Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 30 (2007) 3746–3756
computer communications
www.elsevier.com/locate/comcom

Energy efficient multicast routing in ad hoc wireless networks

Deying Li [a,b], Qin Liu [c], Xiaodong Hu [d], Xiaohua Jia [b,*]

[a] School of Information, Remin University of China, Beijing, China
[b] Department of Computer Science, City University of Hong Kong, Hong Kong
[c] School of Computing, Wuhan University, Hubei Province, China
[d] Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China

Received 2 February 2007; received in revised form 29 July 2007; accepted 5 September 2007
Available online 14 September 2007

Abstract

In this paper, we discuss the energy efficient multicast problem in ad hoc wireless networks. Each node in the network is assumed to have a fixed level of transmission power. The problem of our concern is: given an ad hoc wireless network and a multicast request, how to find a multicast tree such that the total energy cost of the multicast tree is minimized. We first prove this problem is NP-hard and it is unlikely to have an approximation algorithm with a constant performance ratio of the number of nodes in the network. We then propose an algorithm based on the directed Steiner tree method that has a theoretically guaranteed approximation performance ratio. We also propose two efficient heuristics, node-join-tree (NJT) and tree-join-tree (TJT) algorithms. The NJT algorithm can be easily implemented in a distributed fashion. Extensive simulations have been conducted to compare with other methods and the results have shown significant improvement on energy efficiency of the proposed algorithms.

Keywords: Ad hoc wireless network; Energy efficiency; Multicast routing

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 communication from a single source node 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. Broadcast is a special case of multicast, in which all nodes in the network should receive the broadcast message. When omnidirectional 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

* Corresponding author. Tel.: +8522788670.
E-mail address: jia@cs.cityu.edu.hk (X. Jia).

0140-3664/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2007.09.003

the tree. We assume the reception of signals cost no extra energy.

There has been a lot of work on energy efficient broadcast/multicast routing in ad hoc networks [1]. However, most of the existing work assumes that each node can adjust its transmission power based on the distance to the receiving node and the background noise either continuously or in a discrete fashion. We assume that each node has a preconfigured transmission power and we aim at, for each multicast request, finding a multicast tree that has the minimum energy consumption. This is a more practical issue in real systems because each node would have a transmission power after the network is configured and this power level will not be changed for each multicast request. To the best of our knowledge, our work is the first to address the energy efficient multicast routing under fixed level of transmission power.

In this paper, we first prove this problem is NP-hard and unlikely has an approximation algorithm with a constant performance ratio of the number of nodes in the network. We then propose a Steiner tree based algorithm that is a centralized algorithm and has a theoretical guaranteed approximation performance ratio. We also propose two efficient heuristics, node-join-tree and tree-join-tree algorithms. The node-join-tree algorithm can be easily implemented in a distributed fashion. Simulation results have shown significant improvement on energy efficiency of our proposed algorithms.

2. Related work and our contributions

Most of the previous studies on energy efficient multicast/broadcast have focused on configuring energy power of each node. That is, given the geometric positions of a set of nodes in a plane, to find the transmitting power of each node, such that the energy cost of the multicast/broadcast tree is minimized [2–16]. As for broadcast, finding such a broadcast tree is similar to the problem of finding a connected network topology (usually a tree) by using minimal power consumption [18–21]. But, a broadcast tree is directed (from the root leading to all leaf-nodes) while a topology tree is undirected.

Some energy-efficient broadcast algorithms were proposed in [2,3], namely BIP (Broadcast Incremental Power), MST (Minimum Spanning Tree), and SPT (Shortest Path Tree). The proposed algorithms were evaluated through simulations, but little is known about their analytical performances in terms of the approximation ratios. The authors in [4] gave the quantitative characterization of performances of these three greedy heuristics. In [5], the problem of broadcasting in large ad hoc wireless networks was discussed and a method MLE (Minimum Longest Edge) based on MST was proposed. This algorithm provided a scheme to balance the energy consumption among all nodes. In [6], the GPBE (Greedy Perimeter Broadcast Efficiency) algorithm was proposed. GPBE applies the same tree formation procedure as the BIP, but based on another greedy decision metric broadcast efficiency. Simulation results show that GPBE performs better than BIP when the source is located near the center of the deploy region.

The above mentioned work assumes that all nodes are able to adjust their transmission power continuously and use geometrical properties of Euclidean plane. Some other work [7–9,17] assumes that all nodes can adjust their transmission power in a discrete way and apply graph theories and techniques to the construction of broadcast trees. In [7], the minimum-energy broadcast problem was addressed and proved to be NP-hard in general, and an $O(n^{k+2})$ algorithm was proposed for the problem under the assumption that each node is able to reach all the other nodes in the network, where n is the number of nodes and k the number of transmitters. In [8], authors first gave a formal proof of the NP-hardness of the problem for both geometrical version and graph version. A heuristic based on MST algorithm was proposed, but no performance ratio was given. In [9], another heuristic algorithm based on directed Steiner tree method was proposed. The performance ratio of the proposed algorithm is n^{ε}, where ε is a constant between 0 and 1. For the special case of the problem where each node has the same level of transmission power, an algorithm with performance ratio $\log^3 n$ was proposed.

Different from broadcast, multicast is to send a message to a subset of nodes in the network. Some fundamental issues associated with energy-efficient multicast were discussed in [10], and several multicast schemes were proposed and evaluated. Another typical work on multicast routing was MIP (Multicast Incremental Power) method, which was developed in [2] as an extension of the BIP method. An MIP multicast tree is obtained from the BIP broadcast tree by pruning the branches that do not contain the multicast destinations. The similar methods were used in [3,5] for constructing a multicast tree from a broadcast tree. In [11], the authors first proved the approximation ratios of the pruned based multicast tree algorithms p-SPT, p-MST and p-BIP, and all three heuristics have $\Omega(n)$ lower bounds. Then two constant approximation ratio algorithms SPF (Shortest Path First) and MIPF (Minimum Increment Path First) were proposed. In [12], an approximation algorithm for multicast in symmetric wireless ad hoc networks was proposed, and the solution delivered by the proposed algorithm is within $4\ln K$ times of the optimum if the transmission power at each node is finitely adjustable; otherwise, the solution is within either $8\ln K$ or $4\ln K$ times of the optimum, depending on whether or not the amount of power at nodes is incorporated into the running time, where K is the number of destination nodes in a multicast request.

