MEGCOM: Minimum-Energy Group COMmunication in Multi-hop Wireless Networks

Kai Han, Liu Xiang, Jun Luo, and Yang Liu

***Abstract*—Given the increasing demand from wireless applications, designing energy-efficient group communication protocols is of great importance to multi-hop wireless networks. A group communication session involves a set of member nodes, each of them needs to send a certain number of data packets to all other members. In this paper, we consider the problem of building a shared multicast tree spanning the member nodes in static wireless networks such that the total energy consumption of a group communication session using the shared multicast tree is minimized. Since this problem was proven as NP-complete, we propose, under our Min-Energy Group COMmunication (MEGCOM) framework, three distributed approximation algorithms with provable approximation ratios. When the transmission power of each wireless node is fixed and identical, our first two algorithms have the approximation ratios of $\mathcal{O}(\ln(\Delta+1))$ and $\mathcal{O}(1)$, respectively, where Δ is the maximum node degree in the network. When the transmission power of each wireless node is adjustable, our third algorithm again delivers a constant approximation ratio. We also use extensive simulations to verify the performance of our algorithms.**

***Index Terms*—Energy Efficiency, Group Communication, Multicast, Approximation Algorithms**

I. Introduction

Group communication (or all-to-all multicasting) is a very important primitive for distributed systems, as a large body of applications including, among others, social networking, online meeting, network gaming, resource sharing, and data management are heavily relying on its service [1], [2]. This continues to be the case for wireless systems. For example, wireless users may leverage group communication for video conferences or online role-playing games, and a group of rescuers or soldiers may use group communication to share their local information to facilitate effective decision making in crisis management applications. Due to its prominent importance, the great interests on group communiction and related problems have never receded ever since the inception of wireless networks (e.g., [3]–[6]). In a traditional setting, investigations on group communication focus on the system reliability in the face of network failures or group member changes [2], [7]–[9]. As the recent booming of multi-hop wireless networks for various purposes (e.g., wireless mesh networks to allow Internet access in remote areas or wireless sensor networks for habitat monitoring) further demands group communication to work on top of such networks (e.g., [10]), group communication is facing a new challenge: energy constraints for wireless nodes. As it is generally believed that the continuous development of the wireless technology will result in more and more wireless applications making use of group communication primitive (e.g., mobile social networking and gaming), designing energy-efficient group communication protocols has become increasingly imperative.

One naive way for designing an energy-efficient group communication protocol is to employ the existing one-to-many multicast protocols. In other words, one may construct an energy-efficient one-to-many multicast tree for each group member. While this approach may leverage on the abundant research results on building minimum-energy (one-to-many) multicast tree in wireless networks (e.g., [11]–[14]), it is not exactly practical for applications demanding group communication support. More specifically, applications such as social networking or online gaming may involve more than one sender (every node could potentially be a sender in its own multicast session); this is in stark contrast to, for example, video streaming where only one session (thus one sender) exists. Simply combining multiple one-to-many multicast sessions to implement a group communication service would lead to a huge computation and communication overhead in maintaining all the resulting multicast trees. Therefore, a more practical way is to construct one tree spanning the group members as a shared multicast tree [3].

Surprisingly, although the minimum energy one-to-many multicasting problem has been studied extensively, there is little work on min-energy group communication in multi-hop wireless networks. Similar to the min-energy one-to-many multicasting problem, the min-energy shared tree multicasting problem is also NP-hard [3]. Moreover, the problem becomes more challenging because an optimal tree also depends on how many packets each group member is about to multicast. Liang *et al.* [3] attempted to tackle this problem by proposing several approximation algorithms. However, the resulting approximation ratios are far from satisfactory. For example, their approximation ratios are in the same order of $|M|$, where M is set of group members.

In this paper, we study the Minimun-Energy Group COMmunication (MEGCOM) framework in static multi-hop wireless networks. In particular, we seek to minimize the total energy consumption of a group communication session relying on a shared multicast tree. As the induced Minimum-Energy

This work is supported in part by the AcRF Tier 2 Grant ARC 15/11, the National Science Foundation of China under Grant no. 61103007 and 61003052, and the Program for New Century Excellent Talents of the Ministry of Education of China under Grant no. NCET-12-0692.

Kai Han is with School of Computer Engineering, Nanyang Technological University, Singapore, and also with School of Computer Science, Zhongyuan University of Technology, China. E-mail: hankai@gmail.com.

Liu Xiang and Jun Luo are with School of Computer Engineering, Nanyang Technological University, Singapore. E-mail: {xi0001iu, junluo}@ntu.edu.sg

Yang Liu is with School of Information Sciences and Engineering, Henan University of Technology, China. E-mail: enjoyang@gmail.com

All-to-All Multicasting (MEAAM) problem is NP-complete, we propose three distributed approximation algorithms with guaranteed approximation ratios for MEAAM, under both cases of fixed transmission power and adjustable transmission power. Our algorithms exponentially improve the best-known results of quadratic approximation ratio [3] to constant approximation ratio. Specifically:

1) When the transmission power of each wireless node is fixed and identical, our first algorithm has an approximation ratio of $4\ln(\Delta+1)+7$, where Δ is the maximum node degree in the network.
2) When the transmission power of each wireless node is fixed and identical, our second algorithm delivers a 13-approximation to MEAAM.
3) When the transmission power of each wireless node is adjustable, we prove the existence of a constant approximation algorithm for MEAAM, and we also show that a straightforward algorithm leads to a constant approximation ratio of 145.

