G-REMiT: An Algorithm for Building Energy Efficient Multicast Trees in Wireless Ad Hoc Networks

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

Abstract

In this paper, we propose a distributed algorithm called G-REMiT for building an energy efficient multicast tree in a wireless ad hoc network (WANET). G-REMiT employs a more realistic energy consumption model for wireless communication which takes into account not only the energy losses due to radio propagation but also the energy losses in the transceiver electronics. We evaluate the performance of the protocol using two energy consumption model: long range radio and short range radio. We show that for long range radio model, G-REMiT algorithm can achieve better performance than other proposals such as MLU, MLiMST, and MIP, and the energy overhead of for executing G-REMiT is negligible compared with the total energy consumption for the multicast communication. For short range radio, we find that existing energy saving scheme by adjusting node's transmission power is not suitable.

1 Introduction

Recently, attention has been given to wireless ad hoc network (WANET) since the promising civil and military applications [4, 16]. WANET consists of myriad wirelessly connected static devices (such as sensors), and has no infrastructure. All communication among devices in this kind of network are supported by multi-hop transmissions through ad hoc radio connections. In WANET, devices have serious resource (battery) constraints and high vulnerability, they form numerous webs of low-power ad hoc networks to exchange information and/or pursue common goal. So energy efficiency is an important design consideration for WANET [15].

Multicast is an prominent mechanism to communicate information in all kinds of networks, such as overlay network [1]. An overlay network is an abstract network defined over an underlying physical network. Currently, all-wireless overlay networks have attracted considerable consideration due to their potential applications [8]. The base network of all-wireless overlay network is a WANET. Applications in all-wireless overlay network may require many kinds of services from the underlying network, such as group communication, and distributed cache maintenance. Multicast is an important communication primitive for these kinds of services. In this paper, we focus on energy efficient multicasting in WANET.

Multicast tree is an efficient approach to overcome resource wastage in multicasting. There are two kinds of multicast trees: source-based and group-shared [3]. In this paper, we propose G-REMiT: (A protocol for Refining Energy-Efficient Group-shared Multicast Tree). It is a distributed, deadlock-free, asynchronous algorithm, G-REMiT uses a *minimum spanning tree* (MST) as an initial solution and improves the energy efficiency by switching any tree nodes from its connected tree node in multicast tree to another tree node.

The rest of the paper is organized as follows. In Section 2 we summarize some related work. Section 3 describes the system model and some notations used by G-REMiT. The multicast energy cost estimation method employed by each node in the multicast tree is described in Section 4. Section 5 describes G-REMiT algorithm. Section 6 presents the simulation result of G-REMiT and compares it with MIP, MLU, MLiMST algorithms. We conclude with some discussion and future work in Section 7.

2 Related Work

The energy-efficient broadcasting/multicasting tree problem is first presented in [13]. Wieselthier et. al. [13, 14] have proposed a "node-based" elastic model for wireless multicast and the concept of *wireless multicast advantage*. They build a source-based broadcast/multicast

0-7695-1938-5/03 $17.00 © 2003 IEEE

tree by adjusting transmission powers of nodes, followed by elimination of redundant transmissions. Because the problem of constructing the optimal energy-efficient broadcast/multicast tree is NP-hard, several heuristic algorithms for building a source based energy-efficient broadcast/multicast tree are presented. They presented following three source-based broadcast/multicast tree building centralized algorithms: BIP/MIP, BLU/MLU, BLiMST/MLiMST.

Most of the current research has been focused on development of centralized algorithms for construction of energy efficient trees. Centralized algorithm has two limitations: first, a centralized algorithm needs global knowledge which may introduce high communication overhead especially in large scale networks. Second, centralized algorithm needs stopping the data traffic to build a multicast tree and can not adapt to the dynamic nature of WANET. So a distributed algorithm is more suitable for energy efficient multicast tree problem in WANET [14] [2] [12]. In [2], Cagalj et al. have presented an Embedded Wireless Multicast Advantage (EWMA) algorithm to enhance energy efficiency of source-based broadcast tree. They also describe a distributed version of EWMA algorithm. Both G-REMiT and EWMA refine an existing initial tree to an energy-efficient tree. EWMA builds a source-based broadcast tree, but G-REMiT builds an group-shared multicast tree.

3 System Model and Assumptions

We have the following assumptions in our model:

1. Each node in the WANET has only local view of the network.
2. Each node knows the distance between itself and its neighbor nodes using distance estimation schemes such as [9] and [11].
3. We assume that every node can know the number of nodes in the multicast group. For example, group size can be obtained from group membership service.
4. Average of multicast message generation rate (in term of bit/s) at a given node is stable over a period time.