Some research uses the local search technology to refine a multicast tree iteratively. The S-REMiT (Refining Energy-Efficient Source-based Multicast Tree) algorithm tries to minimize total tree power of the initial multicast tree using local search technology [13]. The DMEM (Distributed Minimum Energy Multicast) algorithm was proposed in [14] to reduce as much as possible the total transmission power required by the multicast communication in mobile ad hoc networks. Several localized operations are presented for DMEM algorithm, in which each node requires only the knowledge of and distances to all neighboring tree nodes. Other interesting approaches includes the localized algorithms [15,16] using geographic multicast when the possible information is available.

Besides the above mentioned work, recent research also extends the minimum energy multicast/broadcast with omnidirectional antennas to the use of directional antennas for further energy saving in wireless ad hoc networks, for example [22,23].

The problem that we address in this paper is similar to work in [12], where each node can adjust transmission power in discrete fashion. Fixed power level can be regarded as a special case that each node just has one power level. However, our work differs from [12] in that we consider the asymmetric wireless ad hoc networks, meaning that the sending and the receiving nodes in communication are not necessary to be in the transmission range of each other, while symmetric communication is required in [12]. The main contributions of this paper include: (1) We prove that the graphical version of energy efficient multicast routing problem in ad hoc wireless networks is unlikely to have an approximation algorithm with performance ratio of $\ln(n)$. This implies that the graphical version of energy efficient multicast routing problem is much harder than the geometrical version since the latter has approximation algorithms with constant performance ratios [4]. (2) We propose two routing techniques, node-join-tree and tree-join-tree methods. Among them, the node-join-tree method can be easily implemented in a distributed way. Our simulation study shows that they are more energy efficient than those methods based on the pruning-tree technique such as in [2,3,5].

3. Network model and problem specification

The network is modeled by a directed graph $G = (V,A)$, where V represents the set of nodes and A the set of arcs in the network. Each node, $v \in V$, is associated with a transmission power $p(v)$. For any two nodes v_1 and v_2, if v_2 is in the transmission power range of v_1 (i.e., $d^{\alpha}(v_1,v_2) \leqslant p(v_1)$), α is a constant value between $2 \sim 4$), then there is an arc $(v_1,v_2) \in A$ (i.e., a directed link from v_1 to v_2).

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

$$C(T) = \sum_{v \in NL(T)} p(v). \quad (1)$$

Our problem is how to, given a multicast request (s,D) and $p(v)$ for each node v, find a multicast tree rooted at s and spanning all nodes in D such that total energy cost defined in (1) is minimized. We call it *Minimum Energy Multicast* (MEM) 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.

4. Complexity analysis

In this section we will first prove that the MEM problem is NP-hard, and then we will show that it is even difficult to find a solution whose cost is close to the cost of optimal solution of the MEM problem.

Theorem 1. *The MEM problem is NP-hard.*

Proof. To prove the theorem, it suffices to show that the set cover problem is polynomial time reducible to the MEM problem since the former is NP-hard [24]. The decision version of set cover problem is defined as follows: Given a set I of n elements, $C = \{C_1,C_2, \ldots,C_m\}$, $C_j \subseteq I, j \in \{1,2,\ldots,m\}$, and a positive integer k. Does C contains a *set cover* for I of size k or less, i.e., a subset $J \subseteq \{1,2,\ldots,m\}$ such that $\cup_{j \in J} C_j = I$ and $|J| \leqslant k$?

We now construct a directed graph $G = (V,A)$, where $V = \{s\} \cup C \cup I$. For each $C_j \in C$, there is an arc (s,C_j). For any $i \in I, C_j \in C$, there is an arc from C_j to i if C_j covers i, i.e., $i \in C_j$. See Fig. 1(a), where $m = 5$ and $n = 7$. We also assume that $p(v) = 1$ for every node $v \in V$.

Suppose the multicast request is (s,I), where s is the source and I the destination set. In the following, we will prove that there is a set cover of I with size k if and only if there is a multicast tree for (s,I) with cost $1 + k$.

Let $\{C_{j1},C_{j2},\ldots,C_{jk}\}$ be a set cover with size k. We then construct a multicast tree T in which s transmits the multicast message to $C_{j1},C_{j2},\ldots,C_{jk}$ and $C_{j1},C_{j2},\ldots,C_{jk}$ relay the message to I. See Fig. 1(b), where $k = 3$ and T consists of solid arcs (one of arcs incident to node 5 could be removed). It is obviously that T is a multicast tree for (s,I). The energy cost $C(T)$ is equal to $1 + k$ (for Fig. 1(b), $1 + k = 4 = p(s) + p(2) + p(4) + p(5)$).

Suppose that T is a multicast tree for (s,I) with cost $1 + k$. Since T spans $I, \{C_j | C_j \in T\}$ covers all nodes in I. Therefore, $\{C_j | C_j \in T\}$ is a set cover of I. Furthermore, $C(T) = 1 + |\{C_j | C_j \in T\}|$, thus, $\{C_j | C_j \in T\}$ is a set cover with size k. The proof is then finished. □

The above NP-hardness proof is simpler than the proved given in [17]. Moreover, by using a similar polynomial reduction in the proof of Theorem 1, we can prove another negative result as follows.

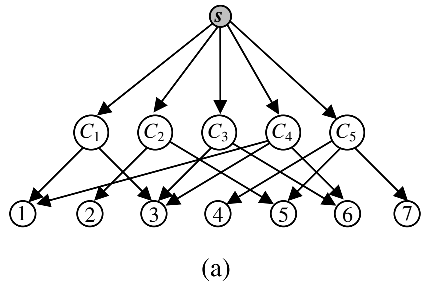


Fig. 1. Reducing the set cover problem to the MEM problem.

