Approximate Minimum-Energy Multicasting in Wireless Ad Hoc Networks

Weifa Liang, *Senior Member*, *IEEE*

Abstract—A wireless ad hoc network consists of mobile nodes that are equipped with energy-limited batteries. As mobile nodes are battery-operated, an important issue in such a network is to minimize the total power consumption for each operation. Multicast is one of fundamental operations in any modern telecommunication network including wireless ad hoc networks. Given a multicast request consisting of a source node and a set of destination nodes, the problem is to build a minimum-energy multicast tree for the request such that the total transmission power consumption in the tree is minimized. Since the problem in a symmetric wireless ad hoc network is NP-complete, we instead devise an approximation algorithm with provable approximation guarantee. The approximation of the solution delivered by the proposed algorithm is within a constant factor of the best-possible approximation achievable unless P = NP.

Index Terms—Wireless communication network, approximation algorithm, power awareness, ad hoc networks, energy consumption optimization, multicasting, broadcasting, minimum node-weighted Steiner tree problem.

1 Introduction

In recent years, multihop wireless ad hoc networks have been receiving significant attention due to their potential applications in civil and military domains. Some of the applications include mobile computing in areas where other infrastructure is unavailable, law enforcement operations, disaster recovery situations, large sporting events or congresses when it is not economical to build a fixed infrastructure for a short temporary usage, and tactical battlefield communications where the hostility of the environment prevents the application of a fixed backbone network.

A multihop wireless ad hoc network is dynamically formed by a collection of mobile nodes. Each of these mobile nodes is operated by a limited-energy battery and usually it is impossible to recharge or replace the batteries during a mission. The communication between two mobile nodes can be either in a single hop transmission in which case the two nodes are within the transmission ranges of each other, or in a multihop transmission where the message is relayed by intermediate mobile nodes. It is well known that wireless communications consume significant amounts of battery power [13]; therefore, the limited battery lifetime imposes a severe constraint on the network performance. Energy efficient operations are critical to enhance the network lifetime. Extensive studies on energy conservation in wireless ad hoc networks have been conducted. For example, energy efficient routing has been addressed in [2], [10], [22], [7], [24], [23], and maintaining network connectivity with the minimization of energy consumption has also been dealt with in [21], [26], [17], [24], [19], [15]. In particular, Ramanathan and Rosales-Hain [21] studied the assigning different transmission power to different nodes to meet a global topological property (e.g., k-connectivity) by proposing algorithms for maintaining the k-connectivity of a network ($k \geq 1$) with an objective to minimizing the total power consumption. Wattenhofer et al. [26] provided a distributed topology control algorithm for the maintenance of strong connectivity of a network. Li et al. [17] proposed an MST-based topology control algorithm that employs local information only (the information of neighboring nodes of a node). Singh et al. [24] provided several power-aware metrics and demonstrated how to use these metrics to find energy-efficient routes in wireless ad hoc networks.

Multicasting is a fundamental problem in any telecommunication network including wireless ad hoc networks. It is an efficient mechanism for one to many communication, and is typically implemented by creating a multicast tree. Due to severe battery power and transmission bandwidth limitations in wireless ad hoc networks, it is essential to develop efficient multicast protocols that are optimized for energy consumption, thereby significantly improving network performance.

1.1 Related Work

In recent years, there has been tremendous research interest on the design of energy-efficient broadcast/multicast routing protocols for wireless ad hoc networks. The minimum-energy broadcast tree problem is to find a broadcast tree rooted at the source node and spanning all the other nodes in a n-node network such that the sum of transmission power at relaying nodes is minimized. Several energy-aware algorithms for it have been proposed in the literature [27], [5], [18], [4], [6], [9], [25], [1]. For example, Wieselthier et al. [27] first considered this problem and proposed greedy heuristics, based on Prim's and Dijkstra's algorithms. Among the three heuristics that they suggested, the most efficient heuristic is called the Broadcast

- The author is with the Department of Computer Science, Australian National University, Canberra, ACT 0200, Australia. E-mail: wliang@cs.anu.edu.au.

Manuscript received 27 Oct. 2003; revised 26 Apr. 2004; accepted 8 Dec. 2004; published online 15 Feb. 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0177-1003.

Incremental Power (BIP). However, whether or not the problem is NP-hard was open until Liang [18] and Cagalj et al. [5] independently showed that it is NP-Complete. The result in [5] implies that there is unlikely to have a polynomial algorithm for the problem with an approximation ratio better than $\Omega(\log n)$ unless P = NP. Liang [18] considered the problem in both symmetric and asymmetric wireless ad hoc networks by proposing approximation algorithms with deterministic approximation ratios of $O(\log^3 n)$ and $O(n^\epsilon)$, respectively, where ϵ is any constant with $0 < \epsilon \leq 1$. Bian et al. [4] presented an improved approximation algorithm for the symmetric wireless ad hoc network with an approximation ratio of $2 \log n$ times of the optimum. Recently, Das et al. [9] provided exact solutions to the problem by modeling it into an integer programming (IP) problem. The applicability of this modeling, however, is very limited due to the fact that the approach based on the IP technique can only find an exact solution for the problem with small size. For a given instance of the problem, although it may take much less time to find an approximate solution with large problem size through relaxation, whether or not the approximate solution has a provable approximation guarantee is not clear. Cartigny et al. [6] devised a distributed algorithm for the problem, based only on the local information of participating nodes. Banerjee et al. [3] proposed schemes for constructing energy efficient broadcast and multicast trees for reliable wireless communication. Recently, Agarwal et al. [1] presented centralized and distributed heuristic algorithms for the problem using a concept called hitch-hiking, which takes advantage of the physical layer design that facilitates the combining of partial information to obtain complete information. For very special networks like Euclidean planar networks, Wan et al. [25] showed that the approximation ratio of minimum spanning tree is between 6 and 12 and that of BIP is between 13/3 and 12. However, Liang [18] showed that the approximation ratio of BIP in a general network is as bad as $\Omega(n)$. One closely related problem to the minimum-energy broadcast tree problem has also been addressed by Wieselthier et al. [28] and Kang and Poovendran [14], which aims to optimize two objectives simultaneously. One is to minimize the total transmission power consumption of the broadcast tree, and the other is to maximize the minimum residual battery energy at nodes in the tree. For this latter problem, they proposed heuristic algorithms as well.

The minimum-energy multicast tree problem is to find a tree in the network rooted at the source node and spanning the nodes in a destination set D. Clearly, the minimum-energy broadcast tree problem is a special case of this general setting. For the multicast problem, Wieselthier et al. [27] provided a heuristic called the Multicast Incremental Power (MIP), and Liang [18] gave an approximation algorithm for asymmetric wireless ad hoc networks with an approximation ratio of $O(|D|^\epsilon)$, where ϵ is a given constant with $0 < \epsilon \leq 1$.

1.2 Contributions

Given a multicast request and a symmetric wireless ad hoc network, in this paper, we devise an approximation algorithm for finding an approximate, minimum-energy multicast tree in the network for the request. 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 the multicast request. To the best of our knowledge, this is the first approximation algorithm for the problem in symmetric wireless ad hoc networks. The approximation of the solution is within a constant factor of the best-possible approximation achievable in polynomial time unless P = NP. The technique is to reduce the problem to a minimum node-weighted Steiner tree problem in a node-weighted auxiliary, undirected graph. The approximate, minimum node-weighted Steiner tree in the auxiliary graph is then transformed into a valid multicast tree in the original network through a series of transformations.

The rest of the paper is organized as follows: Section 2 introduces the wireless communication model and the problem definition. Section 3 proposes an approximation algorithm for the problem with an assumption that the transmission power at every node is finitely adjustable. Section 4 extends the approach in Section 3 to solve the problem by the removal of the assumption. The conclusion is given in Section 5.

2 Preliminaries

In this section, we first introduce the wireless communication model. We then define the problem precisely. We finally outline a known algorithm for finding an approximate, minimum node-weighted Steiner tree, which later will serve as a subroutine in our proposed algorithm.