In the remaining of the paper, we first briefly review the literature in Sec. II. After formally defining the models and problems in Sec. III, we present and analyze our three algorithms in Sec. IV, V, and VI respectively. We also perform extensive simulations of our algorithm in Sec. VII, and we briefly discuss some related issues about MEAAM in Sec. VIII, before finally concluding our paper in Sec. IX. For clarity, important system variables with their acronyms and descriptions are presented in Table I.

TABLE I
SYMBOLS AND NOTATIONS

Notation	Description
G	The graph representing a wireless network
V	Node set of G
E	Edge set of G
M	Set of group members
ε_s	The energy consumption for transmitting a data packet by any node
ε_r	The energy consumption for receiving a data packet by any node
$p(u)$	The number of data packets originated from the group member $u \in M$
k	The total number of data packets originated from all group members in M
Δ	Maximum node degree of G
T_{opt}	The optimal multicast tree for min-energy group communication
$nd(T)$	Set of nodes in tree T
$lv(T)$	Set of degree-one nodes in tree T
$in(T)$	Set of nodes in tree T with degree more than one
$nb(u,G)$	Set of neighboring nodes of u in G
$nb^+(u,G)$	The union of $nb(u,G)$ and $\{u\}$
$nb2(u,G)$	Set of nodes in G within two hops of u (including u itself)
$\Psi(T)$	The total energy consumption of realizing a group communication session using the multicast tree T
T_I	The multicast tree spanning the nodes in M such that $\|in(T_I)\|$ is minimized
T_I^-	A tree constructed from T_I by pruning all the degree-one nodes in T_I
B	The buddy set which is a subset of V and $nb(v,G) \cap M \neq \emptyset$ for $\forall v \in B$.
C	The guardian set found by MEGCOM-LFP
T_S^*	Optimal Steiner tree spanning the nodes in C
T_A	The multicast tree found by MEGCOM-LFP.
C'	The guardian set found by MEGCOM-CFP
$\acute{T}_S^*$	Optimal Steiner tree spanning the nodes in C'
T'_A	The multicast tree found by MEGCOM-CFP.
$\lambda(u,T)$	The transmission power for u to reach all its neighoring nodes in tree T
$\Theta(T)$	The summation of $\lambda(u,T)$ for all $u \in in(T)$
T_W	A multicast tree spanning the nodes in M such that $\Theta(T_W)$ is minimized
$T_W^{(u)}$	Euclidean minimum spanning tree of the nodes in $\{u\} \cup nb(u,T_W)$ for any $u \in in(T_W)$
$\zeta(T)$	The sum of the edge weights of tree T
$\ddot{T}_S$	A 2-approximation Steiner tree in G spanning the nodes in M
$\ddot{T}_S^*$	An optimal Steiner tree in G that spans the nodes in M

II. RELATED WORK

There is a large body of literature on group communication and multicasting, but, given the space limitation, we have to confine our discussions to those aiming at minimizing the energy consumption, but to leave out non-tree-based appraoches such as [15], [16]. As we concentrate on always-active wireless networks in this paper, we also avoid discussing similar topics in duty-cycled wireless sensor networks [17]–[19].

The minimum-energy one-to-many multicasting problem has been studied in [11]–[14]. Wieselthier *et al.* [11] consider a scenario where each node can adjust its transmission power continuously, and propose several greedy heuristics for the minimum-power broadcast/multicast routing problems. Wan *et al.* [12] prove that the heuristics proposed by [11] have linear approximation ratios, and provide several approximation algorithms with constant approximation ratios for the minimum-energy multicasting problem based on the approximate minimum Steiner tree algorithm. Liang [13] considers a scenario in which each wireless node can adjust its transmission power in a discrete fashion, and the communication links are symmetric. He proposes a centralized approximation algorithm with performance ratio $\mathcal{O}(\ln K)$ for building a minimum-energy multicasting tree, where K is the number of destination nodes in a one-to-many multicast session. Li *et al.* [14] consider a case where all nodes have a fixed transmission power and the communication links are asymmetric. They convert the minimum-energy multicasting problem to an instance of the Directed Steiner Tree problem, and present several heuristics.

To the best of our knowledge, the only work that studies the minimum-energy group communication problem is [3]. In [3], Liang *et al.* propose to build a shared multicast tree such that the total energy consumption of realizing a group communication session using the shared tree is minimized. They prove that finding such a shared tree is a NP-complete problem, and use the approximate Steiner tree algorithm proposed by [20] to solve the problem. When the transmission power of each node is fixed and identical, they prove that the approximate minimum Steiner tree has an approximation ratio of $2(|M|+1)$, where M is the set of group members. When the transmission power of each node is adjustable, they prove that the approximate Steiner tree has an approximation ratio of $\Omega(8|M|)$. They also propose a distributed approximation algorithm with an approximation ratio of $\Omega(4|M|^2)$.