Theorem 2. *There is no approximation algorithm with performance ratio $\rho ln(n)$ for the MEM problem for any $\rho < 1$ unless $NP \subset D_{TIME}(n^{poly\ log\ n})$.*

Proof. Suppose, by contradiction argument, that there exists an algorithm $A_{\rho'}$ with approximation performance ratio $\rho'\ln(n)$ for the MEM problem for some $\rho' < 1$. That is, for any instance I' of the MEM problem, algorithm $A_{\rho'}$ returns a solution T' with

$$C(T') \leqslant \rho' ln(n) C_{opt}(I'), \tag{2}$$

where $C_{opt}(I')$ is the cost of the optimal solutions of I'. We now design an algorithm A_ρ for the set cover problem using algorithm $A_{\rho'}$ as a subroutine.

Given an instance I of the set cover problem, set $\Delta = \max\{\lceil \rho'/(1-\rho')\rceil, m\}$. Algorithm A_ρ first makes an instance I' of the MEM problem such that I has a set cover of size k if and only if I' has a multicast tree of cost $1 + k\Delta$. (This can be realized by just assigning $p(C_j) = \Delta$ for each C_j in the reduction of Theorem 1.) Algorithm A_ρ then finds a multicast tree T' by applying algorithm $A_{\rho'}$ to I'. In the end Algorithm A_ρ produces a set cover C of I associated with T'. Now let $C_{opt}(I)$ be the costs of the optimal solution of I. Then by the assumptions we have

$$\begin{aligned} C(T') = 1 + \Delta|C| &\leqslant \rho' ln(n) C_{opt}(I') \\ &\leqslant \rho' ln(n)(1 + \Delta C_{opt}(I)) \quad \text{(by inequality (2))} \\ &= \rho'\Delta ln(n) C_{opt}(I) + \rho' ln(n) \end{aligned}$$

from which we deduce

$$\begin{aligned} |C| &< \rho' ln(n) C_{opt}(I) + (\rho'/\Delta) ln(n) \\ &\leqslant \rho'(1 + 1/\Delta) ln(n) C_{opt}(I). \end{aligned}$$

Since $\rho'(1 + 1/\Delta) < 1$, the above inequality contradicts the result in [25] that claims that the set cover problem has no approximation algorithm with performance ratio $\rho\ln(n)$ for any $\rho < 1$ unless $NP \subset D_{TIME}(n^{\text{poly log } n})$. Hence the proof is complete. □

Since it is widely believed that NP is not a subset of $D_{TIME}(n^{\text{poly log } n})$, the above theorem implies that the MEM problem is unlikely to have an approximation algorithm with a logarithmic performance ratio.

5. Algorithms

In this section we will first propose a Steiner tree based approximation algorithm with a guaranteed performance ratio, and then two efficient heuristics using some greedy strategies.

5.1. Steiner tree based algorithm

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 transform the network graph G to a new graph G' that has weight on arcs.

For each node $v \in V$, we split v to two nodes v' and v'' and connect them with a new arc from v' to v''. For each arc (v',v''), its weight is assigned to $p(v)$, the weight of v in G. For each arc $(v_1,v_2) \in V$, let (v_1'', v_2') be a new arc in G' and give it a weight 0. We get a new directed graph G', where $G' = (V' \cup V'', A'), V' = \{v'|v \in V\}, V'' = \{v''|v \in V\}$, and $A\prime = \{(v\prime, v\prime\prime)|v \in V\} \cup \{(v_1'', v_2')|(v_1, v_2) \in A\}$. Fig. 2 is an example of transforming a network graph G to G'. The number associated with each node in G is the weight of the node, while the numbers on the arcs in G' are the weights of the arcs. For a multicast request (s,D), we denote $D' = \{v''|v \in D\}$. The MEM problem in G can be transformed to the following problem in G': find a directed Steiner tree T rooted from s' and includes all nodes of D' in G' such that the sum of weights of arcs in T is minimized.

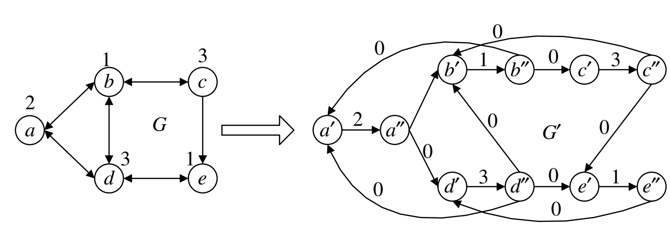


Fig. 2. Transforming the given graph G to the auxiliary graph G'.

This is known as *Directed Steiner Tree* (DST) problem, which has been well-studied. Any algorithm for the DST problem (e.g., [26,27]) can be used to find a solution to the MEM problem. In particular, when applying the approximation algorithm proposed in [26], we can easily prove the following theorem.

Theorem 3. *The DST-based algorithm has an approximation ratio of* $i(i-1)|D|^{1/i}$ *and time complexity of* $O((2|V|)^i|D|^{2i})$ *for any fixed* $i > 1$.

In the above theorem, when setting $i = \log|D|$, we obtain a DST-based algorithm that has an approximation ratio of $O(\log^2|D|)$, which is bounded by $O(\log^2|V|)$, but non-polynomial time of $O(|V|^{3\log|D|})$; When setting $i = 2$, we obtain a DST-based algorithm that has polynomial time of $O(|V|^2|D|^4)$, but a bigger approximation ratio $2|D|^{1/2}$. Here we have a tradeoff between solution's quality and running time.