2.1 Wireless Communication Model

We consider source-initiated, circuit-switched multicast requests (sessions). Each multicast request is a pair $(s; D)$, where s is the source node and D is the set of destination nodes. The wireless ad hoc network is modeled by an undirected graph $M = (N, A)$, where N is the set of nodes with $|N| = n$ and A is the set of edges. There is an edge $(u, v) \in A$ if nodes u and v are within the transmission ranges of each other. For any edge $(u, v) \in A$, its two endpoints u and v are called *neighboring nodes*. We assume that the network topology is stable during the processing period of a multicast request, where we say "processing a multicast request," means that the system either builds a multicast tree for and realizes the request using the built tree, or rejects the request if there are not enough network resources to accommodate the request. After it has finished processing the current multicast request and before its response to the next multicast request, the system allows the nodes in the network to move and a new network topology is then formed. We also assume that each node in the network is equipped with omnidirectional antenna and the transmission power at the node is finitely or infinitely adjustable. Each node can choose one of its power levels to transmit messages. In other words, we assume that there are l_i power levels at node $v_i \in N$. Let $w_{i,l}$ be the power of v_i at its power level l and $w_{i,j_1} \leq w_{i,j_2}$ if $j_1 < j_2$, $1 \leq j_1, j_2 \leq l_i$, and $1 \leq l \leq l_i$.

Among the l_i power levels, one is the minimum operational power level with power $p_{\min}(v_i)$ and another is the maximum operational power level with power $p_{\max}(v_i)$, $1 \leq i \leq n$. One such example is the Cisco Aironet card [11], which offers six power levels 1, 5, 20, 30, 50, and 100 *mW*. Furthermore, given two neighboring nodes u and v, we assume that there is always a corresponding power level between u and v with the same amount of power, which we refer to as *the power level symmetry of neighboring nodes*. Obviously, the amount of power to maintain the power level symmetry between u and v is the minimum power required to keep them within the transmission range of each other. For a transmission in the network from node u to node v, separated by a distance $d_{u,v}$, to guarantee that v is within the transmission range of u, the transmission power at u is modeled to be proportional to $d^{\alpha}_{u,v}$, assuming that the proportionality constant is 1 for notational simplicity, α is a parameter that typically takes a value between 2 and 4, depending on the characteristics of the communication medium. In other words, to make v within the transmission range of u, the power $w_{u,l}$ at u must meet $w_{u,l} \geq d^{\alpha}_{u,v}$.

The reachability of a node in wireless ad hoc networks is fully determined by the transmission power at the node. We have assumed that the power level of a transmission node can be chosen within a given range of values. Therefore, there is a trade-off between reaching more nodes in a single hop by using higher power and reaching fewer nodes in a single hop by using lower power. Note that nodes in any particular multicast tree do not necessarily have to use the same power level, and a node may use different power levels for various multicast trees in which it participates.

A wireless ad hoc network that meets the above requirements is called the *symmetric wireless ad hoc network*. A special case of the symmetric wireless ad hoc network is a network in which every mobile node is equipped with the same type of battery.

2.2 The Minimum-Energy Multicast Tree Problem

Given a wireless ad hoc network $M = (N, A)$ and a multicast request consisting of a source node s and a destination set D ($\subseteq N - \{s\}$), the *minimum-energy multicast tree problem* is to construct a multicast tree rooted at the source node and spanning the nodes in D such that the sum of transmission power at nonleaf nodes is minimized. The problem involves the choice of transmission nodes as well as the transmission power level at every chosen transmission node. Note that the leaf nodes do not contribute any transmission power consumption because they do not transmit any messages, where $K = |D|$. When $D = N - \{s\}$, the problem is referred to as the *minimum-energy broadcast tree problem*.

2.3 Approximation Algorithm for Finding Minimum Node-Weighted Steiner Trees

Given a node-weighted undirected graph $G(V, E)$ and a set D of destination nodes ($D \subseteq V$), the *minimum node-weighted Steiner tree problem* is to find a tree in G spanning the nodes in D such that the weighted sum of the nodes in the tree is minimized. Klein and Ravi [16] provided the first approximation algorithm for the problem, which is briefly described in the following.

The algorithm maintains a node-disjoint set of trees containing all the destination nodes. Initially, each destination node by itself is in a tree.

The algorithm uses a greedy strategy to iteratively merge the trees into larger trees until there is only one tree left. In each iteration, it selects a node and a subset of the current trees of size at least two so as to minimize the ratio

$$\frac{\textit{The weight of the node plus sum of the distances to the trees}}{\textit{number of trees}}. \tag{1}$$

Here, the distance along a path does not include the weights of the two endpoints of the path. Thus, the choice minimizes the average node-to-tree distance. The algorithm uses the shortest paths between the node and the selected trees to merge the trees into one.

It is easy to implement each iteration. For each node v, define the *quotient cost* of v to be the minimum value of (1), taken over all subsets of the current trees. To find the quotient cost of v, compute the distance d_i from v to each of the trees T_i, assuming without loss of the generality that the trees are numbered so that $d_1 \leq d_2 \leq \ldots \leq d_k$. In computing the quotient cost of v, it is sufficient to consider subsets of the form $\{T_1, T_2, \ldots, T_i\}$. Thus, the quotient cost for a given node can be computed in polynomial time by computing the quotient costs of all the nodes. The minimum quotient cost can then be determined. Thus, an iteration can be carried out within polynomial time. The solution delivered by the algorithm is $2 \ln K$ times of the optimum, where K is the number of the destination nodes. Note that the approximation of the solution is within a constant factor of the best possible approximation achievable in polynomial time unless $\tilde{P} \supseteq NP$ [20]. This result is stated in the following lemma.

Lemma 1 [16]. *Given a node-weighted undirected graph $G(V, E)$ and a destination set D with $|V| = n$, $|E| = m$, and $|D| = K$, there is an approximation algorithm for finding a minimum node-weighted Steiner tree in G including the nodes in D, which delivers a solution within $2 \ln K$ times of the optimum. The algorithm takes $O(K^2(m + n \log n))$ time.*

Proof. The approximation ratio of the Klein and Ravi algorithm has been shown in [16]. We here only analyze the time complexity of their algorithm.

It is obvious that the number of iterations is $K - 1$ at most. Within an iteration it takes $O(m + n \log n)$ time to compute single source shortest paths for each source node $v \in D$ using Dijkstra's algorithm, and there are K nodes in D. Therefore, the total running time of the computation within an iteration is $O(Km + Kn \log n)$. While finding a node $v \in D$ that has the minimum quotient cost takes $O(K)$ time, the algorithm takes $O(K^2 m + K^2 n \log n)$ time. □

Guha and Khuller [12] later provided an improved algorithm for the minimum node-weighted Steiner tree problem with a better approximation ratio at the expense of a longer running time. Their improved algorithm delivers a solution within $1.35 \ln K$ times of the optimum.

3 Approximation Algorithm with Finitely Adjustable Power

It is known that the minimum-energy multicast tree problem cannot be polynomially solvable with an approximation ratio better than $\Omega(\log K)$ unless P = NP by reducing the set cover problem to it. We therefore focus on devising an approximation algorithm for it. For the simplicity of discussion, in the following, we assume that the transmission power at each node is finitely adjustable. We then remove this assumption in Section 4.

3.1 An Overview of the Proposed Algorithm

We start with giving an overview of the proposed algorithm. The algorithm first constructs a node-weighted, auxiliary undirected graph $G(V, E, \omega)$ using the original network M, where $\omega : V \mapsto \mathbb{R}$. It then reduces the problem in the original network to a minimum node-weighted Steiner tree problem in G such that the weighted sum of the vertices in the Steiner tree is no greater than the minimum transmission power consumption for the multicast request. However, it is well known that finding such a Steiner tree is NP-hard. Instead, an approximate, minimum node-weighted Steiner tree is then found, which will be used as the base for constructing a valid multicast tree. It finally transforms the approximate, minimum node-weighted Steiner tree in the auxiliary graph into a valid multicast tree in the original network through a series of transformations.

