Multicast Routing with Minimum Energy Cost in Ad Hoc Wireless Networks

Lok Ranjan Das
Dept of Computer Science
Disha Institute of Management And Technology, Raipur
CSVTU, Bhilai (Chhattisgarh)

Abhishek Pandey
Dept of Computer Science
Disha Institute of Management And Technology, Raipur
CSVTU, Bhilai (Chhattisgarh)

ABSTRACT
In this paper, we discuss the energy efficient multicast problem in ad hoc wireless networks. The problem of our concern is: given an ad hoc wireless network and a multicast request, to find a multicast tree such that the total energy cost of the multicast tree is minimized. Each node in the network is assumed to have a fixed level of transmission power. We first prove the problem is NP-hard, and then propose three heuristic algorithms, namely Steiner tree based heuristic, Node-Join-Tree and Tree-Join-Tree greedy algorithms. Extensive simulations have been conducted and the results have demonstrated the efficiency of the proposed algorithms.

General Terms
Wireless ad hoc network, Multicast routing, Simulation.

Keywords
Minimum energy cost, Multicast tree, Total energy cost, Transmission range, Greedy algorithm.

1. INTRODUCTION
Wireless ad hoc networks have received significant attention in recent years due to their potential applications in battlefield, emergency disaster relief and etc. Wireless ad hoc network consists of a collection of mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure. In such a network, each mobile node can serve as a router. A communication session is achieved either through a single-hop transmission if the communication parties are close enough, or through relaying by intermediate nodes otherwise. Energy efficiency is an important issue in ad hoc networks, where mobile nodes are powered by batteries that may not be possible to be recharged or replaced during a mission. The limited battery lifetime imposes a constraint on the network performance. In order to maximize the network lifetime, ideally, the traffic should be routed in such a way that the energy consumption is minimized.

In this paper, we address the problem of multicast routing in ad hoc wireless networks. Multicast is a communication means that a source node sends messages to a group of destinations. Multicast routing is to find a multicast tree, which is rooted from the source and spans all destination nodes. In ad hoc networks, nodes communicate with each other via radio signals, which are broadcast in nature. When unidirectional antennas are used, every transmission by a node can be received by all nodes within its transmission range. If there are multiple destination nodes in the transmission range of a node, a single transmission can reach all these destinations. Since in a multicast tree, only the non-leaf nodes need to transmit messages further down to the destinations and all the leaf nodes are destinations that only receive multicast messages, the energy cost of a multicast tree is the sum of energy cost of all the non-leaf nodes in the tree. We assume the reception of signals cost no extra energy.

There has been a lot work on energy efficient broadcast routing in ad hoc networks [5 – 12]. But, there is little work report specifically on multicast routing. Two typical works [5, 8] on multicast routing are the extensions of their broadcast routing algorithms, where a multicast tree is obtained from a broadcast tree by pruning the branches that do not contain the multicast destinations. The multicast tree, obtained by using this method, may contain many unnecessary intermediate nodes if the destinations are located close to the leaf nodes of the broadcast tree, which makes the multicast tree inefficient in both energy and bandwidth consumption. In this paper, we propose three algorithms that are specially designed for multicast operations. The problem of our concern is: given an ad hoc wireless network where each node has a fixed transmission power and a multicast request, to find a multicast tree such that the total energy cost of the multicast tree is minimized.

2. NETWORK MODEL AND PROBLEM SPECIFICATION
The network is modeled by a directed graph G = (V, A), where V represents the set of wireless nodes and A represents the set of arcs in the network. For each node $v \in V$, transmission power p (v) is given. For any two nodes v1 and v2, if node v2 is in the transmission power range of node v1 (i.e., $d^{\alpha}(v1, v2) \le p(v1)$, α is a constant value between 2 ~ 4), then there is an arc $(v1, v2) \in A$ (i.e., a directed link from node v1 to node v2).