5.2. Node-join-tree (NJT) algorithm

We first define three different sets before getting into details of the algorithm. The first one is *cover-set* C that consists of non-leaf nodes in the multicast tree. It is initialized to contain only the multicast source node s. The nodes in C will transmit multicast messages during multicasting and each of them can *cover* its neighbors in the sense that its transmissions can be received by its neighbors. In the algorithm, we aim at finding the "energy efficient" set C that covers all multicast destinations. The second set is *candidate-set* N, which is the union of the neighbors of all nodes in C. Each time in the algorithm a node in N will be selected to be included in C. By expanding C in this way, it maintains the nodes in C a tree structure (i.e., the multicast tree). The third set is *uncovered-set* U that contains the nodes not covered so far. A node v is *not covered* by C if v is neither in C nor a neighbor of any node in C.

This heuristic grows the multicast tree from s. Initially, C contains only s and U is assigned to D. All neighbors of s are removed from U and added to N, which means those nodes are now covered by s and are the candidates for the next-hop relay nodes for multicasting. Then, a node in N is selected to be included in C (the selection criteria is given below), and its neighbors are removed from U and added into N. This operation is repeated until U becomes empty, which means all destinations in D are now covered by set C and the nodes in $C$are the non-leaf nodes of the multicast tree. The multicast tree is thus obtained.

Let V_i denote the set of neighbors of v_i, i.e., $V_i = \{v_j|(v_i,v_j) \in A\}$ and $v_i \notin V_i$. In order to choose the nodes into cover-set such that the total energy cost defined in (1) is minimized, we use the following function to evaluate every candidate node $v_i \in N$:

$$f(v_i) = \frac{V_i \cap U|}{p(v_i)}. \tag{3}$$

This function represents the number of uncovered nodes a candidate can cover per energy unit. The larger value this function is, the more energy efficiently a candidate covers the destination nodes. Each time, a candidate node with the largest value among the nodes in N will be selected and put into the cover-set. As a result, 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 N that covers any node in U (i.e., there is no uncovered destination in the set of neighbors of any node in N), we select a node that is in the shortest path from s to some nodes in U. The details of the NJT 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: cover-set
  $U = D \backslash V_s$; //U: uncovered-set
  $N = V_s$; //S: candidate set
  While $(U \neq \emptyset)$ **do**
    Choose $v_i \in N$ with the largest $f(v_i)$ defined in (3).
    $C = C \cup \{v_i\}$;
    $U = U \backslash V_i$;
    $N = N \cup V_i$;
  End-while
  Construct the multicast tree T from C.

Theorem 4. *Given a request* (s,D) *in* $G = (V,A)$, *the NJT algorithm can output a multicast tree in time* $O(|V|^2)$.

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

This heuristic algorithm can be easily implemented in a distributed fashion, where each node is supposed to have only the information about its neighbors (1-hop neighbors) and the execution of the algorithm is based on the locally available information. The distributed version of the algorithm can be implemented as below. The multicast tree initially contains only s. Node s evaluates the f value of all its neighbors and selects a neighbor as the next tree-node (tree-nodes are the nodes in set C). A connection path is setup from itself to this neighbor. Every tree-node remembers the f value of its neighbors. The candidate set N consists of the neighbors of the tree-nodes and is maintained by the tree-nodes in a distributed way. Each time when a node is included into the tree, it evaluates the f value of its neighbors, and passes the largest f value back to s along the tree links. When s receives this f value, s can decide which candidate in N should be selected to be included into the tree, because all the tree-nodes have sent the f values of their neighbors to s at the time when they are included in the tree. Then, s sends a tree-growth command to the tree-node of the selected candidate along the tree path. The tree grows link by link in a distributed way until all the destination nodes are covered by the tree-nodes.

5.3. Tree-join-tree (TJT) algorithm

In the above greedy heuristic, the algorithm constructs a multicast tree in a top-down fashion, starting from the source node s. This TJT heuristic takes a global approach, starting from the destination nodes, to construct a multicast tree that has efficient energy cost.

Given a multicast request (s,D), the basic idea of this algorithm is as follows. Initially, each node in D is a subtree. Each time, a node $v \in V$ that uses the least energy to link the roots of two or more subtrees is selected to merge the subtrees into a bigger one, and v becomes the root of the newly merged subtree. This merging operation is repeated until all subtrees are merged into a single tree where s is the root. This final tree is the multicast tree.

In the above algorithm, a subtree is a directed tree and all its leaf-nodes are the nodes in D. A subtree whose root is not s is called an orphan subtree (*orphan* for short). In the initial step of the algorithm, every node in D is an orphan and s is the only subtree that is not an orphan. At the end of the algorithm, all orphans are merged into the subtree whose root is s. Let O denote the set of orphans. To evaluate the energy efficiency of using node v to merge a subset of orphans $O' \subseteq O$, we define a quotient function as:

$$quotient(v, O') = \frac{\text{EnergyCost}(SPT(v, O'))}{|O'|}, \tag{4}$$

where $SPT(v,O')$ is the shortest path tree rooted from v and spanning to the roots of all subtrees in O'. This function evaluates the energy efficiency of v for removing per orphan for the given subset of orphans O'. To see the best energy efficiency that node v can do in removing orphans, we need to find the minimal value of the quotient function for v to merge any arbitrary number of orphans. Therefore, we define the following q function for v as:

$$q(v) = \text{Min}\{\text{quotient}(v, O') | O' \subseteq O\}. \tag{5}$$

This function $q(v)$ represents the energy cost for removing per orphan. The smaller this value is, the less energy it costs for v to remove an orphan. Each time, by choosing node v with the smallest value of this function to merge the orphans, as the result, it takes the least energy to remove all orphans. That is, the final multicast tree would cost least amount of energy. However, the complexity of computing function $q(v)$ is too high. It has to try all the combinations of the orphans that v can merge. To reduce this computing complexity, we take the following approximate method to compute $q(v)$:

Step 1. Compute the shortest path (in terms of energy cost) from v to each of the orphans;
Step 2. Sort these paths in ascending order of their costs;
Step 3. Choose the first i shortest paths that makes $q(v)$ the smallest.