The motivation for constructing such a minimum node-weighted Steiner tree is that the weighted sum of the vertices in the tree is a lower bound on the minimum transmission power consumption for the multicast request. The tree then is transformed into a valid multicast tree through a series of transformations. Within each transformation, the total power inflation from the previous tree to the resulting tree is guaranteed to be bounded. Thus, the transmission power consumption of the valid multicast tree is also bounded.

The algorithm is detailed in the following section.

3.2 Constructing an Auxiliary Graph

Given the wireless ad hoc network $M(N, A)$, assume that the power at each node is finitely adjustable. A node-weighted, auxiliary undirected graph $G = (V, E, \omega)$ is constructed as follows:

To distinguish the nodes in the original network M from those in the auxiliary graph G, the nodes in G are referred to *vertices*. Recall that for given a node $v_i \in N$, $w_{i,l}$ is the power of v_i at its power level l and $w_{i,j_1} < w_{i,j_2}$ if $j_1 < j_2$, $1 \leq l \leq l_i$, and $1 \leq j_1 < j_2 \leq l_i$. A widget $G_i = (V_i, E_i)$ for v_i is built and illustrated in Fig. 1.

$$V_i = \{s_i, u_{i,1}, u_{i,2}, \ldots, u_{i,l_i}\}, \text{ and}$$
$$E_i = \{(s_i, u_{i,j}) \mid 1 \leq j \leq l_i\},$$

where s_i represents node v_i, $u_{i,j}$ represents node v_i working at its transmission power level j, and the weight assigned to $u_{i,j}$ is the power $w_{i,j}$. An edge $(s_i, u_{i,j})$ between s_i and $u_{i,j}$ represents v_i working at its power level j, $1 \leq j \leq l_i$. For the sake of convenience, vertex s_i is called the *mobile vertex*, vertex $u_{i,j}$ is called the *power vertex*, vertex $u_{i,j}$ is a power

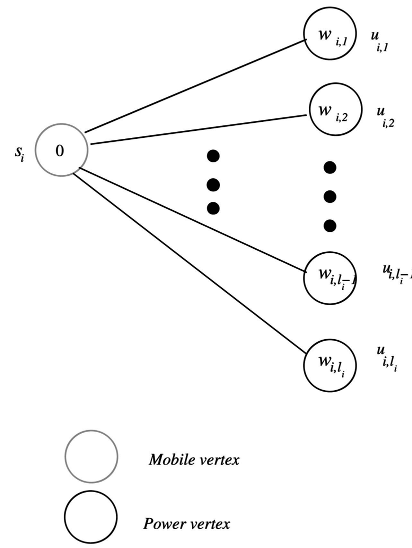


Fig. 1. The widget $G_i = (V_i, E_i)$ for node v_i.

vertex *derived from* s_i, and an edge $(s_i, u_{i,j})$ is a *derived edge* from s_i, $1 \leq j \leq l_i$ and $1 \leq i \leq n$.

Having a widget G_i for every node $v_i \in N$, it is ready to construct $G(V, E, \omega)$. $V = \cup_{i=1}^{n} V_i$ and $E = \cup_{i=1}^{n} E_i \cup E_{dist}$, where E_{dist} is a set of edges in which the two endpoints of an edge are in two different widgets, and E_{dist} is defined as follows: Given two nodes v_i and v_j in M with $i \neq j$, if v_i is working at its power level l and v_j is within the transmission range of v_i or *vice versa*, then there is an edge $(u_{i,l}, s_j)$ in E_{dist}, $1 \leq i, j \leq n$, and $1 \leq l \leq l_i$. In addition, the weight assignment to the nodes in G is as follows: $\omega(u_{i,l}) = w_{i,l}$ for each power vertex $u_{i,l} \in V_i \subset V$; and $\omega(s_i) = 0$ for each mobile vertex $s_i \in V$, $1 \leq l \leq l_i$, and $1 \leq i \leq n$. It can be seen that G has the following properties:

1. G contains

   $$|V| = \sum_{i=1}^{n} |V_i| = \sum_{i=1}^{n} (1 + l_i) = n + \sum_{i=1}^{n} l_i$$

   nodes due to that there are l_i power vertices and one mobile vertex in G for each node $v_i \in N$, $1 \leq i \leq n$. If v_j is within the transmission range of v_i when v_i works at its maximum power level, then there are l_i edges between the power vertices in G_i and s_j. In the worst case, there are $n - 1$ such s_js. Thus, the total number of edges in G incident to s_j is at most $k(n-1)$, $1 \leq j \leq n$, where $k = \max_{i=1}^{n}\{l_i\}$. So, $|E_{dist}| \leq k(n-1)n = kn^2 - kn$. Since $E_i \cap E_j = \emptyset$, $E_i \cap E_{dist} = \emptyset$, and $|E_i| = l_i$, G contains

   $$|E| = | \cup_{i=1}^{n} E_i \cup E_{dist}| =$$
   $$\sum_{i=1}^{n} |E_i| + |E_{dist}| \leq kn + kn^2 - kn = kn^2$$

   edges, $i \neq j$, $1 \leq i, j \leq n$.
2. Given two vertices in G, if both of them are either power vertices or mobile vertices, then there is no edge between them. In other words, for a given mobile vertex in G, only the power vertices are its

neighboring vertices and, for a given power vertex in G, only the mobile vertices are its neighboring vertices.

3. Let $N(u_{i,l})$ be the set of neighboring vertices of a power vertex $u_{i,l}$ in G. Then, $N(u_{i,j_1}) \subseteq N(u_{i,j_2})$ if $1 \le j_1 < j_2 \le l_i$.

3.3 An Approximation Algorithm

Having the auxiliary graph G, without loss of generality we assume that vertex s_1 is the vertex in G that corresponds to the source node in M, and vertices $s_2, s_3, \ldots, s_{K+1}$ are the destination vertices in G that correspond to the nodes in D, where $|D| = K$. The objective is to find a minimum node-weighted Steiner tree in G rooted at s_1 and spanning the vertices in $S = \{s_i \mid 2 \le i \le K+1\}$. We thus have the following lemma:

Lemma 2. *Given a multicast request $(s; D)$ in a wireless ad hoc network, the weighted sum of the vertices in a minimum node-weighted Steiner tree in $G(V, E, \omega)$ rooted at s_1 and spanning the vertices in $\{s_i \mid 2 \le i \le K+1\}$ is a lower bound on the exact solution of the minimum-energy multicast tree in M for the request.*

Proof. Assume that T_{opt} is a minimum-energy multicast tree in M rooted at the source node v_1 and spanning the nodes in D. Following the construction of G, there is a corresponding node-weighted Steiner tree T in G rooted at s_1 and spanning the vertices in S for T_{opt}, and the weighted sum of the vertices in T is equal to the sum of transmission power at the nodes in T_{opt}.

Let T_{st}^{opt} be a minimum node-weighted Steiner tree in G rooted at s_1 and spanning the vertices in S. Then, $W(T_{st}^{opt}) \le W(T)$ because T_{st}^{opt} is a minimum node-weighted Steiner tree in G, where $W(T')$ is the weighted sum of the vertices in a tree T'. The claim then follows.□

Lemma 3. *Let T be a node-weighted Steiner tree in G corresponding to a minimum-energy multicast tree T_{opt} in M and $parent(v)$ the parent of v in T. Then, 1) no more than one edge derived from a mobile vertex is included in T. In other words, for a given node $v_i \in N$, either $(s_i, u_{i,x})$ or $(s_i, u_{i,y})$ but not both of them is included in T, $1 \le x, y \le l_i$, $1 \le i \le n$, and $x \ne y$. 2) Let s_1 be the source vertex and s_j be a mobile vertex in T. Then, the number of edges in the unique path in T from s_1 to s_j is even and, for every other nonleaf mobile vertex $s_{j'}$, $j' \ne 1$ and $j' \ne j$ in the path, the degree of $s_{j'}$ in T is 2. In other words, assume that u_a and u_b are the two power vertices in T adjacent to $s_{j'}$ and $u_a = parent(s_{j'})$, then u_b must be the child power vertex of $s_{j'}$ derived from $s_{j'}$, $1 \le j' \le n$ and $j' \ne 1$.*

Proof.