III. Preliminaries and Problem Definition

A multi-hop wireless network is modeled by an undirected graph $G = (V, E)$, where V is the set of wireless nodes in the network and E is the set of wireless links. Each node in $v \in V$ has a unique identifer $v.id$. The nodes in V are all equipped with an omni-directional antenna. We assume that the transmission (tx) power of each node can be either identically fixed or continuously adjustable. When the tx power is fixed and identical, we use ε_s to denote the energy consumption of transmitting a data packet by a node. When the tx power is adjustable, we assume that the power is adjusted with respect to links (same as the assumption adopted in [3]), hence the communication between any two nodes is symmetric. In this case, the energy required by any node u to transmit a data packet to another node v can be determined by the Euclidean distance between u and v. Following a very common formula, we define such energy consumption to be $d^{\alpha}_{(u,v)}$, where $d_{(u,v)}$ is the Euclidean distance between u and v and α is a constant (usually between 2 and 4). In either case (fixed power or adjustable power), we assume that each link $(u, v) \in E$ is assigned a weight which is the amount of energy required for u to send a data packet to v (or vise versa). Following other related proposals such as [3], [21], we also assume that the receiving (rx) power is always less than the tx power, and we denote by ε_r the energy consumption of receiving a data packet by any node.

In a group communication session, there exists a set of *group members* $M \subseteq V$, and each node $u \in M$ needs to send $p(u)$ data packets to all other nodes in $M \backslash \{u\}$. We denote by k the sum of the numbers of data packets originated from the group members, i.e., $k = \sum_{v \in M} p(v)$. As we have explained in Sec. I, instead of building $|M|$ multicast trees originated from each node in M, building a shared multicast tree spanning the nodes in M to support the group communication session is more convenient in realistic settings. Therefore, in order to minimize the total energy consumption of the group communication session, we actually aim at solving the following Minimum-Energy All-to-All Multicasting (MEAAM) problem:

> Find an *optimal shared multicast tree* T_{opt} such that the energy consumption of carrying all k packets over T_{opt} is minimized.

The hardness of this problem is immediate from [3]:

Proposition 1: The MEAAM problem is NP-complete.

For convenience of description, we define some other notations here. For any multicast tree T in G, we denote by $nd(T)$ the set of nodes in T, by $lv(T)$ the set of degree-one nodes in T, by $in(T)$ the set of internal nodes (nodes with degree more than one) in T, and by $\Psi(T)$ the total energy consumption of realizing a group communication session using T. For any node $u \in V$, we denote the set of neighboring nodes of u in G by $nb(u, G)$, and let $nb^+(u, G) = nb(u, G) \cup \{u\}$ and $nb2(u, G) = \bigcup_{v \in nb(u,G)} nb^+(v, G)$, where the latter is the set of nodes in G within two-hops of u (including u itself). In the following, we present three distributed algorithms to construct approximate trees for T_{opt}, they respectively achieve a logarithmic approximation ratio for fixed tx power, a constant approximation ratio for fixed tx power, and a constant approximation ratio for adjustable tx power.

IV. MEGCOM-LFP: Logarithmic Approximation for MEAAM with Fixed tx Power

In this section, we propose a distributed approximation algorithm for MEAAM with fixed and identical tx power. A brief introduction on the idea of the algorithm comes first, followed by detailed algorithm descriptions and a performance analysis. We also discuss how to maintain the shared group communication tree in the face of member joining and leaving.

A. Algorithm Principles

Unlike the common wisdom that directly uses a Steiner tree to approximate T_{opt}, our algorithm has two stages (with the first one having two sub-stages) to construct a T_A that approximates T_{opt}:

- S1-a: Identify a *buddy set* $B \subseteq V$ such that, for any $v \in B$, $nb(v, G) \cap M \neq \emptyset$.
- S1-b: Find a *guardian set* $C \subseteq B$ for M, such that $M \subseteq \bigcup_{v \in C} nb^+(v, G)$.
- S2 : Construct an approximate Steiner tree T_S to span the nodes in C.

As a result, T_A is the union of the member set M, the guardian set C (along with edges it uses), and the Steiner tree T_S that spans C. We term each $v \in C$ a *guarding node* and a member $u \in M \wedge u \in nb(v, G)$ the *guarded member* of v. The tricky part is that we identify C by only searching among B that contains nodes having at least one neighbor in M, which excludes nodes in M that has no neighbor in M. As shown in Figure 1, all non-member nodes having at least one member in its neighborhood belong to B, whereas a member node belongs to B only if it has at least one other member in its neighborhood.

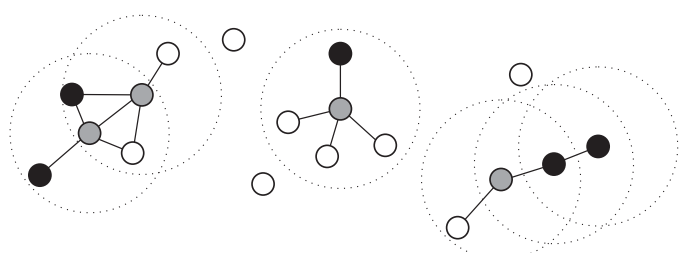


Fig. 1. Examples for the buddy set selection. The black nodes are group members in M, the grey nodes are the non-member nodes chosen for B. Here B contains every node having a concentric tx circle surrounding it; the solid lines indicate edges (or links) implied by those tx circles.

