perform slightly worse than BIP algorithm (BIP is the broadcast case of MIP algorithm), L-REMiT should be better than the two distributed versions of BIP algorithm.

6 Conclusions

In this paper, we proposed L-REMiT algorithm for refining a multicast tree with the goal of extending its lifetime in a WANET. L-REMiT is a distributed algorithm which employs an energy consumption model for wireless communication which takes into account the energy losses due to radio propagation as well as transceiver electronics. This enables L-REMiT to adapt a given multicast tree to a wide variety of wireless networks irrespective of whether they use long-range radios or short-range radios. Implicitly, we have assumed that the L-REMiT token may not be lost; an issue we will address in our future work.

We showed that L-REMiT outperforms other proposals in the literature: MST, MIP, and EWMA. Further, the energy consumption overhead of the algorithm itself is very

small compared with the sum of energy consumption at all nodes in the multicast tree within the lifetime of the tree.

7 Acknowledgments

We thank the anonymous reviewers for their insightful comments which helped to improve the quality of the paper. This work is supported in part by NSF grants ANI-0123980 and ANI-0196156.

References

[1] M. Cagalj, J. P. Hubaux, and C. Enz. Minimum-energy broadcast in all-wireless networks: NP-Completeness and distribution issues. In *Proc. ACM MobiCom 2002*, pages 172–182, Atlanta, Georgia, Sept. 2002.

[2] A. E. F. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the complexity of computing minimum energy consumption broadcast subgraphs. In *Proc. 18th Annual Theoretical Aspects of Comp. Sc. (STACS)*, volume 2010, pages 121–131, Springer-Verlag, 2001.

[3] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proc. IEEE INFOCOM*, pages 1548–1557, Anchorage, AK, Apr. 2001.

[4] R. Gallager, P. A. Humblet, and P. M. Spira. A distributed algorithm for minimum weight spanning trees. *ACM Trans. Programming Lang. & Systems*, 5(1):66–77, Jan. 1983.

[5] I. Kang and R. Poovendran. On the lifetime extension of energy-constrained multihop broadcast networks. In *Proc. 2002 Int'l Joint Conf. on Neural Networks*, pages 365–370, Honolulu,Hawaii, May 2002.

[6] W. C. Y. Lee. *Mobile Communication Engineering*. McGraw-Hall, 1993.

[7] A. Misra and S. Banerjee. MRPC: Maximizing network lifetime for reliable routing in wireless environments. In *IEEE Wireless Communications and Networking Conf. (WCNC)*, Orlando, Florida, Mar. 2002.

[8] B. Wang and S. K. S. Gupta. G-REMiT: An algorithm for building energy efficient multicast trees in wireless ad hoc networks. In *IEEE Int'l Sym. Network Comp. and Applications (NCA-03)*, pages 265–272, Cambridge, MA, Apr. 2003.

[9] B. Wang and S. K. S. Gupta. S-REMiT: An algorithm for enhancing energy-efficiency of multicast trees in wireless ad hoc networks. In *IEEE 2003 Global Comm. Conf. (GLOBECOM 2003) (to appear)*, San Francisco, CA, Dec. 2003.

[10] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast tree in wireless networks. In *Proc. IEEE INFOCOM 2000*, pages 585–594, Tel Aviv, ISRAEL, Mar. 2000.

[11] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Distributed algorithms for energy-efficient broadcasting in ad hoc networks. In *IEEE Military Communications Conf.*, Anaheim, CA, Oct. 2002.

[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):113–131, 2002.