1. Let s_i be a mobile vertex in T. If its both derived edges $(s_i, u_{i,x})$ and $(s_i, u_{i,y})$ are included in T with $x < y$, then two power levels at node v_i are needed for realizing this multicast request, following the construction of T. This contradicts the fact that there is only one power level at v_i chosen in T_{opt} for the realization of the multicast request.
2. Let P be the unique path in T from s_1 to s_j, consisting of the edges $e_1, e_2, \ldots, e_l$. Consider two adjacent edges e_{2i-1} and e_{2i} in P. The edge,
$$e_{2i-1} = (s_{j'}, u_{j',r(j')}) \in E_{j'} \subset E,$$
is an edge in the widget $G_{j'} = (V_{j'}, E_{j'})$ that is derived from the mobile vertex $s_{j'}$, where $u_{j',r(j')}$ is a power vertex corresponding to a power level $r(j')$ at node $v_{j'}$ in M, $1 \le r(j') \le l_{j'}$, $1 \le j' \le n$. The edge $e_{2i} = (u_{j',r(j')}, s_{j''}) \in E_{dist}$ is an edge between the power vertex $u_{j',r(j')}$ and the mobile vertex $s_{j''}$, $1 \le j'' \le n$, $j'' \ne j''$, and $1 \le i \le \lfloor l/2 \rfloor$. Since $e_1 \in E_{1'}$ and $e_l \in E_{dist}$, l must be even, i.e., the number of edges in P is even. The degree of $s_{j'}$ except the source vertex s_1 and leaf vertices is 2 because there are only two edges incident to it, one is from a power vertex (its parent in the tree) to it; and the other is from it to one of its power vertices (its child in the tree). □

Given a node-weighted Steiner tree T_1 in G rooted at s_1 and spanning the vertices in S, if it meets 1) and 2) in Lemma 3, then we say that the tree is a *valid multicast tree* in the wireless ad hoc network. Once T_1 is a valid multicast tree, the power level setting of every mobile node in the tree is straightforward, using the information provided by T_1. This can be done as follows: For a node v_i in M, if there is a corresponding edge $(s_i, u_{i,l})$ in T_1 in the path from s_1 to $u_{i,l}$, then the power level at v_i is adjusted to level l with power $w_{i,l} = \omega(u_{i,l})$, $1 \le l \le l_i$, and $1 \le i \le n$.

The rest is dedicated to finding a node-weighted Steiner tree T in G such that the weighted sum of the vertices in T is minimized. However, it is well known that there is unlikely to have a polynomial algorithm for finding such a Steiner tree in G unless $P = NP$. Instead, an approximate solution is expected. Let T_{app} be an approximate, minimum node-weighted Steiner tree obtained using an algorithm in [16]. Without loss of generality, we assume that all the leaf vertices in T_{app} are in S. Otherwise, those that are not in S can be pruned from the tree for any further consideration. Now, T_{app} may not be a valid multicast tree if it does not meet 1) and 2) of a valid multicast tree in Lemma 3. In other words, T_{app} may have the following violations:

1. For a given node $v_i \in N$, T_{app} contains up to l_i power vertices that are derived from a single mobile vertex s_i, while in a valid multicast tree only one of such power vertices derived from s_i is included.
2. T_{app} contains a power vertex that is not derived from any of its neighboring mobile vertices in the tree, while in a valid multicast tree each power vertex must be derived from one of its neighboring mobile vertices.
3. T_{app} may still not be a valid multicast tree, because, for a given mobile vertex s, it may contain such tree edges that each of them is from s to a power vertex u, and 1) either u is derived from s, but none of the mobile vertices derived from the other neighboring power vertices of s is within the transmission range of s when it works at the power level corresponding to u; 2) or none of the neighboring power vertices of

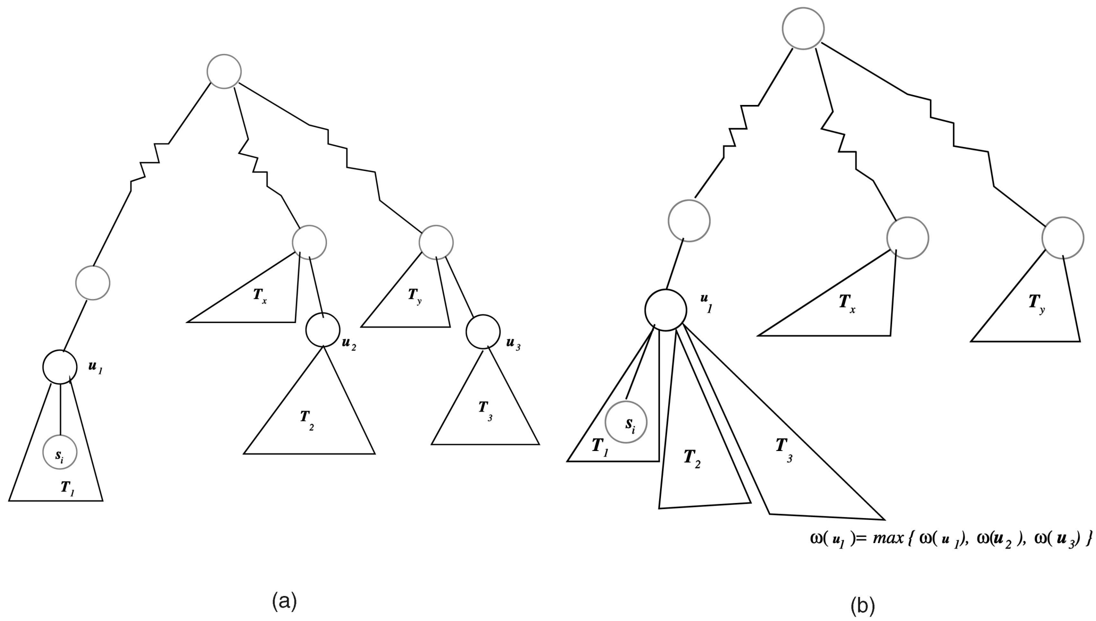


Fig. 2. (a) Power vertices u_1, u_2, and u_3 are derived from a mobile vertex s_i. (b) Merge the power vertices u_1, u_2, and u_3 derived from s_i into a single power vertex u_1 if $s_i \in N_T(u_1)$.

s (not including the parent of s) is derived from it, while in a valid multicast tree the child power vertex must be derived from the mobile vertex for a given edge from a mobile vertex to a power vertex.

In what follows, we show how to transform T_{app} into a valid multicast tree by correcting these three violations one by one, through a series of transformations.

3.3.1 Correction Violation 1

To correct violation 1, T_{app} is modified as follows: Let $u_1, u_2, \ldots, u_r$ be the power vertices in T_{app} derived from a mobile vertex s_i, $1 \le r \le l_i$. For the sake of convenience, $N_T(u)$ will be used to represent the set of neighboring vertices of vertex u in the current tree. Initially, the tree is T_{app} itself, and subject to be updated dynamically.

Given two power vertices u_i and u_j in the current tree with $i \neq j$, we first assume that there is not any ancestor-and-descendant relationship between them, we then remove this constraint later, $1 \le i, j \le r$. We proceed as follows:

Case 1. If there is a power vertex u_j such that $s_i \in N_T(u_j)$, then, for each $u_{j'}$ with $j \neq j'$, the edge $(s(j'), u_{j'})$ in the path in the current tree from the root to $u_{j'}$ is removed, and vertices $u_{j'}$ and u_j are merged into a single vertex, where $s(j')$ is a mobile vertex, $1 \le j$, $j' \le r \le l_i$, and $1 \le j \le r$. Now, the power vertex u_j is replaced by another power vertex derived from s_i at a power level of power $\max\{\omega(u_1), \omega(u_2), \ldots, \omega(u_r)\}$. Obviously, the resulting graph is still a tree and the weighted sum of the vertices in the tree is no greater than that of T_{app}.