Suppose, (s, D) is a given multicast request, where s is a source and D a set of destinations. Let T be a multicast directed tree sourced from s. There are two kinds of nodes in T: the nodes that need to transmit / relay multicast messages for s, and the nodes that only receive multicast messages. The nodes that receive messages only are the leaf-nodes in T. We assume only the nodes that transmit messages consume energy for a multicast. That is, the nodes that only receive messages are assumed to incur no energy cost for a multicast. Let NL (T) denote the set of non-leaf nodes of T. The total energy cost C (T) of T can be represented as:

$$C(T) = \sum_{v \in NL(T)} p(v)$$

Our problem is: given a multicast request (s, D) and p(v) for each node v, to find a multicast tree rooted from s and spanning all nodes in D such that total energy cost of the tree is minimized. We call this problem Minimum Energy Multicast (MEM for short) problem. We assume the locations of nodes are static or change slowly. Node mobility is not considered in this paper. Ad hoc networks are quite different from the wired networks due to the nature of wireless communication and the lack of infrastructure support. They pose many new challenges that are never seen in wired or cellular networks; even the mobility is not addressed.

3. ALGORITHMS

We first prove the MEM problem is NP-hard.

3.1 The MEM problem is NP-hard.

Proof. The minimum set cover problem, which is NP-hard [13], is polynomial time reducible to the MEM problem. The details of the proof are omitted.

The next, we propose three heuristic algorithms to tackle the MEM problem. The first one is based on the directed Steiner tree. The other two are greedy algorithms.

3.1.1 Transforming to Directed Steiner Tree Problem

The MEM problem is to find a multicast tree such that the total energy cost of those transmitting nodes in the tree is minimized. In our network model, we assign the transmission power of a node as the weight of it. We first transform the network graph G to a new graph G' that has weight on arcs.

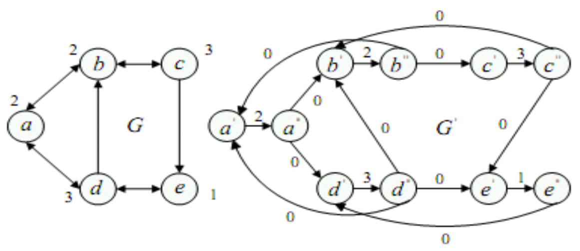

Fig. 1. The transformation from *G* to *G'*

For a multicast request (s, D), we denote D' = {v'| v ∈ D}. The MEM problem in G can be transformed to the following problem in G' : find a directed Steiner tree T rooted from s' and including all nodes of D' in G' such that sum of weights of arcs in T is minimized.

This is a typical directed Steiner tree problem. The directed Steiner tree problem has been well-studied and several heuristics have been proposed [14]. Any heuristic of the directed Steiner tree problem can be used to find the solution to the MEM problem.

3.1.2 A Node-Join-Tree Greedy Algorithm

We introduce three sets. One is cover-set *C*, containing the nodes which transmit/relay message. The second is uncovered-set *U*, containing the nodes that are not covered so far. The third is candidate-set *S*, containing the candidates to be selected as the next transmit/relay node. This heuristic starts from *s*. Initially, *C* contains only *s* and *U* is assigned to *D*. All outgoing neighbors of *s* are added to *S* and are removed from *U* if any of them are in *U*. The next, a node in *S* is selected to be included in *C* (the selection criteria is explained below), and its outgoing neighbors are added into *S* and also removed from *U*. This operation is repeated until *U* becomes empty.

In order to choose the nodes into cover-set such that the total energy cost of the multicast tree is minimized, we introduce the following function:

$$|V_i \cap U| / p(v_i)\text{, where } V_i=\{v_j \mid (v_i, v_j) \in A\}.$$

Notice that V_i is a set of outgoing neighbors of node v_i, $v_i \notin V_i$, and U a set of nodes which are not covered so far.

This function represents the number of nodes that a node can cover per energy unit. Each time, a node with the maximum value of this function will be selected into the cover-set. By doing so, the total energy cost of the multicast tree can be made as small as possible.
In order to guide the growth of the tree towards the destinations when there is no node in *S* that covers any node in *U* (i.e., there is no uncovered destination in the set of outgoing neighbors of any node in *S*, we select a node that is in a shortest path from *s* to some nodes in *U*.