The new subtree is the union of the first i paths (i.e., a SPT). The detailed TJT algorithm is as the following:

Input $G = (V,A)$ and (s,D)
Output$T$: a multicast tree rooted from s.
  $O = \{\{d\}|d \in D\}$; //O: set of orphans.
  While $(O \neq \emptyset)$ **do**
    Choose $v \in V$ with the smallest $q(v)$ defined in (5);
    Link v to the roots of the orphans by the SPT;
    If $(v = s)$ or v is in a subtree rooted from s **then**
      Remove the new SPT from O; // the SPT is not an orphan
  End-while
  Output the multicast tree rooted from s.

Theorem 5. *Given a request* (s,D) *in* $G(V,A)$, *the TJT algorithm can output a multicast tree in time* $O(|D||V|^3)$.

Proof. It is easy to see that the greedy algorithm can output a multicast tree. The while-loop has at most $|D|$ iterations. For each loop, computing the quotient cost for all nodes takes time $O(|V||V|^2)$. Thus, the while-loop takes time $O(|D||V|^3)$. Therefore, the whole algorithm ends in time $O(|D||V|^3)$. □

6. Simulations

In the simulations, we compare our two greedy algorithms, NJT and TJT with a pruning-based method. To the best of our knowledge, there is no other work considers energy efficient multicast under the fixed transmission power assumption. The original MIP method in [2,3] 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 MIP method is to prune a broadcast tree into a multicast tree, we follow this idea to use our TJT 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 [2,3,5].

We study how the total energy cost is affected by varying three parameters over a wide range: 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). Since the average hop from the source to the multicast members is an important metric to reflect the average transmission delay, we also report the average hop for each parameter setting.

The simulation is conducted in a 100×100 2D 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 that is not connected is discarded. Then we run the three algorithms on this network.

In Fig. 3, we fix N and R while vary M. The two subfigures Fig. 3(a) $\sim$ (b) are obtained by setting R to 0.2 and 0.5 respectively. As we can see, the total energy cost of multicast tree increases as the growth of M in both subfigures. When R is 0.2, we can find this increase becomes slow after M reaches 20. 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 adds little extra energy cost. We can find NJT 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 when M reaches 30 or above, because when M gets close to N, the multicast trees generated from the BCT-p and the TJT 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, since it may contain many unnecessary intermediate nodes if the destinations are located close to the leaf nodes of the broadcast tree. When R is increased to 0.5 as shown in Fig. 3(b), the average transmission power of each node can cover a quarter of a square region. As we can see, the energy cost of multicast tree becomes saturated quickly even when M reaches about 15. This is because with a larger transmission range, only a few nodes are sufficient to cover the whole region. Therefore, any further increase of destinations will not add too many new transmitting nodes (i.e., non-leaf nodes) in the multicast tree.

Fig. 4 shows the average hop of each node when we fix N and R while vary M. As we can see, the average hop decreases with the increase of R, because when the average transmission range of nodes becomes larger, fewer hops are needed between the source and the multicast member. In both subfigures, NJT has the minimum average hop, while TJT and BCT-p perform closely, although the three curves almost overlap when $R = 0.5$. The reason is that the TJT searches for a low cost node at each step, but has no direction to the source. This makes the multicast tree have more layers than the tree generated by NJT. However, the difference among three algorithms is not large with respect to the average hop, especially when R is large. Compared with Fig. 3, we can find the total energy cost increases as the decrease of the average hop. This is due to the non-linear attenuation of transmission power. 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.

Fig. 5 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. 5, we can see for both NJT and TJT, the energy cost decreases steadily as the increase of N. The reason is that the addition of nodes 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 are more chances to

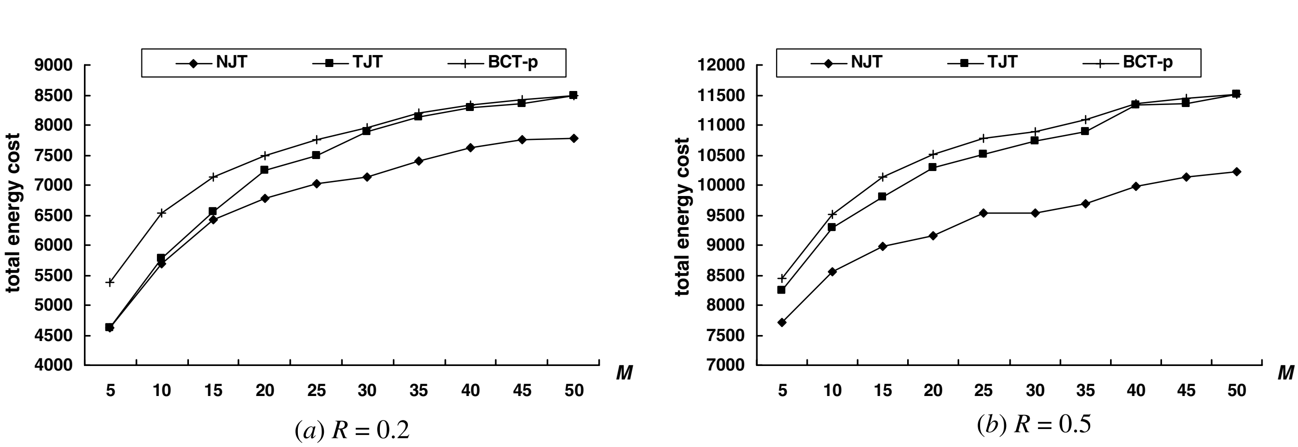
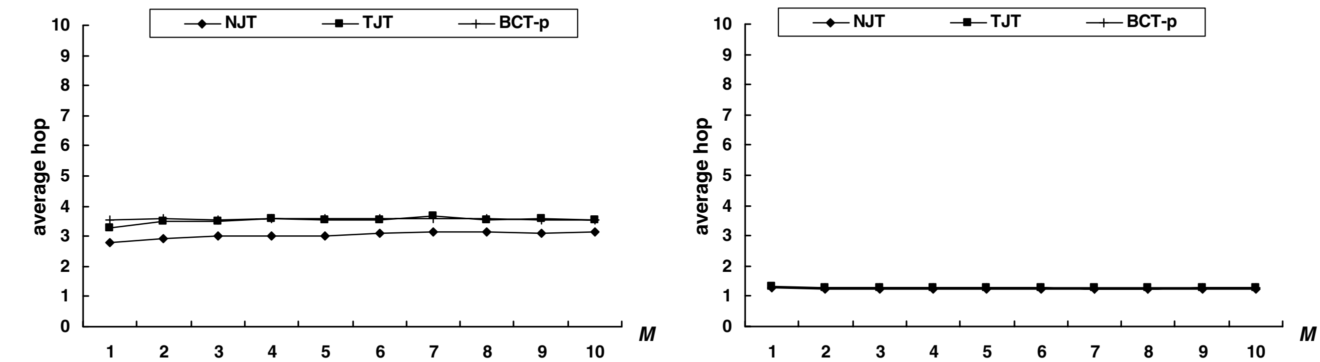