Fig. 2a illustrates this merging transformation, where u_1, u_2 and u_3 are the power vertices derived from a mobile vertex s_i and s_i is a neighboring vertex of u_1 in the current tree. The modification to the tree is to merge u_2 and u_3 with u_1 and to update the power level at u_1. The resulting tree is shown in Fig. 2b.

Case 2. If s_i is not a neighboring vertex of any power vertex derived from it, let u_j be the power vertex with weight $\omega(u_j) = \max\{\omega(u_1), \omega(u_2), \ldots, \omega(u_r)\}$. Merge all the other power vertices $u_1, u_2, \ldots, u_{j-1}, u_{j+1}$ and u_r to u_j. This procedure continues until there is at most one power vertex left in the resulting tree derived from every mobile vertex.

We now remove the restriction and assume that u_i is an ancestor of u_j in the tree with $i \neq j$. For this case, the tree is modified as follows: The edge $(parent(u_j), u_j)$ is removed from the tree, vertices u_j and u_i are merged into a single power vertex, and the weight assigned to the single power vertex power is $\max\{\omega(u_i), \omega(u_j)\}$, where $parent(u)$ is the parent of vertex u in the tree.

It is obvious that the above merging is valid, which can be seen as follows: Assume that s is the mobile node from which both u_i and u_j are derived and $\omega(u_j) < \omega(u_i)$. In the tree, if s works at its power level corresponding to u_l to transmit a message, every mobile vertex in $N_T(u_l)$ is able to receive the message, $l = i, j$. Then, if s works at the power level of u_i to transmit a message, every mobile vertex in $N_T(u_i) \cup N_T(u_j)$ is able to receive the message. The merging operation thus is valid.

Let T' be the resulting tree derived from T_{app} after a series of the above merging operations. The weighted sum of the vertices in T' is bounded as follows: Let $W(T_{app})$ be the weighted sum of the vertices in T_{app} and U the set of power vertices in T_{app}, then $W(T_{app}) = \sum_{u \in U} \omega(u)$ because the weight $\omega(s)$ assigned to each mobile vertex s in T_{app} is zero. Since T' is obtained after a series of merging operations on T_{app}, $W(T') \le W(T_{app})$.

3.3.2 Correction Violation 2

Following the construction of T', it is now known that for every mobile vertex there is at most one power vertex in T' derived from it. However, T' may still not be a valid multicast tree because a power vertex may not be derived

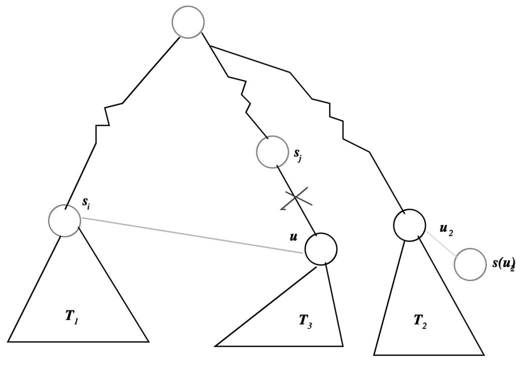


Fig. 3. Link a mobile vertex with its power vertex.

from any of its neighboring mobile vertices. To correct this type of violation, T' must be modified such that every power vertex in the resulting tree is derived from one of its neighboring mobile vertices in the following: Assume that u is a power vertex in T' derived from s_i. 1) If $s_i \in N_T(u)$, it is done. 2) If both s_i and u are in T' but $s_i \notin N_T(u)$, T' is modified as follows: Let (s_j, u) be an edge in the path in the current tree from the root to u, delete edge (s_j, u) from and add an edge (s_i, u) to the current tree. As a result, s_i is a neighboring vertex of u in the resulting tree. 3) Otherwise (the power vertex u is derived from a mobile vertex s_i but s_i is not in the tree), the mobile vertex s_i and an edge (s_i, u) between u and s_i are added to the current tree. The validation of this modification is justified as follows: If $s_i \notin N_T(u)$ and s_i is not in T', then adding s_i and the edge (s_i, u) to the tree is valid because u is derived from s_i. If both s_i and u are in T' but $s_i \notin N_T(u)$, adding the edge (s_i, u) to the tree is valid because u is one of the power vertices derived from s_i, and all the mobile vertices in $N_T(u)$ are within the transmission range of s_i if s_i works at a power level corresponding to vertex u.

Fig. 3 illustrates the above modification. If u is derived from s_i, a new edge between s_i and u is added to and an existing edge (s_j, u) is removed from the tree. For a power vertex u_2 derived from a mobile vertex $s(u_2)$ that is not in the tree initially, the mobile vertex $s(u_2)$ and an edge $(u_2, s(u_2))$ are added to the tree. The weight $\omega(s(u_2))$ assigned to $s(u_2)$ is zero.

Let T'' be the resulting tree after a series of above transformations to T'. T'' has the following properties: 1) Every power vertex is derived from one of its neighboring mobile vertices. 2) $W(T'') = W(T') \leq W(T_{app})$ since each modification does not update the weights of the existing vertices in T' and the weight assigned to every newly added mobile vertex is zero, so, $W(T'') = W(T')$. Meanwhile, $W(T') \leq W(T_{app})$. Therefore, $W(T'') = W(T') \leq W(T_{app})$.

3.3.3 Correction Violation 3

T'' may still not be a valid multicast tree because, for a given mobile vertex s, T'' may contain such tree edges that each of them is from s to a power vertex u, and 1) either u is derived from s, but none of the mobile vertices derived from the other neighboring power vertices of s is within the transmission range of s when it works at the power level corresponding to u; or 2) none of the neighboring power vertices of s (not including the parent of s) is derived from it. Thus, T'' needs to be further transformed in order to become a valid multicast tree. Note that, compared with previously introduced transformations, the weighted sum of the vertices in the resulting tree after this transformation may increase. However, we later show that the total extra weighted sum of the vertices in the resulting tree is no more than that in T''. We perform the transformation to T'' as follows: Associated with each mobile vertex in T'', there are three stages, which are represented by three colors white, gray, and black. Initially, all mobile vertices in T'' are colored *white*. When a mobile vertex is visited for the first time, it is colored *gray*. After finishing the visit to a mobile vertex, the mobile vertex is colored *black*. In this case, the relationship between either the mobile vertex and its parent or the mobile vertex and its children in the resulting tree is determined already. Let Q be a queue containing all gray vertices, which is empty initially. Q contains only mobile vertices clearly.

We traverse T'', starting from its root (a mobile vertex), using the Breadth-First Search technique. Color the root vertex with gray and add it to Q. If Q is empty, the transformation terminates; otherwise, pick the head s of Q and proceed as follows:

1. If s is a leaf vertex, the gray vertex that colored s initially becomes the parent of s in the resulting tree. s is colored black and removed from Q for any further consideration.
2. If s is the root vertex, it is its parent by itself in the resulting tree, i.e., $parent(s) = s$.
3. Otherwise, the parent $parent(s)$ of s already exists and s can be dealt by two subcases: Case a and Case b.

Case a. If a neighboring vertex u of s in the tree is derived from s and $u \neq parent(s)$, then s becomes the parent of u by setting $parent(u) = s$, and the power of u is now set to be $\max_{u' \in N_T(s) - \{parent(s)\}}\{\omega(u')\}$. It is already known that every power vertex u' in the tree is derived from a mobile vertex $s(u')$, following the construction of T''. Color $s(u')$ with gray and add it to Q, set u to be the parent of $s(u')$ for all $u' \in N_T(s) - \{u, parent(s)\}$, and color s'' with gray and add it to Q for every $s'' \in N_T(u) - \{s\}$. In other words, add an edge $(u, s(u'))$ to and remove the edge (s, u') from the tree for every power vertex $u' \in N_T(s) - \{u, parent(s)\}$. Color s with black and remove it from Q. Fig. 4 illustrates this modification, where $N_T(s) - \{parent(s)\} = \{u, u_1, u_2, u_3\}$. The existing edges (s, u_1), (s, u_2), and (s, u_3) are removed from and the new edges $(u, s(u_1))$, $(u, s(u_2))$, and $(u, s(u_3))$ are added to the tree.