The details of the Node-Join-Tree greedy algorithm are as the following:

Input G= (V, A) and a multicast request (s, D)
Output T: a multicast tree for (s, D)
C = {s}; //C: the cover-set
U = D −V_S; //U: the uncovered-set
S = V_S; //S: the candidate set
While ($U \neq \emptyset$) **do**
Choose $v_i \in S-C$ such that max ($|V_i \cap U| / p(v_i)$)
C = C∪ {v_i};
U= U-V_i;
S=S∪ V_i;
Construct the multicast tree T from C.

3.2 Given a request (s, D) in G (V, A), the node-join-tree greedy algorithm can output a multicast tree in time O(n^2).

Proof. It is easy to know that the greedy algorithm can output a multicast tree. In the while-loop, there is at most n loop and for each loop, finding max value takes O(n), then the while-loop can finish in the time of O(n^2), and the construction of a multicast tree in the last line takes the time of O($|V|^2$)=O(n^2). Therefore, the whole algorithm ends in the time O (n^2 + n^2) =O (n^2).

3.2.1 A Tree-Join-Tree Greedy Algorithm

In the above greedy heuristic, the algorithm constructs the multicast tree *T* in a top-down fashion, starting from the source node. In contrast, the following heuristic takes a global approach, starting from any node in the network, to construct a multicast tree that has efficient energy cost.

Let *s* be the multicast source and *D* the destination set, where $|D| = k$. The basic idea of this algorithm is as follows. Initially, each node in *D* is a subtree. Each time, we find a node $v \in V$ that uses the least amount of energy to link the roots of at least two subtrees and merge them into a bigger one, where *v* becomes the root of the newly merged subtree. A subtree is directed. All the leaf-nodes of a subtree are nodes in *D* and they can be reached from the root along the subtree. Repeat this merging operation until all subtrees form into a single tree where *s* is the root, which is the multicast tree. We need some notations before getting into details of the algorithm.

Def 1. Orphan-subtree. An orphan subtree (orphan for short) is a subtree whose root is not s.

Def 2. Merged-tree (v, S), where S is a set of subtrees and v∈ V. A merged tree, denoted by merged-tree(v, S), is the shortest path tree (in terms of energy cost) from v to reach the roots of all the subtrees in S except the one that contains v (if v is in one of the subtrees in S). Notice that the root of the tree merged-tree (v, S) is v if v is not in any of the subtrees in S, or is the root of the subtree that contains v (if this subtree is in S).

Def 3. Quotient-cost (v, S), where S is a set of subtrees and v∈ V .Quotient-cost (v, S) is the cost of the merged tree merged-tree (v, S) divided by |S|.

The quotient-cost (v,S) represents the energy cost per subtree for merging the set of subtrees from v. Since our objective is to use less energy to merge more subtrees, quotient-cost (v, S) is a good cost metric for choosing node v (which will be the new root) to merge the set subtrees S.

Def 4. min-quotient-cost (*v*). Let *O* be a set of orphans and *S* ⊆ *O*, min-quotient-cost (*v*) = min {quotient- cost (*v*,*S*)}.

The min-quotient-cost (*v*) gives the best number of orphans that node *v* can merge. However, the complexity of computing the optimal value of min-quotient-cost (*v*) is too high (we need to try all the combinations of the orphans). We take an approximation method to compute min-quotient-cost (*v*) as the following:

1) Compute the shortest path (in terms of energy cost) from v to each of the orphans;
2) Sort these paths in ascending order of their cost;
3) Choose the first *i* shortest paths, $i \geq 1$ if *v* is in the subtree whose root is *s*, or $i \geq 2$ otherwise.

Step 3 makes min-quotient-cost (*v*) the smallest, because the new subtree is the union of the first *i* paths. Notice that if *v* is in the subtree rooted from *s*, quotient-cost(*v*) is the cost of the tree linking |*S*| orphans divided by (|*S*| + 1), because one orphan is eliminated when |*S*| is 1.