We use wireless communication model in [5] [7]. The connectivity of network depends on the transmission power. Each node can choose its power level p, where $p_{min} \leq p \leq p_{max}$. A node may use a different power levels in each multicast tree in which it participates. Let $p_{i,j}$ be the minimum power needed for link between nodes i and j for a packet transmission. Then,

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

where $r_{i,j}$ is the Euclidean distance between i and j, E_{elec} is a distant-independent constant that accounts for real-world overheads of electronics and digital processing, K is constant dependent upon the properties of the antenna and α is a constant which is dependent on the propagation losses in the medium [14]. We assume $K = 1$ and ignore it in the rest of the paper since it does not affect the results in this paper.

Most of the related work in this area uses the wireless model [13] [14] as:

$$p_{i,j} = K(r_{i,j})^{\alpha}, \qquad (1b)$$

However, this model is not accurate for short range radios, since E_{elec} can substantially exceed the maximum value of the $K(r_{i,j})^{\alpha}$ [10] [5]. Hence E_{elec} cannot be ignored.

Compared to wired networks, WANETs have "wireless multicast advantage" [14] 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.

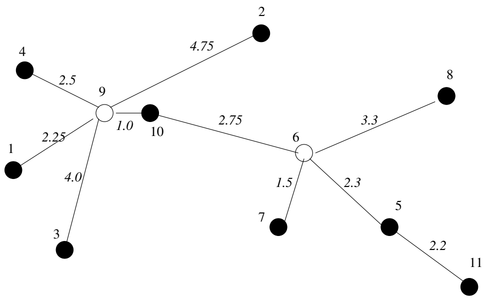

Figure 1. A Multicast Tree. Nodes 1, 2, 3, 4, 5, 7, 8, 10 and 11 are multicast group members. Node 6 and 9 are forwarding nodes. The figure only shows the tree links. There are other links in the network such as link between node 1 and 4 which are not shown for the sake of clarity of the figure and also because G-REMiT ignores non-tree links. Label associated with the edges are Euclidean distances.

In our model, node i has two kinds of coverage area. One is Control coveRage area (CR_i), another is Data coveRage area (DR_i). $DR_i \subseteq CR_i$. Also radius of CR is same for every node. For example, in Figure 1, radius of CR_{10} is 3.2, and the Euclidean distance between node 10 and 7 is 3.1. It means that node 10's control message may reach node 7, but radius of DR_{10} is 2.75, it means that node 10's data message may only reach node 6.

In this paper we use a **Core Based Tree (CBT)** [3] (CBT is a typically kind of group-shared tree for multicasting) i.e. the multicast tree spans all the multicast receivers and senders. Furthermore we assume for simplicity that the set of senders is a subset of the set of receivers, i.e., the multicast group is a **closed group**. A node's neighbor which also belongs to the multicast tree is referred to as a **tree neighbor**. A **connected-tree neighbor** of a node is a tree neighbor which is connected to the node by a *tree link*. A node that has more than one connected-tree neighbor is called a **branch node**. A multicast message is forwarded along the tree links starting from a source node: every node on the tree which receives a new multicast message forwards it to all its tree neighbors except the one from which it received the message.

4 Multicast Energy Cost Estimation

The problem of computing a minimum-cost multicast tree in a wired network can be modeled as a Steiner tree problem which is a NP-hard problem. In Ad hoc networks, this problem remains NP-hard. So a heuristic algorithm is needed to build energy-efficient multicast trees efficiently.

In our proposed heuristic algorithm, every node has its own local view which includes its neighbor's information. Based on the local view, it can estimate the current energy consumption cost of its neighbors and also the energy consumption cost of its neighbors, if the node switches its connected tree neighbor to another node. Using these estimations, the node can know what is the difference of energy cost at these nodes if it switch its connected tree neighbor to another. In other words, a node can switch its connected tree neighbor to achieve a lower energy consumption based on node's local information.

4.1 Energy Cost of a Node in a Multicast Session

Because we use a group shared tree, a multicast message can arrive at a node from several different links. An important point to note is that *the energy cost at every tree node is not only decided by the distance between connected tree neighbors but also decided by where the message is coming from.* Since, the energy cost at a tree node is proportional to the distance to the farthest tree neighbor ignoring the neighbor from which it received the multicast message.

For example, consider the multicast tree shown in Figure 1, in which nodes 1, 2, 3, 4, 5, 7, 8, 10 and 11 are multicast group nodes and nodes 9 and 6 are forwarding nodes. Suppose node 2 is a multicast source. Then in node 2's multicast session, it will send the multicast message along the tree link to node 9. Node 9 will forward the multicast message to nodes 1,3,4, and 10. Similarly, node 10 will forward it to node 6 and so on. The energy consumed at node 9 in this multicast session is $max(p_{9,1}, p_{9,3}, p_{9,4}, p_{9,10}) = p_{9,3}$. However, if we consider node 10's multicast session, the energy consumed at node 9 in node 10's multicast session will be $max(p_{9,1}, p_{9,2}, p_{9,3}, p_{9,4}) = p_{9,2}$, since node 9 now receives the multicast message from link $(10, 9)$ and has to transmit at a greater power level so as to reach node 2.