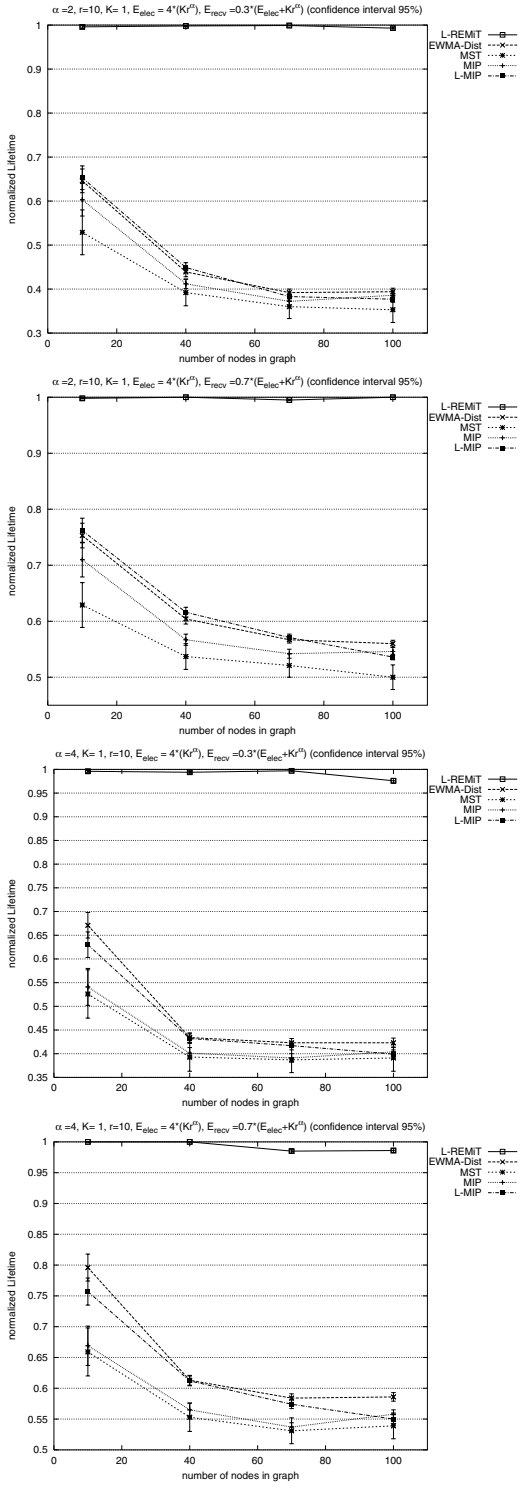

Figure 3. $Normalized\ Life$ (short range radios, 100% nodes are in multicast group and $\alpha = 2$ (above two) and $\alpha = 4$ (below two)).