Fig. 3. The total energy cost for $N = 50$.



(*a*) $R = 0.2$ (*b*) $R = 0.5$

Fig. 4. The average hop for $N = 50$.

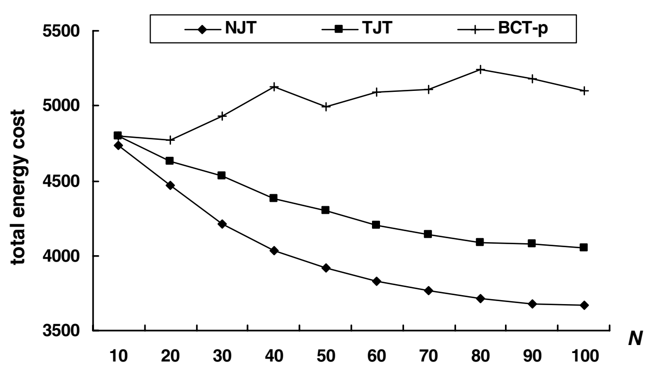


Fig. 5. The energy cost for $M = 10$, $R = 0.2$.

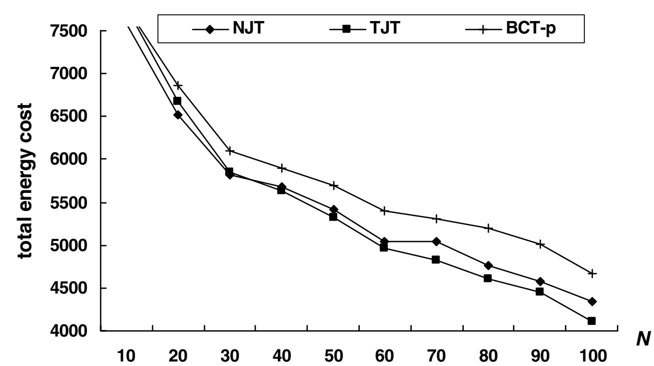


Fig. 7. The energy cost for $M = 10$, $R = 1/\sqrt{N}$.

include more unnecessary nodes in the multicast tree. Fig. 6 shows the average hop for $M = 10$, $R = 0.2$. The result is consistent with Fig. 4 that NJT has the lowest average hop and BCT-p gives the highest average hop, and their difference enlarges as more nodes are added.

In the simulation of Fig. 5 and 6, the average transmission range of nodes remains unchanged as the increase of N. In the network model where each node can adjust its transmission power, a node needs less power to make the network connected as the increase of N. In order to evaluate how the heuristics perform under tight energy budget, in the simulations of Fig. 7 and 8, R is adjusted as N changes. If there is only one single node, $R = 1$ is large

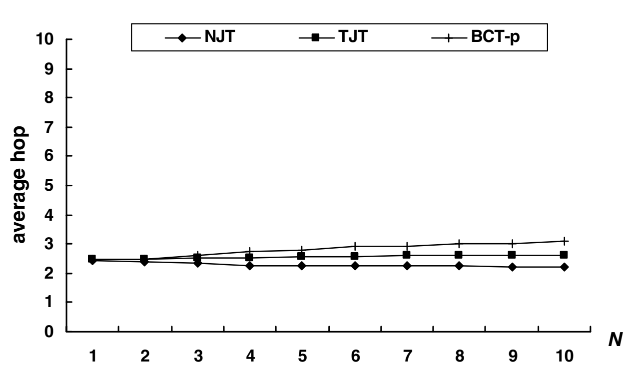


Fig. 6. The average hop for $M = 10$, $R = 0.2$.

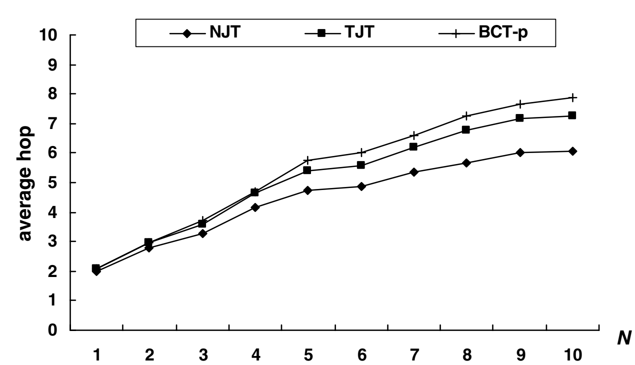


Fig. 8. The average hop for $M = 10$, $R = 1/\sqrt{N}$.

enough to cover the whole square region (notice that R is respect to the diagonal distance in the square region). Since the node positions are randomly generated, if there are 4 nodes equally spaced, $R = 1/2$ is enough to cover the whole region; and if there are 9 nodes, $R = 1/3$ is enough to cover the region. 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 $R = 1/\sqrt{N}$.