In this paper, we assume that the energy cost for a link layer broadcast message to a receiver is negligible compared to the transmission cost to the sender based on the data of real experiments[1] [5]. So we only consider the energy cost for transmitting a packet as the energy cost at a node. Let i be a tree node, nodes $i_1, i_2, \ldots, i_m$ be its connected tree neighbors in the multicast tree, and assume that the multicast message is forwarded by node i_k to node i. Let $L_i[j]$ be the j-th longest tree link among all of the tree links which are incident on node i and let $d_i[j]$ be the length of $L_i[j]$. We calculate EC_i, the energy cost of node i for a multicast session, as follows:

if i_k is the other end point of $L_i[1]$
then $EC_i = E_{elec} + (d_i[2])^\alpha$
else $EC_i = E_{elec} + (d_i[1])^\alpha$. (2)

So in one multicast session, $EC_i = E_{elec} + (d_i[1])^\alpha or E_{elec} + (d_i[2])^\alpha$ as determined by i_k.

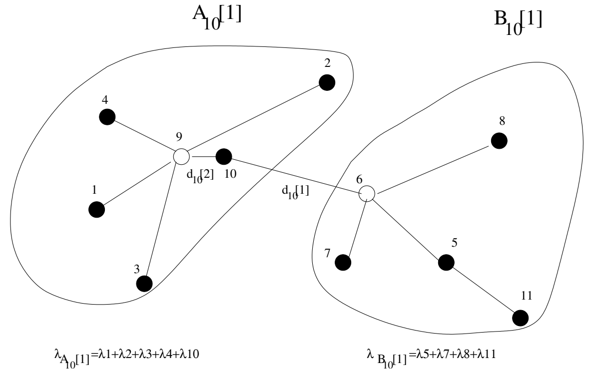

Figure 2. Energy Consumed by Node 10 per unit time is $P_{10} = \lambda_{B_{10}[1]}(E_{elec} + (d_{10}[2])^\alpha) + \lambda_{A_{10}[1]}(E_{elec} + (d_{10}[1])^\alpha)$; Because $B_{10}[1]$ = 5,7,8,11, $w_{10}[1]$ = 4 ($w_{10}[1]$ is the number of group nodes in sub-tree $B_{10}[1]$).

4.2 Energy Consumption Metric of a Node

Next lets determine how much energy is consumed at a node per time unit. For example consider node 10 in Figure 2. Because node 6 and 9 are only forwarding nodes

---

[1] In [5], Feeney and Nilsson have empirically determined that for Lucent IEEE 802.11 11Mbps WaveLan Card the energy cost of a broadcast reception is $0.26 \times$ size $+ 50$ μJ and energy cost for a broadcast send is $2.1 \times$ size $+ 272$ μJ, where size is the length of the packet in bytes. Hence a link layer broadcast receiver's energy cost for receiving a packet is at most 20% of the transmission energy cost for the sender.

of the multicast tree, there are 9 different kinds of multicast sessions whose are node 1,2,3,4,5,7,8,10,11. Obviously, node 10 will send/forward multicast message with $p_{9,10} = E_{elec} + (d_{10}[2])^\alpha$ in multicast session of node 5,7,8 and 11. But node 10 will send/forward multicast message with $p_{6,10} = E_{elec} + (d_{10}[1])^\alpha$ in multicast session of node 1,2,3,4,10. So energy consumed at node 10 per time unit is $P_{10} = \lambda_{B_{10}[1]}(E_{elec} + (d_{10}[2])^\alpha) + \lambda_{A_{10}[1]}(E_{elec} + (d_{10}[1])^\alpha)$. Note that when link (10,6) is deleted from the multicast tree two subtrees are formed $A_{10}[1]$ and $B_{10}[1]$ as show in Figure 2.

In general, let $A_i[j]$ and $B_i[j]$ be the two subtrees created as a result of deletion of link $L_i[j]$ from the multicast tree. Without loss of generality, let i be located in $A_i[j]$. Further, let λ_i be the message generation rate of node i and λ_{sum} be the aggregate message generation rate over all the nodes. We define the sum of multicast message generation rate of group nodes in $A_i[j]$ to be $\lambda_{A_i}[j] = \sum_{k \in A_i[j]} \lambda_k$ and the sum of multicast message generation rate of group nodes in $B_i[j]$ to be $\lambda_{B_i[j]} = \lambda_{sum} - \lambda_{A_i}[j] = \sum_{k \in B_i[j]} \lambda_k$. The energy consumed of node i per unit time (i's energy consumption metric) is

$$\begin{aligned} P_i &= \lambda_{B_i[1]}(E_{elec} + (d_i[2])^\alpha) + \lambda_{A_i[1]}(E_{elec} + (d_i[1])^\alpha) \\ &= \lambda_{B_i[1]}(d_i[2])^\alpha + \lambda_{A_i[1]}(d_i[1])^\alpha + \lambda_{sum} E_{elec}. \quad (3) \end{aligned}$$

We use function h to denote r.h.s. of Formula (3) i.e. $P_i = h(\lambda_{A_i[1]}, \lambda_{B_i[1]}, d_i[2], d_i[1], E_{elec})$.