The motivation behind our algorithm is that a good approximation T_A should have more nodes in $lv(T_A)$ and less nodes in $in(T_A)$, as a node in $in(T_A)$ consumes tx power for all packets and rx power for those not originated from it, whereas a node in $lv(T_A)$ consumes only tx power for packets originated from it and otherwise consumes only rx power. We use two examples in Figure 2 to illustrate the advantages of trees constructed by our algorithm compared to straightforward Steiner trees that span M. For example, we can see that the optimal Steiner tree of Fig. 2(a) is the tree T_1 shown in Fig. 2(b), whose total energy consumption is larger than that of the multicast tree T_2 shown in Fig. 2(c). Actually, since all data packets must go through the internal nodes of a multicast tree for group communication and T_1 has more internal nodes than T_2, the energy consumed by T_1's internal nodes is already close to the total energy consumption of T_2, which makes T_1 less energy-efficient. Therefore, we

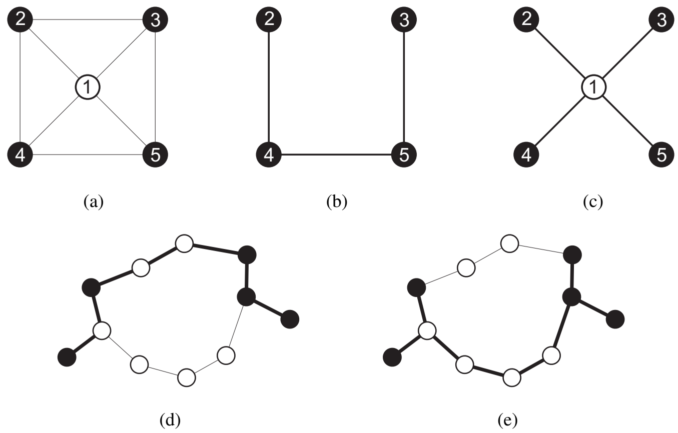


Fig. 2. Comparing our algorithm with Steiner tree based on two instances of MEAAM. In the first example (a)–(c), the set of group members is $\{2, 3, 4, 5\}$. The number of data packets originated from each group member is 50. The tx power of each node is 10. The reception power at each node is 1. The optimal Steiner tree T_1 in (b) leads to a total energy consumption of 5600, while our multicast tree (optimal for this case) T_2 in (c) gives 4800. The energy consumed by the internal nodes of T_1 and T_2 are 4300 and 2200, respectively. In the second example (d)–(e), a Steiner-based heuristic (d) can result in 6 nodes in $in(T_A)$, whereas our multicast tree (e) only has 5 such nodes.

argue that a Steiner-based heuristic may not give a proper approximation to T_{opt}. By far, the best-known centralized approximation algorithm for MEAAM applies a Steiner-based heuristic [3], obtaining an approximation ratio of $2(|M| + 1)$. The performance analysis of our algorithm in Sec. IV-C shows that our algorithm achieves a much better approximation ratio.

B. Algorithm Details

As explained in Sec. IV-A, our algorithm has two stages. In fact, each node can be in different states in each stage, so we let each node $v \in V$ to maintain a variable $v.state$ in order to keep track of the algorithm progress. Initially, every node starts with a state **Inactive**, and the algorithm terminates when all members in M are **Treed** (i.e., join the shared group communication tree) and all **Treed** non-members are spanned by shortest paths to form an approximate Steiner tree. We show the finite state machine for each node in Figure 3. The

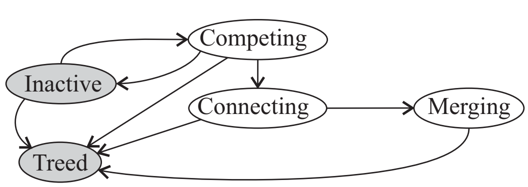


Fig. 3. The finite state machine for each node.

state **Competing** is involved in the guardian set identification (S1), while the two states **Connecting** and **Merging** are for approximate Steiner tree construction (S2).

During the execution of our algorithms, some control messages with specific fields are exchanged between the wireless nodes, and we illustrate the format of a control message in Fig. 4. We can see that a control message consists of four fields: $msgType$, sid, tid and $data$. The $msgType$ is a three-bit long field which indicates the purpose of this message, and its value can be Compete, CNTReq, Decision, Accept and Ack. The meanings of these values are closely related to the process of algorithm execution, and will be shown in the following paragraphs. The sid field denotes the unique id of the node from which the message is originated, and the tid field is the id of the target node. The $data$ field records the information that the sender node needs to convey to the target node.

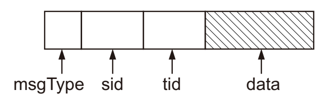


Fig. 4. The fields of a control message.

In the following, we first present the pseudocodes of our algorithm by **Algorithm 1**, then we explain the algorithm execution in details.

As we do not assume that the distributed tree construction process can start synchronously at all nodes, we set a periodically executed *task* SCAN() (lines 1 to 18) to drive the process. This task includes two part: lines 2 to 6 take care of identifying a guardian set C in a **greedy** manner, while other codes are for Steiner tree. In addition, there are four *procedure*s that respond to different events.

At the beginning of the guardian set identification stage (S1), every node v with **Inactive** state records its neighboring group member nodes with the state of **Inactive** or **Competing** in $v.S$, and counts the number of nodes in $v.S$ by $v.ct$ (line 5). Then each node v having at least one member neighbor in $v.S$ implicitly joins the **buddy set** B by changing its state to **Competing** (line 6), as it has the right to compete for acting as a guarding node. As the guarded (member) sets do not intersect if two buddy nodes are more than two hops way from each other, it is sufficient to confine the competitions within the two-hop range for each node (line 6). The node that guards the most wins (line 24); it hence changes its state to **Connecting** (thus enter the second stage, line 25) and also notifies the guarded members to change their states (line 26). When a guarded member v receives such a notification, it records its guarding node in $v.gd$, and changes its state to **Treed** (line 37). As a result, the guarded members will not be counted in later competition since they have the **Treed** state (line 4), and the algorithm has already terminated for them. At the end of S1, all nodes in **Connecting** state implicitly compose the **guardian set** C.