The detailed Tree-Join-Tree greedy algorithm is as the following:

Input *G*= (*V*, *A*) and (*s*, D)
Output *T*: a multicast tree rooted from *s*.
  O = {{*i*}| *i* ∈ *D*}; //*O*: the set of orphans.
  While (|*O*| > 0) **do**
    Choose *v* ∈ *V* with the smallest min-quotient-cost(*v*);
    Link *v* to the roots of orphans by shortest path tree;
    If *v* = *s* or *v* is in a subtree rooted from *s* **t**hen Remove the new tree from *O*; Output the multicast tree rooted from s.

3.2.2 Given a request (s, D) in G(V, A), the tree-join- tree greedy algorithm can output a multicast tree in time O(kn^3).

Proof. It is easy to see that the greedy algorithm can output a multicast tree. The while-loop has at most k iterations (k = |D|). For each loop, computing the quotient cost for all nodes takes O (nn2). Thus, the while-loop takes time O (kn3). Therefore, the whole algorithm ends in time O (kn3).

4. SIMULATIONS

In the simulations, we compare our two greedy algorithms, Node-Join-Tree greedy (NJT-g) and Tree-Join-Tree Greedy (TJT-g) with a MIP-like method. The original MIP method in [5, 6] assumes each node can adjust its transmission power dynamically and tries to adjust the power of each node to a minimal level, which is different from our network model. Since the basic idea of the MIP method is to prune a broadcast tree into a multicast tree, we follow this idea to use our TJT-g algorithm to construct a broadcast tree and then prune it into the required multicast tree. We call this method BCT-p (Broadcast Tree and pruning). We believe the BCT-p method can fairly represent the family of multicast routing methods in [5, 12].

We study how the total energy cost is affected by varying three parameters: the total number of nodes in the network (N), the number of destination nodes in the multicast group (M), and the radius of power coverage (R).

The simulation is conducted in a 100×100 2-D free-space by randomly allocating 50 nodes. The unit of R is respect to the diagonal distance in the square region, i.e., when R = 1, a node's transmission range covers the whole region. The power model is $P=r^2$, where P is the transmission power and r the radius that the signal can reach. The radius of transmitter range for each node is generated from a normal distribution with both mean and variance equal to R.

We present averages of 100 separate runs for each result shown in the figures. In each run of the simulations, for given N, M, and R, we randomly place N nodes in the square, and randomly select a source node, M destination nodes, and the radius of each node. Any topology which is not connected is discarded. Then we run the three algorithms on this network.

In Fig. 2, we fix N and R while vary M. As we can see, the total energy cost of multicast trees increases as the growth of M. After M reaches 30, this increase becomes very slow. This is because when M reaches a certain number, the transmitting nodes in a multicast tree can almost cover the whole region anyway. The further inclusion of more nodes into the destination group will not add in extra energy cost. From Fig. 2, we can also find that the NJT-g algorithm performs the best. The energy cost of the BCT-p method increases quickly when M is small and it merges with the curve of TJT-g when M reaches 35 or above, because when M gets close to N, the multicast trees generated from the BCT-p and the TJT-g methods become almost the same except the leaf nodes. This result tells us that the multicast tree pruned from a broadcast tree is not efficient when the number of destinations is small.