If we introduce $\bar{\lambda}$, we can simplify formula (3) as following:

$$P_i \approx \bar{\lambda}[w_i[1](d_i[2])^\alpha + (|G| - w_i[1])(d_i[1])^\alpha + |G| E_{elec}].$$

where $w_i[j]$ is the number of group nodes in $B_i[j]$ and G is the set of multicast group nodes.

Because $\bar{\lambda}$ is an independent variable of node i, by deleting variable $\bar{\lambda}$, relative energy consumed at node i per time unit (i's relative energy consumption metric) is

$$R_i = w_i[1](d_i[2])^\alpha + (|G| - w_i[1])(d_i[1])^\alpha + |G| E_{elec}. \quad (4)$$

5 G-REMiT Protocol Details

5.1 Criterion for Switching

We use link-based *minimum weight spanning tree* (MST) as the initial tree because MST perform quite well even as a final solution of our problem, which can be seen from the simulation results in section 6. Then we improve the initial tree by exchanging some existing branches in the initial tree for new branches so that the total energy cost of the tree is lower. We call the difference of total energy cost of the trees before and after the branch exchange an *gain*. In our heuristic, the notion of *gain* is used as the criterion for the changing of branches.

To motivate G-REMiT we consider how node 2 in Figure 1 decides to change its branch node, node 9, to node 6. We refer to this change event as $Change_2^{9,6}$. We assume $\alpha = 2, E_{elec} = 0$.

Node 2 will estimate the change in the energy consumption at node 2, 9, and 6 if it makes $Change_2^{9,6}$. First, node 2 will estimate the current relative energy consumed at node 2, 6 and 9, i.e. R_2, R_6 and R_9 respectively, by using Formula (4):

$R_2 = r_{9,2}^2 = 22.59; R_6 = [r_{6,10}^2 + 8r_{6,8}^2] = 94.68; R_9 = [r_{9,3}^2 + 8r_{9,2}^2] = 191.43$.

Then node 2 can estimate the new energy consumed at node 2,9,6 after $Change_2^{9,6}$, i.e R'_2, R'_6 and R'_9 respectively: $R'_2 = r_{2,6}^2 = 12.96; R'_6 = [r_{6,2}^2 + 8r_{6,8}^2] = 100.08; R'_9 = [r_{9,4}^2 + 8r_{9,3}^2] = 93.33$.

The *gain* ($g_2^{9,6}$) obtained by switching at node 2 from node 9 to node 6 is:

$$g_2^{9,6} = (R_2 + R_9 + R_6) - (R'_2 + R'_9 + R'_6) = 102.33;$$

Likewise node 2 can compute the gain in energy cost if it switches to node 10 and node 8:

$$g_2^{9,10} = (R_2 + R_9 + R_{10}) - (R'_2 + R'_9 + R'_{10}) = 10.89$$

$$g_2^{9,8} = (R_2 + R_9 + R_8) - (R'_2 + R'_9 + R'_8) = 44.82$$

Having the *gains* from node 2's view, our algorithm selects a node with the highest positive gain as the new connected tree neighbor. Thus node 6 is selected as the new connected tree neighbor of node 2. Node 2 will select node 6 as its parent node in the multicast tree and disconnect from node 9. So in Figure 1, tree link between nodes 2 and 9 will be deleted, and tree link between nodes 2 and 6 will be added to the multicast tree. Because DR_9 does not need cover node 2 any more, radius of DR_9 will decrease to $r_{9,3}$. DR_6 should be increased to cover node 2, hence radius of DR_6 will increase to $r_{6,2}$.

5.2 Some Basic Results

Because G-REMiT algorithm is a distributed algorithm, let us first introduce some notations for node's local view. We denote the set of tree nodes that are within CR_i as V_i. Let node j be a tree neighbor of i, that is, $j \in V_i$. We denote the set of nodes that are connected to node i through a tree link as C_i. The nodes which are in V_i but are not connected to node i through a tree link are said to be non-connected tree neighbors of i. We denote this set as N_i ($N_i = V_i - C_i$). We denote the sextuple ($d_i[1]$, $d_i[2]$, $d_i[3]$, δ_i, $w_i[1]$, $w_i[2]$), where δ_i is degree of node i in the multicast tree, as l_i. We use $Change_i^{x,j}$ to refer to the changes in the multicast tree caused by node i breaking the tree link to its connected tree neighbor x and forming a tree link to one of its non-connected tree neighbor, node j (see Figure 3).