From Fig. 7, we have two interesting findings. First, TJT performs better than NJT when $N \geqslant 40$, which is different from the previous results. This is because the network topology is very sparse (with less arcs) with a small transmission range of each node, and for NJT there is very likely that the source can not find a node from the candidate set that can reach any uncovered node as the ratio N/M increases. Therefore, NJT selects a node in the shortest path from the source to a destination to grow the tree, and this energy efficient unicast route may not lead to an energy efficient multicast tree. However, the performance of NJT is still much better than BCT-p. 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. Fig. 8 shows the average hop for this parameter setting. We can find the average hop increases with the growth of N, and this leads to the decrease in the total energy cost. NJT still has the minimum average hop and its superiority to TJT and BCP-p become more obvious as the growth of N.

In Fig. 9 and 10, we fix N and M unchanged and see how the energy cost and the average hop changes as varying R. There are two interesting observations in Fig. 9. 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 $0.20 \sim 0.50$. 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. 10 consolidates our analysis again that the decrease of the average hop corresponds to the increase of the total energy cost.

To sum up the simulations, both NJT and TJT are better than BCT-p in almost all cases, and NJT generally performs best with the lowest energy cost and the smallest average hop. But when the network is quite sparse and the ratio N/M is high, TJT yields better results than NJT in terms of the energy cost. For a given algorithm, the increase of the average hop corresponds to the decrease of the energy cost, which verifies the non-linear power attenuation law that relaying message via short distance links consumes less energy than sending message over a long distance link directly.

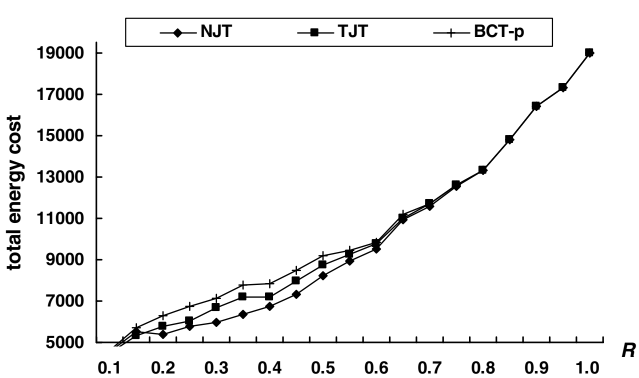


Fig. 9. The energy cost for $N = 50$, $M = 10$.

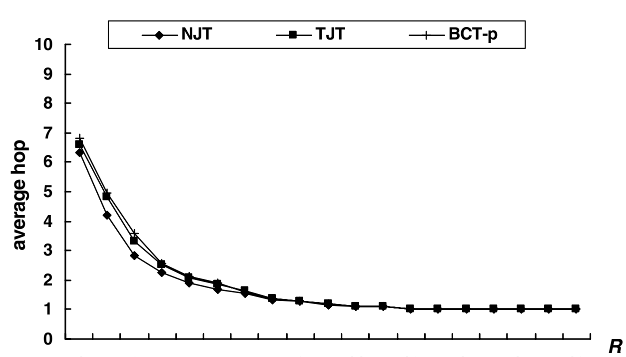


Fig. 10. The average hop for $N = 50$, $M = 10$.

7. Conclusion

We have studied the energy efficient multicast routing problem in ad hoc wireless networks. Three methods have been proposed, a Steiner tree based method, a node-join-tree greedy (NJT) method and a tree-join-tree greedy (TJT) method. Although the Steiner tree based method is a centralized method, which is helpful for theoretical analysis of multicast routing algorithms. It gives guaranteed performance ratio. The NJT algorithm can be implemented in a distributed fashion efficiently. It only requires each node to have the information about its direct neighbors. The TJT algorithm can also be implemented in a distributed way, but it will incur heavy communication cost, because the nodes need to elect a *best node* to merge orphans in each step of the algorithm. Besides of that, each node is also required to know the cost of the shortest paths to all other nodes (this information is need to compute $q(v)$ defined in (5)). This information can be obtained via some topology information exchange protocols used in ad hoc networks.

Extensive simulations have been conducted to compare our NJT and TJT methods with the typical MIP-like method. Simulation results have shown that our proposed methods outperform the MIP-like method.

Acknowledgement

This work was supported by a grant from Research Grants Council of Hong Kong [Project No. CityU

114006] and a grant from NSF China [Grant No. 60633020].

References

[1] S. Guo, O. Yang, Energy-Aware Multicasting in Wireless Ad Hoc Networks: A Survey and Discussion, Elsevier Computer Communications 30 (2007) 2129–2148.

[2] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, ‘‘On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks’’, Proceedings of the IEEE INFOCOM, New York, June 2002.

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

[4] P.J. Wan, G. Calinescu, X.Y. Li, and O. Frieder, Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks, Proceedings of the IEEE INFOCOM 2001 Conference, Anchorage, Alaska USA, April 2001.

[5] Maggie Xiaoyan Cheng, Jianhua Sun, Manki Min, and Ding-Zhu Du, ‘‘Energy Efficient Broadcast and Multicast Routing in Ad Hoc Wireless Networks’’, Proceedings of 22ndIEEE International Performance, Computing, and Communications Conference, Phoenix, Arizona, USA, 2003.

[6] I. Kang, and R. Poovendran, ‘‘A Novel Power-Efficient Broadcast Routing Algorithm Exploiting Broadcast Efficiency’’, Proceedings of IEEE Vehicular Technology Conference (VTC), Orlando, October, 2003, pp. 2926-2930.

[7] O. Egecioglu and T.F. Gonzalez, ‘‘Minimum-Energy Broadcast in Simple Graphs with Limited Node Power’’, Proceedings of IASED International Conference on Parallel and Distributed Computing and Systems, Anaheim, CA, Aug 2001, pp.334-338.

[8] M. Cagalj, J.P. Hubaux, and C. Enz, ‘‘Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues’’, Proceedings of 8th Annual International Conference on Mobile Computing and Networking, Atlanta, Georgia, Sept 2002.