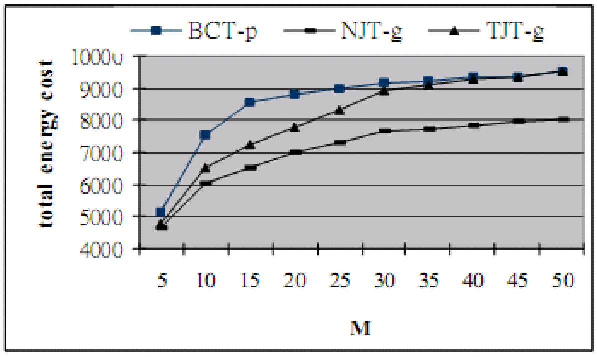

Fig. 2. ***N*** **= 50,** ***R*** **= 0.2**

Fig. 3 shows the result of fixing M and R, while varying N. The increase of N starts from 10, which is the number of multicast destinations (M). In the simulations, the first 10 nodes placed in the region become the multicast destinations. As N increases, more new nodes are added in while the positions of the existing nodes remain unchanged. From Fig. 3, we can see the cost for both NJT-g and TJT-g increases fast initially as the increase of N. But, when N reaches to some point (N = 30), the energy cost starts decreasing steadily as the further increase of N. This is because of the non-linear attenuation of transmission power. Initially, when N is small, the distance between nodes is large and it requires a lot more power for a node to reach its neighbors. As the increase of N, there are more nodes in the region, which gives much more choices to use smaller transmission power to relay message from node to node (i.e., there are more short distance links, which cost less energy, in the tree). As for BCT-p method, the cost does not drop as the increase of N. This is because in the BCT-p method, a multicast tree is trimmed from a broadcast tree. As the increase of N, there is more chance to include more unnecessary nodes in the multicast tree.

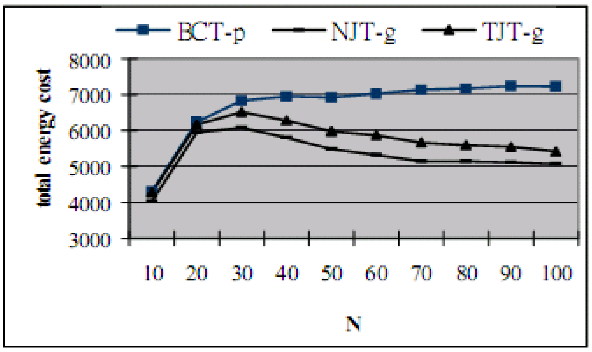

Fig. 3. *M* = 10, *R* = 0.2

In the simulation of Fig. 4, R is adjusted as N changes. We set R=$1/\sqrt{N}$ so that the nodes are just powerful enough to form a connected topology. In fact, it is observed that around half of the cases there the resulting network is connected for R=$1/\sqrt{N}$.

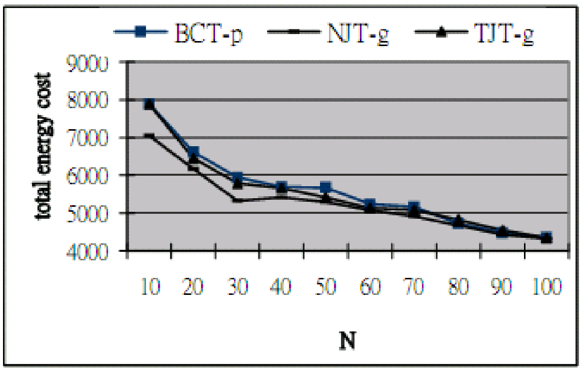

Fig. 4. $M = 10, R = \frac{1}{\sqrt{N}}$

From Fig. 4, we have two interesting findings. First, all the algorithms perform closely. By examining the multicast trees computed by the three algorithms carefully, we found they are very similar.

This is because the network topology is very sparse (with less arcs) with a small transmission range of each node and there is little routing choice in such a topology. Second, the cost of all algorithms decreases as the increase of N. This is again due to the non-linear power attenuation of radio signals. As the increase of N, nodes need less power to connect to their neighbors and this power-saving overpowers the extra cost brought in by the increase of N. This result tells us that with more nodes using small transmission power in a region, broadcast / multicast would cost less energy than the case of having fewer nodes with higher transmission power.

In Fig. 5, we fix *N* and *M* unchanged and see how the energy cost changes as varying *R*. There are two interesting observations in Fig. 5. First, the curves of the three methods merge together at the both ends when *R* is small and *R* is large. Their performances differ from each other only when *R is* in between 0.20 – 0.50. When R *is* too small, the network is barely connected; all three methods produce similar results due to the limited choice for routing. When *R* becomes large, all algorithms give nearly the same results again, because each node can cover a large region and only a few transmitting nodes in a multicast tree can cover all the destinations (notice that only the non-leaf nodes, i.e., transmitting nodes, in the multicast tree contribute to the cost). Second, the energy cost increases as the increase of *R*. This is consistent with the non-linear power attenuation law. With larger *R*, the energy cost increases much fast than the decrease of the transmitting nodes in a multicast tree. In fact, by the non-linear power attenuation law, two nodes communicating directly would cost more energy than relaying messages by a third node between them. This is also true in multicast.