According to the algorithm execution described above, it can be seen that one necessary condition for any node v to be a guardian node is that at least one of v's neighboring node is in M (otherwise v should not enter the **Competing**

Algorithm 1: Finding an approximate multicast tree T_A for the MEAAM problem

Input: The network $G = (V, E)$ and the member set M. For each node $v \in V$, $v.state = \textbf{Inactive}$; $v.ns = 0$
Output: An approximate multicast tree T_A: all the **Treed** nodes and all the marked paths/edges

```
1  task SCAN()                                           /* executed periodically */
2  if v.state = Inactive ∧ ∃u ∈ nb(v,G) : u ∈ M then
3      forall the u ∈ nb(v,G) : u ∈ M do
4          if u.state = Inactive or Competing then
5              v.S ← v.S ∪ {u};  v.ct ← v.ct + 1
6      if v.ct > 0 then  v.state ← Competing;
       SENDMSG('Compete', v.id, u.id, v.ct) to ∀u ∈ nb2(v,G)
7  else if v.state = Connecting then
8      forall the u ∈ M do
9          if u.state = Treed or Connecting then
10             v.S ← v.S ∪ {u.gd};  v.ct ← v.ct + 1
11             v.Path ← v.Path ∪ SP(v ⇝ u.gd)
12     if v.ct = |M| then
13         v.prefer ← arg min_u {SP(v ⇝ u) ∈ v.Path}
14         SENDMSG('CNTReq', v.id, v.prefer.id, v.ld) to v.prefer
15 else if v.state = Treed ∧ v.gd = v then  DISCOVER()
16 else if v.state = Merging then
17     if ¬DISCOVER() then v.state ← Treed
18     else SENDMSG('CNTReq', v.id, v.ld) to v.prefer
19 upon RECVMSG(msgType, sid, tid, data) at v with v.id = tid
20 u ← the node with id sid
21 switch msgType do
22     case 'Compete'
23         if v.ns = |nb2(v,G)| then
24             if v.ct = max_{z ∈ nb2(v,G) ∪ {v}} {z.ct} then
25                 v.state ← Connecting;  v.gd ← v;  v.ld ← v
26                 SENDMSG('Decision', v.id, NULL) to ∀z ∈ v.S
27             else  v.state ← Inactive
28             v.S ← ∅;  v.ct ← 0;  v.ns ← 0
29         else v.ns = v.ns + 1
30     case 'CNTReq'
31         if v.prefer = u then
32             mark the path suggested by the request
33             SENDMSG('Accept', v.id, min{v.ld, data}) to u
34             if v.ld < data then  v.state ← Merging
35             else  v.state ← Treed;  v.ld ← data
36     case 'Decision'
37         v.state ← Treed;  v.gd ← u;  mark the edge to u
38     case 'Accept'
39         mark the accepted path
40         if v.ld ≤ data then  v.state ← Merging
41         else  v.state ← Treed;  v.ld ← data
```

state according to line 2). This is due to the reason that, any degree-one nodes in any group communication tree must be in M, and a guardian node can be intuitively deemed as a "fringe" internal node in a group communication tree: it is adjacent to at least one degree-one node in that tree. As **Algorithm 1** employs a greedy strategy to identify the guardian set, it essentially finds a set of fringe internal nodes whose cardinality is (approximately) minimized.

A guarding node v that enters the Steiner tree construction stage (S2) will periodically check whether other members are guarded and, for an already guarded member u, record u's guarding node in $v.S$ and get the shortest path to u's guarding node (lines 8 to 11). If the targeted multi-hop wireless network has a proactive routing protocol (e.g., OLSR [22]), then a node only needs to check its local routing table. Otherwise routing queries may need to be sent if the network uses a reactive routing protocol (e.g., AODV [23] DSR or [24]). The tree construction starts after all members are either guarded or have entered S2. Our algorithm is similar to the Kruskal-based shortest path heuristic proposed in [25]. The basic idea is to construct a minimum spanning tree over the extended graph $G'(V, E')$, where an edge $(u, v) \in E'$ has a length of $SP(u \leftrightsquigarrow v)$ (the shortest path in G between u and v). As this heuristic is a distributed implementation of the 2-approximation algorithm proposed in [20], the resulting approximation ratio is indeed 2.

The second stage (S2) starts with all the nodes in **Connecting** state, and it gradually finds a tree that spans these nodes. In order to eventually lead to a minimum spanning tree over $G'(V, E')$, only the shortest edge in E' should be added each time. This can be easily distributed initially (when a node is not connected to any other node yet), as the local information on the shortest paths from a node is sufficient for it to decide which edge in E' is to be added (line 13). However, after some nodes are connected to form several fragments, the local information is not sufficient anymore, hence we need a leader (the node in state **Merging**, chosen by the smallest id principle in lines 34 and 40, and recorded by $v.ld$) and certain consensus for obtaining a common preferred node to connect to. This consensus is implemented in the procedure DISCOVER(). Essentially, this procedure returns true if such a preferred node can be identified, then the leader (on behalf of the current fragment) sends a connection request to that node (line 18). Otherwise the algorithm terminates, as the approximate Steiner tree has been constructed (line 17).[1] It is important to note that a connection request is accepted only if both ends consider each other as preferred (lines 14, 18, and 31); this is meant to avoid producing loops.