[9] W. Liang, ‘‘Constructing Minimum-Energy Broadcast Trees in Wireless Ad Hoc Networks’’, Proceedings of 3th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, Switzerland, June 2002, pp. 112-122.

[10] J.E. Wieselthier, G.D. Nguyen, A. Ephremides, Algorithm for Energy-Efficient Multicasting in Static Ad Hoc Wireless Networks, Mobile Networks and Applications 6 (3) (2001) 251–263.

[11] P.J. Wan, G. Calinescu, C. Frieder, Minimum-Power Multicast Routing in Static Ad Hoc Wireless Networks, IEEE/ACM Transactions on Networking 12 (3) (2004) 507–514.

[12] W. Liang, Approximate Minimum-Energy Multicasting in Wireless Ad Hoc Networks, IEEE Transactions on Mobile Computing 5 (4) (2006) 377–387.

[13] B. Wang, S. Gupta, S-REMiT: a Distributed Algorithm for Source-Based Energy Efficient Multicasting in Wireless Ad Hoc Networks, Proceeding of IEEE GLOBECOM (2003) 3519–3524.

[14] S. Guo, O. Yang, Localized Operations for Distributed Minimum Energy Multicast Algorithm in Mobile Ad Hoc Networks, IEEE Transactions on Parallel Distributed. Systems 18 (2) (2007) 186–198.

[15] P.M. Ruiz, I. Stojmenovic, Cost-Efficient Multicast Routing in Ad Hoc and Sensor Networks’’ Handbook on Approximation Algorithms and Meta-Heuristics, Chapman & Hall/CRC, 2007.

[16] J.A. Sanchez, P.M. Ruiz, and I. Stojmenovic, Energy Efficient Geographic Multicast Routing for Sensor and Actuator Networks, Elsevier Computer Communications, in press.

[17] Deying Li, Xiaohua Jia, Hai Liu, Energy Efficient Broadcast Routing in Ad Hoc Wireless Networks, IEEE Transactions On Mobile Computing 3 (2) (2004) 144–151.

[18] V. Rodoplu, T. Meng, Minimum Energy Mobile Wireless Networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1333–1344.

[19] R. Ramanathan, R. Rosales-Hain, Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment, Proceedings of the IEEE INFOCOM 2000 Conference, Tel-Aviv (2000) 404–413.

[20] R. Wattenhofer, L. Li, P. Bahl, and Yi-Min Wang, ‘‘Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks’’, Proceedings of the IEEE INFOCOM 2001 Conference, Alaska, USA, April 2001.

[21] X. Li, W. Song, and W. Wang, ‘‘A Unified Energy Efficient Topology for Unicast and Broadcast’’, Proceedings of ACM MOBICOM, Cologne, Germany, August 28–September 02, 2005, pp. 1-15.

[22] F. Dai, Q. Dai, J. Wu, ‘‘Power Efficient Routing Trees for Ad Hoc Wireless Networks Using Directional Antenna’’, Elsevier Ad Hoc Networks 3 (5) (2005) 21–628.

[23] S. Guo, O. Yang, Minimum-Energy Multicast in Wireless Ad Hoc Networks with Adaptive Antennas: MILP formulations and Heuristic Algorithms, IEEE Transaction on Mobile Computing 5 (4) (2006) 333–346.

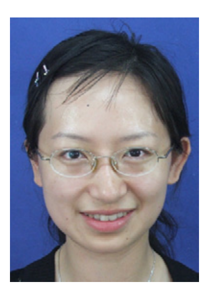
[24] D.S. Hochbaum, Approximation Algorithms for NP-hard Problems, PWS Publishing Company, 1997.

[25] U. Feige, ‘‘A Threshold of lnn for Approximating Set Cover’’, Proceedings of 28th ACM Symposium on Theory of Computing, Philadelphia, PA, May 1996, pp.314-318.

[26] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, ‘‘Approximation Algorithms for Directed Steiner Problems’’, Proceedings of 9th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, Jan 1998, pp.192-200.

[27] L. Zosin and S. Khuller, ‘‘On Directed Steiner Trees’’, Proceedings of 13th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, Jan 2002, pp.59-63.

Deying Li received the B.S. (1985) and M.S. (1988) degrees from the Central China Normal University, China, and received the Ph.D. (2004) degree from the Department of Computer Science, City University of Hong Kong. She is currently an Associate Professor in the Department of Computer Science at Renmin University of China. Her research interests include computer networks, ad hoc networks, sensor networks, and algorithm design and analysis.



Qin Liu received her B.Sc. and M.Sc. degrees in Computer Science from Central China Normal University, Wuhan, China, in 2001 and 2004, respectively. She obtained her Ph.D. degree in Computer Science from Wuhan University, China, in 2007. She is currently an assistant Professor in School of Computing at Wuhan University. Her research interests include distributed systems and computer networks.

Xiaodong Hu received his B.S. degree in applied mathematics from Tsinghua University, Beijing, China, in 1985, and his Ph.D degree in operations research and cybernetics from Institute of Applied Mathematics, Chinese Academy of Sciences, in 1989. Currently, he is a research professor at Institute of Applied Mathematics, Chinese Academy of Sciences. His research interests include combinatorial optimization and computer communication networks. He is the member of American Mathematical Society (since 1992) and member of IEEE Computer Society (since 1997).

Xiaohua Jia received his B.Sc. (1984) and M.Eng. (1987) from the University of Science and Technology of China, and D.Sc. (1991) in Information Science from University of Tokyo, Japan. He is currently a Chair Professor in Department of Computer Science at City University of Hong Kong and Cheung Kong Professor with the School of Computing, Wuhan University, China. Prof. Jia is an editor of IEEE Trans. on Parallel and Distributed Systems, Wireless Networks, Journal of World Wide Web, Journal of Combinatorial Optimization, etc. His research interests include distributed systems, computer networks, wireless sensor and mobile ad hoc networks.