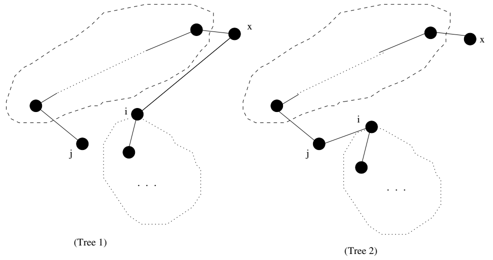

Figure 3. Before and After $Change_i^{x,j}$.

We present the following Lemmas without proof:

Lemma 1 *Nodes that are on $path_{j,i}$ are the only nodes' energy cost in the multicast tree that may be affected by $Change_i^{x,j}$.*

Lemma 2 *If i is not on $path_{j,x}$, the tree remains connected after $Change_i^{x,j}$.*

Informally, in Figure 3, if i is on $path_{j,x}$, tree will be disconnected.

5.3 Algorithm Description

Now we describe our distributed G-REMiT algorithm. Our algorithm is divided into two phases. In the first phase, all nodes run a distributed algorithm proposed by Gallager et al. [6] to build a MST tree. We require that at the end of the first phase, node i ($i \in N$, where N is set of all nodes in the multicast tree) has all local information l_k for all nodes $k \in V_i$. Node i obtains its l_i by piggybacking information on regular messages. Nodes obtain l_k by hearing k's one-hop local broadcasting within CR_k.

In the second phase, the difficulty in this distributed environment is when and how to terminate the refinement. We organize the second phase in rounds. Each round of the second phase is led by the core node z. z will terminate G-REMiT algorithm when there is no energy $gain$s by switching any node the tree. In each round, a Depth-First Search (DFS) algorithm is used to pass G-REMiT token to the nodes one by one. The G-REMiT token gives the permission to a node to do refinement. Only the node which has the G-REMiT token can do refinement. Other nodes can only respond to the requesting node.

When i obtains the G-REMiT token, it does the following steps to refine the tree. We use R_i', R_x' and R_{i_j}' to denote the energy cost at i, j, x after $Change_i^{x,j}$, respectively. (see Figure 6 for illustrations of these steps):

1. **Branch node x selection:** Node i selects a branch node x according to the following order: the other end point of $L_i[1]$, $L_i[2]$, and $L_i[3]$, followed by other connected nodes in C_i. If there is no such node x available, go to step 6.
2. **New Branch node j selection:** Select a new branch node candidate j with the highest positive $gain$ ($g_i^{x,j} := (R_i + R_x + R_j) - (R_i' + R_x' + R_j')$), which will not result in tree disconnection if node i makes $Change_i^{x,j}$. If there is no such node j available, then go to step 6.
3. **$Path_{j,i}$ Exploring:** Node i sends $Path\_Exploring(path\_gain)$ message along $path_{j,i}$. Every node on the $path_{j,i}$ may change $path\_gain$ value if its longest link is on $path_{j,i}$ ($Path\_Exploring(path\_gain)$ is forwarded hop-by-hop along $path_{j,i}$). When node i gets back $Path\_Exploring$, it will check $path\_gain$. If $path\_gain$ is positive, node i executes the next step. Otherwise, it goes back to first step to select another node x.
4. **Make $Change_i^{x,j}$:** Node i makes $Change_i^{x,j}$ by $Join\_REQ$ and a $Join\_REP$ negotiation with node j. Node j will send $Join\_REP$ back to node i. If node i gets $Join\_REP$ message, it will change C_i and N_i, send $leave$ message to node x and go to next step. Otherwise, it will go back to step 2 to select a another new branch node j.
5. **$Path_{i,x}$ Updating and V_i Notification:** Node i sends $Path\_Updating$ along $path_{x,i}$ to update l_k ($k \in path_{x,i}$). Also node i will local broadcast to nodes in V_i about the $Change_i^{x,j}$.
6. **Token Passing:** Node i will pass the token according to the DFS algorithm.

Following are two examples which illustrate the second phase of G-REMiT algorithm: 1) single refinement at a node and 2) the process of refinements at all the nodes in the multicast tree.

Example 1: This example illustrates one refinement at one node. Node 2 gets the G-REMiT token, node 2 does the following:

1. Node 2 select node 9 as node x.
2. Node 2 calculates $gains$ as explained previously in the paper and finds out $g_2^{9,6}$ is the highest positive value.
3. Node 2 sends $Path\_Exploring(path\_gain)$ along $path_{6,2}$. Node 6 will forward the message to node 10. Node 10 will change $path\_gain$ because $L_{10}[1]$ is on $path_{6,2}$. Then node 10 forwards the message to node 9. Node 9 will forward the message to node 2. Node 2 will find $path\_gain$ is still positive after $Change_2^{9,6}$.