We further explain the relationship between the algorithm execution described above and the algorithm motivations described in Sec. IV-A as follows. As we have indicated before, the first stage of **Algorithm 1** (S1) essentially identifies all the fringe internal nodes in T_A, while the total number of such fringe internal nodes is approximately minimized. The second stage **Algorithm 1** (S2) intends to find other internal nodes in T_A to connect the fringe internal nodes identified in S1, and the number of such newly added internal nodes in S2 is again approximately minimized thanks to the employed Steiner tree heuristic. In summary, the whole process of **Algorithm 1** focuses on the identification of a minimum set of internal nodes in the group communication tree, which is exactly the motivation described in Sec. IV-A.

C. Performance Analysis

We examine the approximation ratio and the complexity of our algorithm in this section. In particular, we show that T_A resulting from **Algorithm 1** is a $[4 \ln(\Delta + 1) + 7]$-approximation to T_{opt}, where Δ is the maximum node degree of G. Our result is an exponential improvement to the best known result [3]. We also show that the complexity of **Algorithm 1** is polynomial in $|V|$.

[1] Actually, there are some details on pruning overlapping edges of different shortest paths, but we omit them for brevity.

1) Approximation Ratio: Our proof for the approximation ratio is presented in three steps. In ***Lemma** 1*, we provide an upper bound of the number of the internal nodes in T_A. In ***Lemma** 2*, we provide two lower bounds of the total energy consumption of any multicast tree for the MEAAM problem. These bounds are used by ***Theorem** 1* to finally show the approximation ratio of **Algorithm 1**.

***Lemma** 1:* Let T_I be the multicast tree spanning the nodes in M such that the number of internal nodes in T_I (i.e., $|in(T_I)|$) is minimized, we have: $|in(T_A)| \leq [4\ln(\Delta+1)+6]\cdot|in(T_I)|$.

Proof: Let T_S^* and T_S, respectively, be the minimum Steiner tree and the approximate Steiner tree (produced by S2 of **Algorithm 1**) that span the nodes in C. Given the fact that S2 of **Algorithm 1** falls into a minimum spanning tree based Steiner-heuristic, we can know that T_S is a 2-approximation of T_S^* [20]. Note that all the tree links have the same weight in the fixed power case. Hence, we get $|nd(T_S)|-1 \leq 2(|nd(T_S^*)|-1)$. Let $X = M\backslash nd(T_S)$. We have $nd(T_S) \subseteq nd(T_A)$ and $X = nd(T_A)\backslash nd(T_S)$ according to **Algorithm 1**. If $X = \emptyset$, then $T_A = T_S$. Otherwise, any node u in X must be guarded by certain node u' in $C \subseteq nd(T_S)$ according to stage S1 of **Algorithm 1**, hence u is only connected to u' in T_A. This means that u is a degree-one node in T_A. Therefore, we know $|in(T_A)| \leq |nd(T_S)|$ and thus:

$$|in(T_A)| \leq 2|nd(T_S^*)|. \tag{1}$$

Let T_I^- be the tree constructed from T_I by pruning all the degree-one nodes in T_I (we assume $|nd(T_I^-)| \neq 0$, otherwise, the proof becomes trivial). For any node u which is in C but not in T_I^- (i.e., $u \in C\backslash nd(T_I^-)$), we can find a group member $u_1 \in M$ such that u is adjacent to u_1. This is because that the guardian set C is a subset of the buddy set B, and any node in B has at least one neighboring node in M according to S1-a. If u_1 is not in T_I^-, then there must exist a node u_2 in T_I^- such that u_2 is adjacent to u_1, because any group member node in M is either an internal node in T_I or is adjacent to an internal node in T_I. In other words, any node in $C\backslash nd(T_I^-)$ can be connected to T_I^- by a path whose length is no more than 2. This means that we can find a tree in G spanning C whose number of nodes is no more than $|nd(T_I^-)|+2|C|$. Since T_S^* is an optimal Steiner tree spanning the nodes in C, we have:

$$|nd(T_S^*)| \leq |nd(T_I^-)|+2|C| = |in(T_I)|+2|C|. \tag{2}$$

For any node $v \in B$, we define $\sigma(v) = nb^+(v,G)\cap M$. Clearly, we have $|\sigma(v)| \leq \Delta+1$. Let $C^* \subseteq B$ be the guardian set in G that contains the minimum number of nodes. Since S1 of **Algorithm 1** identifies the guardian set C in a greedy manner, its approximation ratio is the same as that of the greedy set covering algorithm proposed in [26], hence we have:

$$\begin{aligned}|C| &\leq \left[\ln\left(\max_{v\in B}|\sigma(v)|\right)+1\right]\cdot|C^*| \\ &\leq [\ln(\Delta+1)+1]\cdot|C^*|.\end{aligned} \tag{3}$$