Since the power at the power vertex u derived from s is updated, the resulting tree may take extra power consumption for the updating. Denote by $\Delta(s)$ the extra power needed for the update related to s, i.e., the extra cost associated with s when its color changes from gray to black. Then, $\Delta(s) = 0$ if the power $\omega(u)$ $(= \max_{u' \in N_T(s) - \{parent(s)\}}\{\omega(u')\})$ is the maximum one among its neighboring vertices (except its parent); otherwise, $\Delta(s) = \max_{u' \in N_T(s) - \{parent(s)\}}\{\omega(u')\} - \omega(u)$.

We now justify that the above transformation is valid. Assume that u is derived from s, and s is gray already. Let

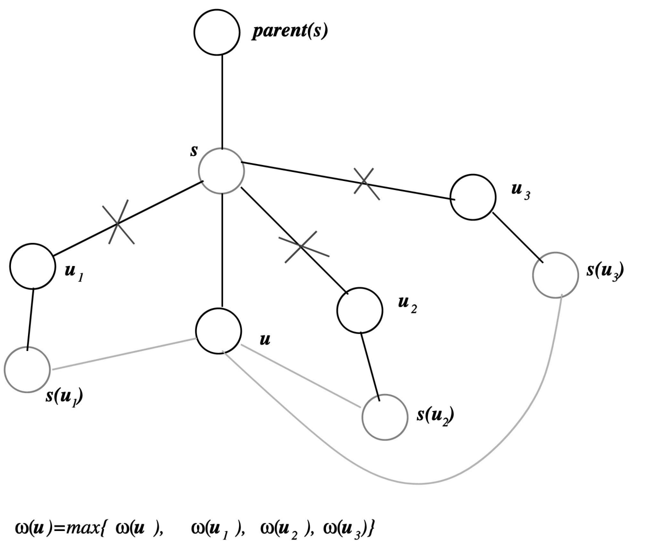


Fig. 4. u is a power vertex derived from s.

$N_T(s) - \{parent(s)\} = \{u, u_1, u_2, \ldots, u_r\}$, where u_i is derived from $s(u_i)$, $1 \leq i \leq r$. Let

$$w(u_0) = \max_{u' \in N_T(s) - \{parent(s)\}} \{\omega(u')\}.$$

Then, s is within the transmission range of $s(u_i)$ for each power vertex u_i, $1 \leq i \leq r$. Following the power level symmetry between s and $s(u_0)$ due to that they are within the transmission ranges of each other, s has a power level with power $\omega(u_0)$. When s works at the power level with power $\omega(u_0)$, every $s(u_i)$ is within its transmission range for all i, $1 \leq i \leq r$.

Case b. None of its neighboring vertices except $parent(s)$ is derived from s. Then, a power vertex $u(s)$ for s is added to the tree and the power at $u(s)$ is set to be $\max_{u' \in N_T(s) - \{parent(s)\}} \{\omega(u')\}$. An edge $(s, u(s))$ is added to the tree as well. Thus, s becomes the parent of $u(s)$. Note that the vertex $u(s)$ must not be in the tree previously. Otherwise, it is a neighboring vertex of s already, following the construction of T''. Delete the existing edge (s, u') from and add a new edge $(u(s), s(u'))$ to the tree for every $u' \in N_T(s) - \{parent(s)\}$. The power vertex $u(s)$ now is the parent of $s(u')$ for every $u' \in N_T(s) - \{parent(s)\}$. Color $s(u')$ with gray and add it to Q for every $u' \in N_T(s) - \{parent(s)\}$. Color s with black and remove it from Q. As a result, $\Delta(s) = \max_{u' \in N_T(s) - \{parent(s)\}} \{\omega(u')\}$. The validation of this transformation can be justified as we did for Case a omitted.

Fig. 5 illustrates Case b, where $u(s)$ is a newly added power vertex, u_1, u_2 and u_3 are the neighboring vertices of s, and none of them is derived from it. The existing edges (s, u_1), (s, u_2), and (s, u_3) are removed from and the new edges $(u(s), s(u_1))$, $(u(s), s(u_2))$, and $(u(s), s(u_3))$ are added to the tree.

Let T''' be the resulting tree. Clearly, T''' is a valid multicast tree, which meets 1) and 2) in Lemma 3. It has the following property:

Lemma 4. *Let T''' be the resulting tree after a series of modifications to T''. Then, $W(T''') \leq 2W(T'') \leq 2W(T_{app})$.*

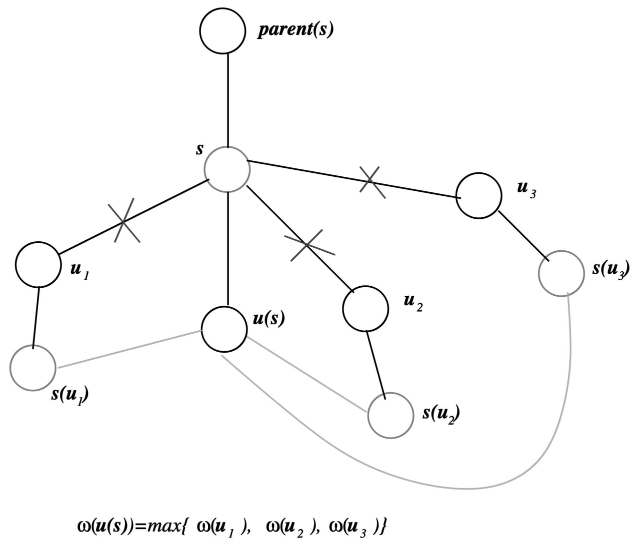


Fig. 5. The power vertex of s is not in the tree.

Proof. Let U and MS be the sets of power vertices and mobile vertices in T'', respectively. We already know that the weight of each mobile vertex in either T''' or T'' is zero. Therefore, we only pay attention on the weight of each power vertex in T'' or T'''. We aim to show that $sum_{s \in MS} \Delta(s) \leq \sum_{u \in U} \omega(u)$. Recall that $\Delta(s)$ is the extra cost associated with s for the transformation from T'' to T'''. What we need is to show that the weight of a power vertex in T'' is used as the extra cost for a mobile vertex at most once in the construction of T''' as follows:

Given two mobile vertices s_i and s_j in T'', if $N_T(s_i) \cap N_T(s_j) = \emptyset$, then $\Delta(s_i)$ and $\Delta(s_j)$ will not make use of a weight from the same power vertex in U; otherwise, it can be seen that $N_T(s_i) \cap N_T(s_j) = \{u'\}$ because T'' is a tree. The rest is to show that only $\Delta(s_i)$ or $\Delta(s_j)$ but not both of them uses the weight of u'. This is illustrated in Fig. 6.

Among all neighboring vertices of u in T'' including the vertex s_i, assume that s_i is colored with gray first, which means that the parent $parent(s_i)$ of s_i in T''' has already been determined. Now, assume that the power vertex u is derived from s_i and s_i is being visited. If u is not in T'', a new power vertex $u(s_i)$ in T''' for s_i is added. The power level of u (or $u(s_i)$) is either its current power level or set to be a power level with

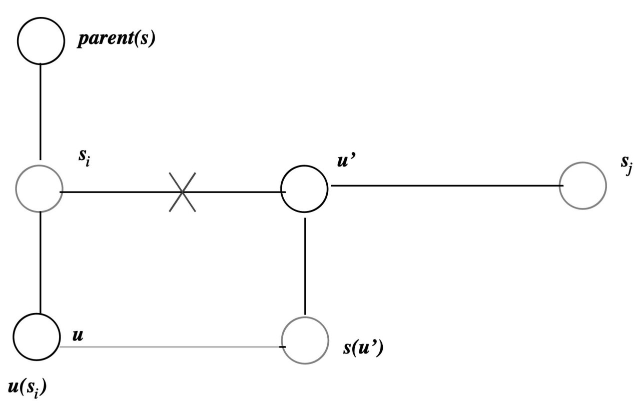


Fig. 6. An illustration of the proof.