4. Node 2 now sends $Join\_REQ$ to node 6. When node 6 responds to node 2 with $Join\_REP$ message, node 2 will moves node 6 from N_2 to C_2 and it will send $leave$ message to node 9. Then it will remove node 9 from C_2 to add it to N_2.

5. Node 2 sends $Path\_Updating$ along $path_{9,2}$ to update l_{10}. Then node 2 will send $Local - Updating$ to nodes in V_2 about $Change_2^{9,6}$.

6. Finally, Node 2 will pass the token to node 9 according to DFS algorithm.

☐

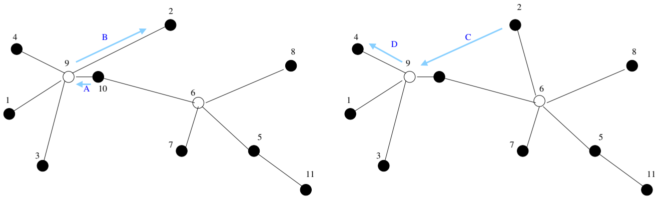

Figure 4. Snapshot after step A) and B) of G-REMiT; Snapshot after step C) and D) of G-REMiT

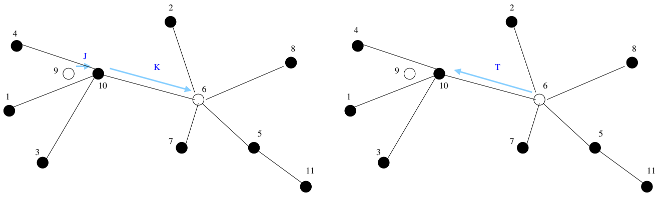

Figure 5. Snapshot after step J) to K) of G-REMiT; Snapshot after step T) of G-REMiT

Example 2: In Figure 1, if the core node of multicast tree is node 10. Following are the steps happened in the multicast tree for one round G-REMiT.

A) Node 10 selects node 9 and passes the G-REMiT token to node 9 as Figure 4.

B) Node 9 gets the token, it does the refinement which is similar to example 1. After the refinement, no branch node switching happens at node 9. Then node 9 selects node 2 and passes the G-REMiT token to node 2 as Figure 4.

C) Node 2 gets the token, it does the refinement as the example 1. After the refinement, node 2 makes $Change_2^{9,6}$. Because node 2 has no children, node 2 passes the G-REMiT token back to node 9 (node 2's parent before refinement). Figure 4 illustrates the tree after $Change_2^{9,6}$ and token passed back from node 2 to node 9.

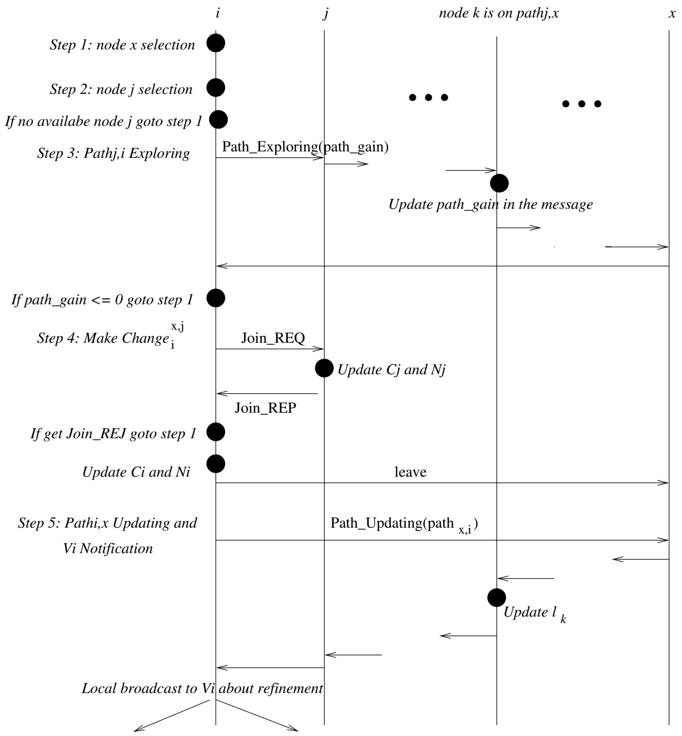

Figure 6. Overview of node i's behavior in second phase of G-REMiT algorithm

D) Node 9 gets the token. Node 9 selects node 4 and passes the G-REMiT token to node 4 as Figure 4.

E–I) Similar to step B) to C), node 1,3,4 do refinement and pass token back to node 9 respectively. Figure 5 gives a snapshot of the multicast after these refinements.