Note that $nd(T_I^-)\cap B$ is also a guardian set. so

$$|C^*| \leq |nd(T_I^-)| = |in(T_I)|. \tag{4}$$

Summarizing all the results obtained by far, we have:

$$\begin{aligned}|in(T_A)| &\leq 2(|in(T_I)|+2|C|) \\ &\leq 2|in(T_I)|+4\left[\ln(\Delta+1)+1\right]\cdot|in(T_I)| \\ &= [4\ln(\Delta+1)+6]\cdot|in(T_I)|,\end{aligned}$$

this bounds $|in(T_A)|$ from above by $|in(T_I)|$. ■

***Lemma** 2:* For any multicast tree T in G spanning the nodes in M, we have:

1) $\Psi(T) \geq k\cdot[\varepsilon_s+(|M|-1)\cdot\varepsilon_r]$
2) $\Psi(T) \geq |in(T)|\cdot(\varepsilon_s+\varepsilon_r)\cdot k$

Proof: Each node $u \in M$ must transmit its own $p(u)$ data packets. Therefore, the total energy consumption for transmitting data packets in a group communication session is at least $\sum_{u\in M}p(u)\cdot\varepsilon_s = k\cdot\varepsilon_s$. Moreover, the total energy consumption for receiving the data packets originated from u is $p(u)\cdot(|nd(T)|-1)\cdot\varepsilon_r$. Therefore,

$$\begin{aligned}\Psi(T) &\geq k\cdot\varepsilon_s+\sum_{u\in M}p(u)\cdot(|nd(T)|-1)\cdot\varepsilon_r \\ &\geq k\cdot[\varepsilon_s+(|M|-1)\cdot\varepsilon_r].\end{aligned}$$

The total energy consumption for realizing a group communication session using T can also be written as:

$$\begin{aligned}\Psi(T) &= \sum\nolimits_{u\in lv(T)}p(u)\cdot(|in(T)|+1)\cdot\varepsilon_s+ \\ &\quad \sum\nolimits_{u\in in(T)\cap M}p(u)\cdot|in(T)|\cdot\varepsilon_s+ \\ &\quad k\cdot(|nd(T)|-1)\cdot\varepsilon_r \\ &= k\cdot|in(T)|\cdot\varepsilon_s+\sum\nolimits_{u\in lv(T)}p(u)\cdot\varepsilon_s+ \\ &\quad k\cdot(|nd(T)|-1)\cdot\varepsilon_r.\end{aligned} \tag{5}$$

Therefore:

$$\begin{aligned}\Psi(T) &\geq k\cdot|in(T)|\cdot\varepsilon_s+k\cdot(|in(T)|+|lv(T)|-1)\cdot\varepsilon_r \\ &\geq |in(T)|\cdot(\varepsilon_s+\varepsilon_r)\cdot k.\end{aligned}$$

These give the two claimed lower bounds on $\Psi(T)$. ■

***Theorem** 1:* The multicast tree T_A constructed by **Algorithm 1** has an approximation ratio of $4ln(\Delta+1)+7$ for the MEAAM problem.

Proof: As $|lv(T_A)| \leq |M|$, using (5) we can get:

$$\begin{aligned}\Psi(T_A) &= k\cdot|in(T_A)|\cdot\varepsilon_s+\sum\nolimits_{u\in lv(T_A)}p(u)\cdot\varepsilon_s+ \\ &\quad k\cdot(|nd(T_A)|-1)\cdot\varepsilon_r \\ &\leq k\cdot|in(T_A)|\cdot\varepsilon_s+\sum\nolimits_{u\in M}p(u)\cdot\varepsilon_s+ \\ &\quad k\cdot(|in(T_A)|+|M|-1)\cdot\varepsilon_r \\ &\leq k\cdot|in(T_A)|\cdot(\varepsilon_s+\varepsilon_r)+ \\ &\quad k\cdot[\varepsilon_s+(|M|-1)\cdot\varepsilon_r].\end{aligned} \tag{6}$$

According to ***Lemma** 2*, we can get:

$$|in(T_{opt})|\cdot(\varepsilon_s+\varepsilon_r)\cdot k \leq \Psi(T_{opt}),$$

and

$$k\cdot[\varepsilon_s+(|M|-1)\cdot\varepsilon_r] \leq \Psi(T_{opt}).$$

On the other hand, according to the definition of T_I, we have:

$$|in(T_I)| \leq |in(T_{opt})|.$$

Let $\kappa = \Delta + 1$. Combining the above inequalities with ***Lemma** 1*, we have:

$$\begin{aligned}\Psi(T_A) &\leq (4ln\kappa + 6) \cdot |in(T_I)| \cdot (\varepsilon_s + \varepsilon_r) \cdot k + \Psi(T_{opt}) \\ &\leq (4ln\kappa + 6) \cdot |in(T_{opt})| \cdot (\varepsilon_s + \varepsilon_r) \cdot k + \Psi(T_{opt}) \\ &\leq (4ln\kappa + 6) \cdot \Psi(T_{opt}) + \Psi(T_{opt}) \\ &= [4\ln(\Delta + 1) + 7] \cdot \Psi(T_{opt}),\end{aligned}$$

hence the claimed approximation ratio. ■

2) Algorithm Complexity: The first stage of **Algorithm 1** has a time complexity of $\mathcal{O}(\Delta)$, as for every $v : nb(v, G) \cap M \neq \emptyset$, at least one node in $nb(v, G)$ leaves $\mathbf{Inactive}$ state every round.[2] In each round, all nodes in $nb2(v, G)$ sends a message, so the message complexity is $\mathcal{O}(\Delta \cdot |V|)$, where we use $|V|$ to bound from above the cardinality of $nb2(v, G)$. The second stage is designed for connecting the nodes in the guardian set. Its worst-case time complexity and message complexity are $\mathcal{O}(\mathrm{D}(G) \cdot |M|)$ and $\mathcal{O}(|M| \cdot |V|)$, respectively, as (i) only one group member leaves the $\mathbf{Connecting}$ state in each round in the worst case (hence $|M|$ rounds in total), (ii) differing from the first stage where the length of each round is a constant, this time the length of a round is determined by the round-trip time of a path (hence bounded by $\mathrm{D}(G)$, the diameter of G), and (iii) the number of messages in each round is bounded by $|V|$.