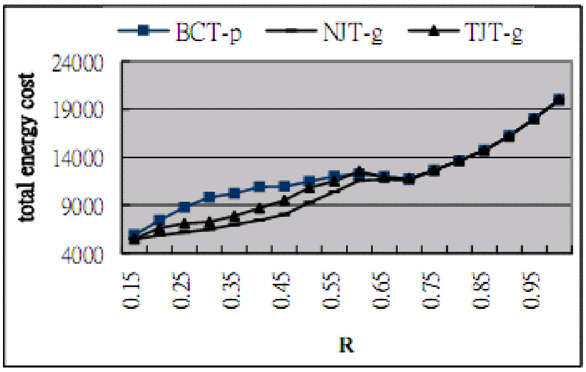

Fig.5. *N* = 50, *M* = 10

5. CONCLUSION

Three methods have been proposed, a Steiner tree based method, a node-join-tree greedy (NJT-g) method and a tree-join-tree greedy (TJT-g) method. In all three methods, we assume nodes are static and the proposed methods are centralized algorithms. However, our two greedy methods can be easily extended to distributed algorithms, especially for the NJT-g algorithm. The NJT-g algorithm basically uses only the neighbor information to construct a multicast tree from the source node and grow the tree step by step towards all destinations, which is in a distributed fashion.

6. REFERENCES

[1] Volkan Rodoplu and Teresa Meng, “Minimum Energy Mobile Wireless Networks”, IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, Aug 1999, pp.1333-1344.

[2] Ram Ramanathan and Regina Rosales-Hain, “Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment”, IEEE INFOCOM 2000, Tel-Aviv, Mar 2000, pp. 404-413.

[3] Roger Wattenhofer, Li Li, Paramvir Bahl and Yi-Min Wang, “Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks”, IEEE INFOCOM 2001, Alaska, USA, April, 2001.

[4] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “Algorithm for Energy-Efficient Multicasting in Static Ad Hoc Wireless Networks”, Mobile Networks and Applications 6, 2001, pp. 251-263.

[5] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks”, IEEE Infocom’2000.

[6] J. E. Wieselthier, G. D. Nguyen and A. Ephremides, “Energy-Efficient Broadcast and Multicast Trees in Wireless Networks,” Mobile Networks and Applications 7, 2002, pp. 481-492.

[7] P. J. Wan, G. Calinescu, X. Y. Li and O. Frieder, “ Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks”, IEEE Infocom’2001.

[8] Maggie Xiaoyan Cheng, Jianhua Sun, Manki Min, Ding-Zhu Du, "Energy Efficient Broadcast and Multicast Routing in Ad Hoc Wireless Networks", 22nd IEEE International Performance, Computing, and Communications Conference, Phoenix, Arizona, USA, 2003.

[9] Omer Egecioglu and Teofilo F. Gonzalez, “Minimum-Energy Broadcast in Simple Graphs with Limited Node Power”, In Proc. IASED International Conference on Parallel and Distributed Computing and Systems (PDCS 2001), Anaheim, CA, August 2001, pp. 334-338.

[10] M. Cagalj, JP Hubaux, C. Enz, “Minimum-Energy Broadcast in All- wireless Networks: NP-Completeness and Distribution Issues”, MOBICOM’02.

[11] W. Liang, “Constructing Minimum-Energy Broadcast Trees in Wireless Ad Hoc Networks”, MOBIHOC’02.

[12] Deying Li, Xiaohua Jia and Hai Liu, “Energy Efficient Broadcast Routing in Ad Hoc Wireless Networks”, IEEE Trans. On Mobile Computing, vol.3, no.1, 2004.

[13] D. S. Hochbaum, Approximation Algorithms for NP-hard Problems, PWS Publishing Company, 1997.

[14] L. Zosin, S. Khuller, “On Directed Steiner Trees”, Proceedings of the thirteenth annual ACM-SIAM Symposium on Discrete Algorithms, 2002, pp.59-63.