J) Node 9 gets back G-REMiT token and finds that all of its children nodes have been refined. And node 9 has no children anymore. Because node 9 is not a group member. Node 9 will leave the multicast tree. Then node 9 passes the G-REMiT token back to its original parent node – node 10. Figure 5 shows the multicast tree until step J).

K) Node 10 gets back the G-REMiT token, it passes G-REMiT token to a node 6 because node 6 is the only non-refined children as Figure 5.

L–S) Similar to step B) to C), node 5,6,7,8,11 do refinement. Figure 5 gives a snapshot of the multicast after these refinements.

T) Node 10 gets back G-REMiT token from node 6, node 10 terminates G-REMiT algorithm in this round. Figure 5 illustrates the finial multicast after this round refinement.

☐

5.4 Complexity of G-REMiT algorithm

The message complexity of each node switching is $O(H)$. Hence the message complexity of one round in which each node performs at most one connected branch nodes switching is $O(HN\delta_{max})$, where N is the number of nodes in the tree, and δ_{max} is the maximum number of neighbor any node has in the tree. The computational complexity of one BN switching is $O(\delta_{max})$. Therefore the computational complexity of one round is $O(N\delta_{max}))$. The space complexity of G-REMiT for each node is $O(\delta_{max})$ since the size of V is $O(\delta_{max})$.

6 Performance Evaluation

We used simulations to evaluate the performance of G-REMiT algorithms. The goal of simulations was to evaluate the effectiveness of G-REMiT algorithm in building the energy-efficient multicast tree. We compare our algorithm with MLU,MLiMST,MIP algorithms [14]. Graphs with a specified number of nodes (10, 20, 40, 60, 80, 100, and 200) are randomly generated. Every node is within the maximum transmission range (r) of at least one other node in the network. In other words, the graph is connected. We assume that the network bandwidth is 50kb/s,average of multicast message generation rate of every node is randomly generated as 0kb/s, 1kbs/, ..., 50kb/s. We use free space propagation model, so $\alpha = 2$. We also use two different E_{elec} values to represent the long range radio and short range radio. Because E_{elec} is hardware dependent, by analyzing the experimental data in [10] [5], we decide to use $E_{elec} = 0.1r^2_{max}$ to represent long range radio and $E_{elec} = 4r^2_{max}$ to represent short range radio.

6.1 Performance Metrics

We use TPC (Total Power Consumed) to denote the total energy consumed per unit time for all the nodes in the multicast tree. Using Formulas (3), we get the following formula:

$$\begin{aligned} TPC &= \sum_{i=1}^{N} P_i \\ &= \sum_{i=1}^{N} h(\lambda_{A_i[1]}, \lambda_{B_i[1]}, d_i[2], d_i[1], E_{elec}). \quad (5) \end{aligned}$$

where N is the number of nodes in the trees.

For a given graph o and multicast group G, let the four kinds algorithms be denoted by T_{alg}, where $alg \in A = \{$G-REMiT, MLU, MLiMST, or MIP$\}$. We use $TPC(T)$ to denote the TPC value of a tree T. We use TPC of multicast tree to define the performance metric: $Normalized$ TPC with algorithm alg is:

$$\frac{TPC_{alg}}{TPC_{best}},$$

where $TPC_{best} = \min(TPC(T_{alg})), alg \in A$. Using this metric, we can compare see how close the multicast trees obtained from G-REMiT, MIP, MLU, MLiMST are in terms of their TPC value.