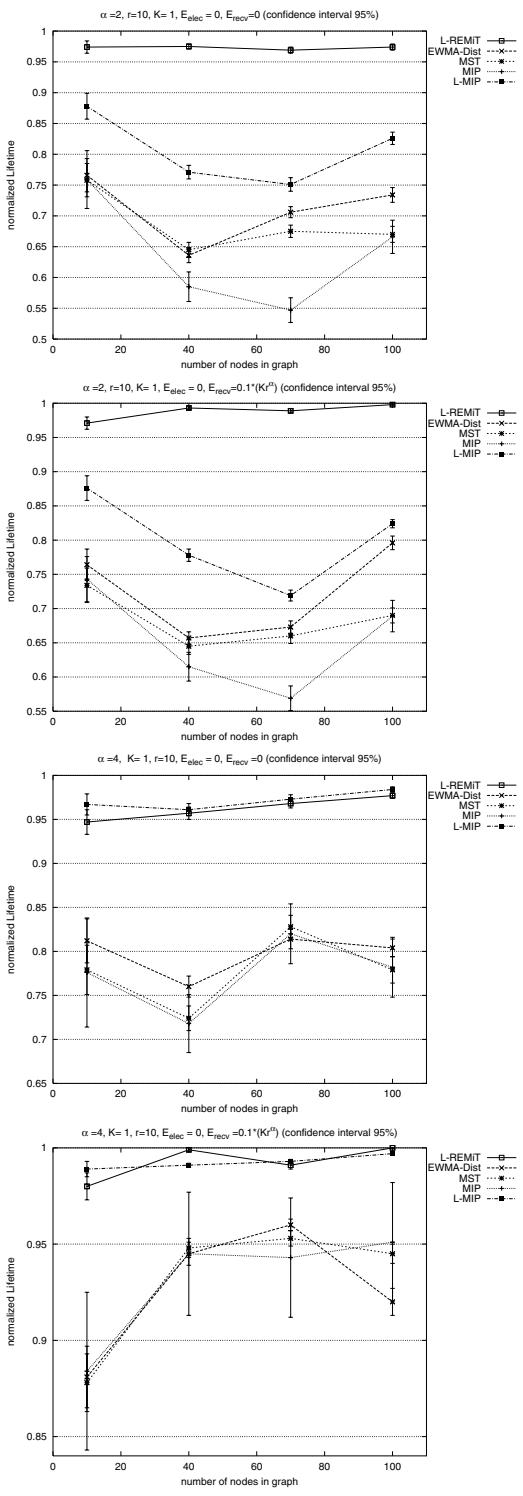

Figure 4. $Normalized\ Life$ (long range radios, 100% nodes are in multicast group and $\alpha = 2$ (above two) and $\alpha = 4$ (below two)).

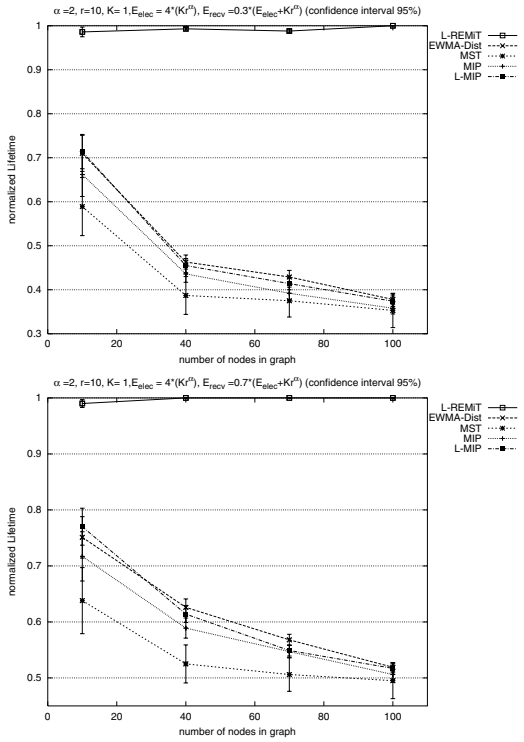

Figure 5. $Normalized\ Life$ (short range radios, 50% nodes are in multicast group and $\alpha = 2$).

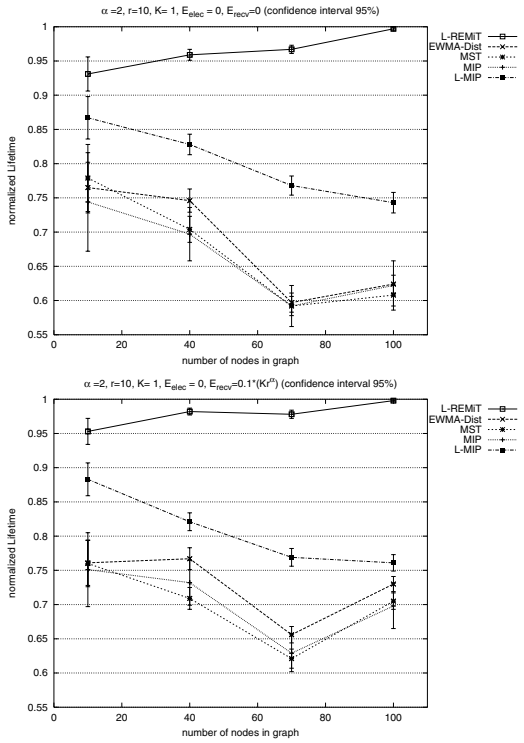

Figure 6. $Normalized\ Life$ (long range radios, 50% nodes are in multicast group and $\alpha = 2$).