power $\omega(u''_0) = \max_{u_x \in N_T(s_i)-\{parent(s_i)\}}\{\omega(u_x)\}$. If $u''_0 = u$ or $u''_0 \neq u'$, then, $\Delta(s_i)$ does not use the weight of u'. It is done. Otherwise, $\Delta(s_i) = \omega(u')$. Now, we need to show that the weight $\omega(u')$ of u' will not be used again by any other mobile vertex $s_j \in N_T(u') - \{s_i\}$. Following the construction of T''', there must be a mobile vertex $s(u')$ in T'' for u' and both $(u', s(u'))$ and (u', s_j) are the tree edges. The tree edge (s_i, u') is no longer existent when s_i is colored with black, $u(s_i)$ is being visited, or $s(u')$ is colored with gray and s_j is still white. When $s(u')$ is visited for the first time (colored with gray), the extra power $\Delta(s(u'))$ associated with it is either $\Delta(s(u')) = 0$ or $\Delta(s(u')) = \max_{u''' \in N_T(s(u'))-\{u,u'\}}\{\omega(u''')\} - \omega(s(u'))$, and $s(u')$ is the parent of s_j in the resulting tree. Thus, $\Delta(s_j)$ does not use the weight $\omega(u')$ of u', which means that the weight of each power vertex in U is used as an extra amount of power for a mobile vertex at most once, in the construction of T'''. Therefore, $\sum_{s\in WS}\Delta(s) \leq \sum_{u\in U}\omega(u) = W(T'') = W(T') \leq W(T_{app})$, and the weighted sum of the vertices in $T'''W(T''') = \sum_{s\in WS}\Delta(s) + \sum_{u\in U}\omega(u) \leq 2\sum_{u\in U}\omega(u) \leq 2W(T_{app})$. □

In summary, the proposed algorithm is presented below:

Algorithm Mini_Multicast_Tree(N, L, k)

begin

1. Construct a node-weighed auxiliary, undirected graph $G(V, E, \omega)$.
2. Find an approximate, minimum node-weighted Steiner tree T_{app} in G using the algorithm in [16].
3. Modify T_{app} by merging those power vertices derived from a single mobile vertex into a power vertex. Let T' be the resulting tree.
4. Modify T' such that every power vertex is derived from one of its neighboring vertices. Let T'' be the resulting tree.
5. Modify T'' to make it a valid multicast tree T'''.
6. Set the power level for each node in T''', using the information provided by T'''.

end

We thus have the following theorem:

Theorem 1. *Given a symmetric wireless ad hoc network $M(N, A)$ with the power at each node being finitely adjustable, a source node, and a set D of destination nodes, there is an approximation algorithm for finding a minimum-energy multicast tree rooted at the source node and spanning the nodes in D, which delivers a solution (the sum of transmission power at nonleaf nodes in the tree) within $4 \ln K$ times of the optimum. The algorithm takes $O(kK^2n^2)$ time, where $K = |D|$, $k = \max_{i=1}^{n}\{l_i\}$, and l_i is the number of power levels at node v_i in N, $1 \leq i \leq n$.*

Proof. Following algorithm `Mini_Multicast_Tree`, Step 1 takes $O(kn^2)$ time because G contains $(k+1)n$ vertices and kn^2 edges at most. Step 2 is the dominant step in terms of running time, which takes $O(K^2(kn^2 + (k+1)n\log(k+1)n)) = O(kK^2n^2)$ time by Lemma 1 because G contains $(k+1)n$ vertices and kn^2 edges and there are K destination vertices. From Step 3 to Step 5 each takes $O(n^2)$ time by examining the set of neighboring vertices of each vertex in a tree, while the tree contains $O(n)$ vertices and each vertex has $O(n)$ neighboring vertices. Step 6 takes $O(n)$ time by checking each tree edge. Thus, the entire algorithm takes $O(kK^2n^2)$ time. Note that the dominant step of the proposed algorithm is Step 2, which is to find an approximate, minimum node-weighted Steiner tree in a node-weighted undirected graph. Any improvement on the running time at this step will improve the running time of the proposed algorithm.

The approximation ratio of the proposed algorithm is analyzed as follows: It is already known that $W(T''') \leq 2W(T_{app})$ by Lemma 4, while $W(T_{app}) \leq 2\ln KW(T_{st}^{opt})$ by the approximation algorithm in [16]. Thus, $W(T''') \leq 4\ln KW(T_{st}^{opt}) \leq 4\ln KW(T)$, where the weighted sum of the vertices in T is equal to the sum of the transmission power at the nodes in the minimum-energy multicast tree in the symmetric wireless ad hoc network M and $K = |D|$. □

4 Approximation Algorithm with Infinitely Adjustable Power

So far, we have assumed that the transmission power at each node is finitely adjustable. In this section, we remove this constraint by assuming that the transmission power at each node is infinitely adjustable. Under this latter model, we provide two approximation algorithms for the problem, which trade-off the running time of the proposed algorithm and accuracy of the solution obtained.

4.1 An Approximation Algorithm Depending on Node Power

In this section, we devise an approximation algorithm whose running time depends on the amount of power at nodes. The idea behind the proposed algorithm follows: For a node $v_i \in N$, the range of its battery power is partitioned into a number of power intervals, and each of these power intervals corresponds to a power level. Let x be the minimum integer such that $2^x \geq p_{\min}(v_i)$ and y the minimum integer such that $p_{\max}(v_i) \leq 2^y$. Then, the power at node v_i ranged from $p_{\min}(v_i)$ to $p_{max}(v_i)$ is divided into $y - x + 1$ intervals $[p_{\min}(v_i), 2^x), [2^x, 2^{x+1}), \ldots, [2^{y-1}, p_{\max}(v_i)]$. Therefore, there are $y - x + 1$ power levels at node v_i and each power level corresponds to a specific range of power. The problem with the power at each node being infinitely adjustable then becomes the problem with the power at each node being finitely adjustable, and this latter case has already been discussed in the previous section. Therefore, we have the following theorem:

Theorem 2. *Given a symmetric wireless ad hoc network $M(N, A)$ with the power at each node being infinitely adjustable, a source node, and a set D of destination nodes, there is an approximation algorithm for finding a minimum-energy multicast tree rooted at the source node and spanning the nodes in D, which delivers a solution (the sum of transmission power at nonleaf nodes in the tree) within $8 \ln K$ times of the optimum. The algorithm takes $O(K^2n^2 \log(p_{\max}/p_{\min}))$ time, where $p_{\max}$ and $p_{\min}$ are the maximum and minimum power at nodes in N and $K = |D|$.*

Proof. It is obvious that the maximum number of power levels at a node in M is $k = \lceil \log(p_{\max}/p_{\min}) \rceil$. Thus, the analysis of the time complexity is straightforward, omitted.

In what follows, we analyze the approximation ratio of the proposed algorithm. Assume that in the optimal solution of the problem, the power at node v_i is pw_i. Let $2^{p_i} \leq pw_i \leq 2^{p_i+1}$. Then, the range of the battery power at node v_i is partitioned into a number of power intervals and each of them corresponds to a power level, the power interval (or the corresponding power node) of pw_i is either $[2^{p_i}, 2^{p_i+1})$ or $[2^{p_i+1}, 2^{p_i+2})$. Assume that vertex u_{i,j_i} is the corresponding power vertex of v_i in the minimum node-weighted Steiner tree in G, i.e., the transmission power at v_i is $\omega(u_{i,j_i})$, then $\omega(u_{i,j_i}) \leq 2^{p_i+1}$. Thus, the approximate solution of the minimum node-weighted Steiner tree problem is $4 \ln K \sum_{i=1}^{n} \omega(u_{i,j_i}) \leq 4 \ln K \sum_{i=1}^{n} 2^{p_i+1} = 8 \ln K \sum_{i=1}^{n} 2^{p_i}$, since $\sum_{i=1}^{n} pw_i \geq \sum_{i=1}^{n} 2^{p_i}$. Therefore, the approximate solution is $8 \ln K$ times of the optimum. □