6.2 Simulation Results For G-REMiT

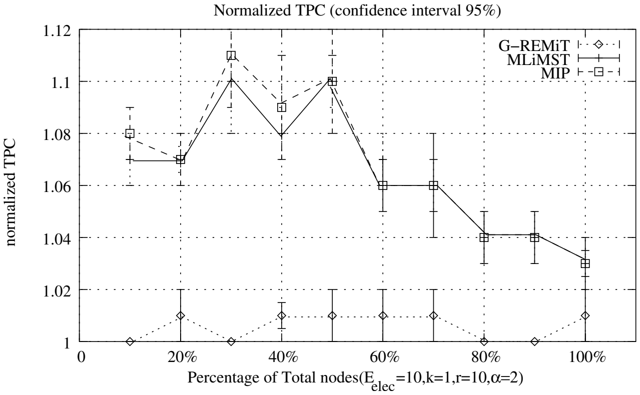

Figure 7. $Normalized\ TPC$ **for a graph with 100 nodes.**

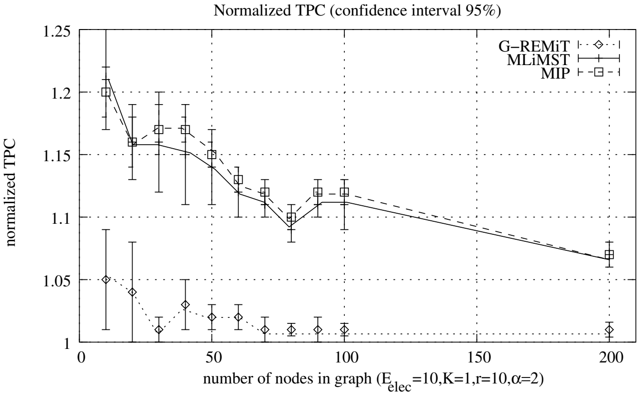

Figure 8. $Normalized\ TPC$ **when** 50% **nodes are multicast group nodes.**

The performance of algorithms for long range radio are shown in Figure 7 and 8. In the figures, we can see the $normalized TPC$ achieved by the algorithms on networks of different sizes and group sizes respectively. To make the figure clear to read, we do not show the performance of MLU algorithm in the figures because the $normalized$ TPC of MLU algorithm is around 1.7-2 which is the worst among all of the other algorithm. The figures show that

G-REMiT algorithm has the better performance than MIP, MLiMST algorithm. We also consider energy overhead of G-REMiT itself in the simulation. We find that the energy overhead of G-REMiT is always below 1.5% of TPC of the multicast tree.

For short range radio, we also did some simulations. Based on experiments, we found that all algorithms has very similar performance in short range radios. The main reason for such behavior is that the node's E_{elec} is substantially greater than the maximum value of Kr^{α}. So the energy saving by reducing data transmission range is very little. Consequently, the simplest and reasonable approach for short range radio is that every node tries to cover as many nodes as possible.

Based on the simulation results, we find that using *wireless multicast advantage* [14] G-REMiT has better performance than MIP,MLU, MLiMST. Further these three algorithms are centralized algorithms. On the other hand, G-REMiT is distributed algorithm, so we think it is more suitable for WANETs.

7 Conclusions

In this paper, we first presented an energy consumption metric suitable for evaluating energy-efficiency of multicast protocols in WANET. Using the novel observation that minimum energy-cost for forwarding a multicast message to a node's neighbor in the multicast tree is dependent upon the link on which the message arrives at the node, we proposed the G-REMiT algorithm to construct an energy-efficient multicast tree. Our simulation results show that the G-REMiT algorithm performs much better than most existing energy efficient multicast algorithms. Moreover, G-REMiT is a distributed algorithm, a feature that other authors have believed to be both necessary and challenging.

In future, we plan to explore energy efficient multicasting problem of short-range radios. We also plan to extend G-REMiT algorithm to mobile ad hoc networks and study its performance.

8 Acknowledgments

This work is supported in part by NSF grants ANI-0123980 and ANI-0196156.

References

[1] K. Birman. Technology challenges for virtual overlay networks. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART A: SYSTEMS AND HUMANS*, 31(4):319–327, July 2001.

[2] M. Cagalj, J. Hubaux, and C. Enz. Minimum-energy broadcast in all-wireless networks: Np-completeness and distribution issues. In *Proceedings of ACM MobiCom 2002*, pages 172–182, Atlanta, Georgia, Sept. 2002.

[3] C.-C. Chiang, M. Gerla, and L. Zhang. Adaptive shared tree multicast in mobile wireless networks. In *Proceedings of IEEE Globecom'98*, pages 1817–1822, Sydney, Australia, Nov. 1998.

[4] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of ACM MobiCom 1999*, pages 263–270, Seattle, Washington, Aug. 1999.

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

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

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pages 3005–3014, Maui, Hawaii, Jan. 2000.

[8] R. H. Katz and E. A. Brewer. The case for wireless overlay networks. In *Proceedings of SPIE Conference on Multimedia and Networking (MMNC96)*, pages 77–88, San Jose,CA, Jan. 1996.

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

[10] R. Min and A. Chandrakasan. Top five myths about the energy consumption of wireless communication. *ACM Sigmobile Mobile Communication and Communications Review (MC2R)*, 6(4), Oct. 2002.

[11] R. Steele. *Mobile Radio Communications*. PentechPress, 1992.

[12] P.-J. Wan, G. Calinescu, and O.-Y. Li. Minimum-energy broadcast routing in static ad hoc wireless networks. In *Proceedings of the IEEE INFOCOM 2001*, pages 1161–1171, Anchorage, Alaska, Apr. 2001.

[13] J. Wieselthier, G. Nguyen, and A. Ephremides. Algorithms for energy-efficient multicasting in ad hoc wireless networks. In *Military Communications Conference Proceedings*, volume 2, pages 1414–1418, 1999.

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

[15] J. Wieselthier, G. Nguyen, and A. Ephremides. Resource-limited energy-efficient wireless multicast of session traffic. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pages 3460–3469, Maui, Hawaii, Jan. 2001.

[16] F. Ye, H. Y. Luo, J. Cheng, S. W. Lu, and L. X. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of ACM MobiCom 2002*, pages 148–159, Atlanta, Georgia, Sept. 2002.