4.2 An Approximation Algorithm Independent of Node Power

In this section, we provide an approximation algorithm whose running time is independent of the amount of power at nodes. Associated with every node $v \in V$, let $N(v)$ be the set of neighboring nodes of v in the network, i.e., the set of the mobile nodes that are within the transmission range of v when v works at its maximum power level. Clearly, there are at most $|N(v)|$ power levels at v, and the power at the power level derived from its neighboring node u ($u \in N(v)$) is at least $d_{u,v}^{\alpha}$, and $|N(v)| \leq n - 1$. In other words, the maximum number of power levels at a node is $k = n - 1$. Thus, the problem with the power at each node being infinitely adjustable now becomes the problem with the power at each node being finitely adjustable. We thus have the following theorem:

Theorem 3. *Given a symmetric wireless ad hoc network $M(N, A)$ with the power at each node being infinitely adjustable, a source node, and a set D of destination nodes, there is an approximation algorithm for finding a minimum-energy multicast tree rooted at the source node and spanning the nodes in D, which delivers a solution (the sum of transmission power at nonleaf nodes in the tree) within $4 \ln K$ times of the optimum. The algorithm takes $O(K^2 n^3)$ time, where $K = |D|$.*

Proof. The analysis of the running time complexity and approximation ratio of the proposed algorithm is similar to the one given in the proof body of Theorem 2, omitted. □

5 Conclusions

In this paper, we have considered the minimum-energy multicast tree problem in a symmetric wireless ad hoc network by devising the first approximation algorithm with an approximation ratio of $4 \ln K$, where K is the number of destination nodes in a multicast session. The approximation of the solution delivered by the proposed algorithm is within a constant factor of the best-possible approximation achievable in polynomial time unless P = NP.

Acknowledgments

The authors would like to thank the anonymous referees for their constructive comments and valuable suggestions which have helped improve the quality and presentation of the paper.

References

[1] M. Agarwal, J.H. Cho, L. Gao, and J. Wu, “Energy Efficient Broadcast in Wireless Ad Hoc Networks with Hitch-Hiking,” *Proc. INFOCOM ’04 Conf.*, 2004.

[2] D.J. Baker and A. Ephremides, “The Architectural Organization of a Mobile Radio Network Via Distributed Algorithm,” *IEEE Trans. Comm.*, vol. 29, pp. 56-73, 1981.

[3] S. Banerjee, A. Midra, J. Yeo, and A. Agarwala, “Energy-Efficient Broadcast and Multicast Trees for Reliable Wireless Communication,” *Proc. IEEE Wireless Comm. and Networking Conf.*, 2003.

[4] F. Bian, A. Goel, C.S. Raghavendra, and X. Li, “Energy-Efficient Broadcasting in Wireless Ad Hoc Networks: Lower Bounds and Algorithms,” *J. Interconnection Networks*, vol. 3, pp. 149-166, 2002.

[5] M. Cagalj, J.-P. Hubaux, and C. Enz, “Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues,” *Proc. MobiCom ’02 Conf.*, 2002.

[6] J. Cartigny, D. Simplot, and I. Stojmenovic, “Localized Minimum-Energy Broadcasting in Ad Hoc Networks,” *Proc. INFOCOM ’03 Conf.*, 2003.

[7] J.-H. Chang and L. Tassiulas, “Fast Approximate Algorithms for Maximum Lifetime Routing in Wireless Ad-Hoc Networks,” *Proc. IFIP-TC6/European Commission Int’l Conf.*, pp. 702-713, 2000.

[8] I. Chlamtac and A. Farago, “A New Approach to the Design and Analysis of Peer-to-Peer Mobile Networks,” *Wireless Networks*, vol. 5, pp. 149-156, 1999.

[9] A.K. Das, R.J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, “Minimum Power Broadcast Trees for Wireless Networks: Integer Programming Formulations,” *Proc. INFOCOM ’03 Conf.*, pp. 1001-1010, 2003.

[10] A. Ephremides, J.E. Wieselthier, and D.J. Baker, “A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling,” *Proc. IEEE*, vol. 75, pp. 56-58, 1987.

[11] M. Gerharz, C. de Waal, P. Martini, and P. James, “A Cooperative Nearest Neighbours Topology Control Algorithm for Wireless Ad Hoc Networks,” *Proc. 12th Int’l Conf. Computer Comm. and Networks*, pp. 412-417, 2003.

[12] S. Guha and S. Khuller, “Improved Methods for Approximating Node Weighted Steiner Trees and Connected Dominating Sets,” *Information and Computation*, vol. 150, pp. 57-74, 1999.

[13] C.E. Jones, K.M. Sivalingam, P. Agrawal, and J.C. Chen, “A Survey of Energy Efficient Network Protocols for Wireless Networks,” *Wireless Networks*, vol. 7, pp. 343-358, 2001.

[14] I. Kang and R. Poovendran, “Maximizing Static Network Lifetime of Wireless Broadcast Ad Hoc Networks,” *Proc. Implicit Computational Complexity Workshop*, 2003.

[15] V. Kawadia and P.R. Kumar, “Power Control and Clustering in Ad Hoc Networks,” *Proc. INFOCOM ’03 Conf.*, 2003.

[16] P.N. Klein and R. Ravi, “A Nearly Best-Possible Approximation Algorithm for Node-Weighted Steiner Trees,” *J. Algorithms*, vol. 19, pp. 104-114, 1995.

[17] N. Li, J.C. Hou, and L. Sha, “Design and Analysis of an MST-Based Topology Control Algorithm,” *Proc. INFOCOM ’03 Conf.*, 2003.

[18] W. Liang, “Constructing Minimum-Energy Broadcast Trees in Wireless Ad Hoc Networks,” *Proc. MOBIHOC ’02 Conf.*, pp. 112-122, 2002.

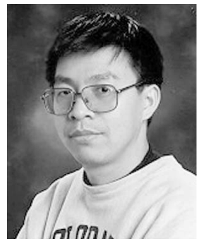
[19] E.L. Lloyd, R. Liu, M.V. Marathe, R. Ramanathan, and S.S. Ravi, “Algorithmic Aspects of Topology Control Problems for Ad Hoc Networks,” *Proc. MOBIHOC ’02 Conf.*, pp. 123-134, 2002.

[20] C. Lund and M. Yannakakis, “On the Hardness of Approximating Minimization Problems,” *J. ACM*, vol. 41, pp. 960-981, 1994.

[21] R. Ramanathan and R. Rosales-Hain, “Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment,” *Proc. INFOCOM ’00 Conf.*, 2000.

[22] V. Rodoplu and T.H. Meng, “Minimum Energy Mobile Wireless Networks,” *Proc. Implicit Computational Complexity Workshop*, vol. 3, pp. 1633-1639, 1998.

[23] A. Srinivas and E. Modiano, "Minimum Energy Disjoint Path Routing in Wireless Ad-Hoc Networks," *Proc. MOBICOM '03 Conf.*, 2003.
[24] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proc. MOBICOM '98 Conf.*, pp. 181-190, 1998.
[25] P.-J. Wan, G. Calinescu, and X.-Y. Li, "Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks," *Proc. INFOCOM '01 Conf.*, 2001.
[26] R. Wattenhofer, L. Li, P. Bahl, and Y-M Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," *Proc. INFOCOM '01 Conf.*, 2001.
[27] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "On Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Proc. INFOCOM '00 Conf.*, 2000.
[28] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "Resource Management in Energy-Limited, Bandwidth-Limited, Transceiver-Limited Wireless Networks for Session-Based Multicasting," *Computer Networks*, vol. 39, pp. 113-131, 2002.



Weifa Liang received the BSc degree from Wuhan University, China, in 1984, the ME degree from the University of Science and Technology of China in 1989, and the PhD degree from the Australian National University in 1998, all in computer science. He is currently a senior lecturer in the Department of Computer Science at the Australian National University. His research interests include the design of energy-efficient routing protocols for wireless ad hoc networks, routing protocol design for WDM optical networks, design and analysis of parallel and distributed algorithms, data warehousing and OLAP, query optimization, and graph theory. He is a senior member of the IEEE.