D. Tree Maintenance

We discuss the tree maintenance with respect to the two individual stages separately. Actually, maintaining the shared tree simply for allowing the group communication session to continue is a separate issue that has been tackled in the literature (e.g., [27]–[29]). Therefore, our focus is only on how to maintain the proven approximation ratio of the tree.

When a member v joins or leaves the group communication session, if it has no existing guarding node in $nb(v, G)$ (for joining) or is the only guarded node (for leaving), then the impact directly goes to the Steiner tree (which will be discussed later). Otherwise v could directly join or leave in a localized manner by notifying its guarding node. The complication comes when a joining or leaving violates the greedy cover principle (thus affecting the approximation ratio). Therefore, we may need to re-execute S1 of **Algorithm 1** under those circumstances to maintain the greedy cover, hence maintain the proven approximation ratio.

If a member joining or leaving does not lead to any changes in the existing guardian set C, then the approximate Steiner tree remains intact. If a member joining brings one more node to C, we may need to re-execute S2 of **Algorithm 1** in the worst case (when this newly joined guarding node results in several short cuts). Fortunately, we also have another choice of having this node directly connected to a closest (in terms of shortest path) guarding node. According to [30], this will lead to a $\mathcal{O}(\log |M|)$ approximation to Steiner tree, and in turn a $\mathcal{O}(\ln |M| \cdot \ln \Delta)$ approximation to MEAAM. If a member leaving removes a node from C, it may partition the tree into several fragments. The existing procedures of **Algorithm 1** can take care of this case as far as a new leader is elected for each fragment, as this is exactly the same as the fragment merging phase of the Steiner tree construction stage.

[2]Our algorithm works in asynchronous settings; the concept of *round* is used only for evaluating the time complexity.

V. MEGCOM-CFP: Constant Approximation for MEAAM with Fixed tx Power

The algorithm proposed in Sec. IV has a relatively high message complexity as we involve non-members in both stages. In this section, we propose a simplified algorithm that only involves group members in the first stage. Interestingly, we can show that this simplified algorithm has a constant approximation ratio.

A. The Algorithm

As this new algorithm shares the same second stage (S2) with **Algorithm 1**, we only show the updated first stage (S1) in **Algorithm 2** and then explain details accordingly.

Algorithm 2: Finding an approximate multicast tree T'_A for the MEAAM problem

Input: $G = (V, E)$ and M. For each node $v \in V$,
$v.state = \mathbf{Inactive};\ v.nb = nb(v, G);\ v.ns = 0$
Output: An approximate multicast tree T'_A

1 **task** SCAN() /* executed periodically */
2 **if** $v.state = \mathbf{Inactive} \wedge v \in M$ **then**
3 SENDMSG($'\mathrm{Compete}'$, $v.id, u.id$, NULL) to $\forall u \in v.nb$

4 **upon** RECVMSG($msgType, sid, tid, data$) at v with $v.id = tid$
5 $u \leftarrow$ the node with id sid
6 **switch** *msgType* **do**
7 **case** $'\mathrm{Compete}'$
8 **if** $v.state = \mathbf{Inactive}$ **then**
9 **if** $v.id > sid$ **then**
SENDMSG($'\mathrm{Accept}'$, $v.id, u.id$, NULL) to u
10 **else** SENDMSG($'\mathrm{ACK}'$, $v.id, u.id, v.state$) to u
11 **case** $'\mathrm{Ack}'$
12 **if** $data = \mathbf{Connecting}$ **then**
13 $v.state \leftarrow \mathbf{Treed};\ v.gd \leftarrow u;$ mark the edge to u
14 **else if** $data = \mathbf{Treed}$ **then** $v.nb \leftarrow v.nb \backslash \{u\}$
15 **case** $'\mathrm{Accept}'$
16 **if** $v.ns = |nb(v, G)|$ **then**
17 $v.state \leftarrow \mathbf{Connecting};\ v.gd \leftarrow v$
18 **else** $v.ns = v.ns + 1$

The objective is again to identify a guardian set C' for M, but the algorithm differs from **Algorithm 1** in that C' is constructed from M, i.e., $C' \subseteq M$. Similar to **Algorithm 1**, all the nodes that eventually change to $\mathbf{Connecting}$ state are in C'. The competition is again based on the smallest id principle. A node in $\mathbf{Inactive}$ state accepts any neighbor to be its guardian if its own id is larger (lines 8 to 9). If a node receives the acceptance from all its $\mathbf{Inactive}$ neighbors, it joins the guardian set C' (lines 16 to 17) by changing its state to $\mathbf{Connecting}$. A node that is not in $\mathbf{Inactive}$ state always acknowledges the competition message by announcing its current state (line 10). When a node v receives an acknowledgement message from node u that carries a $\mathbf{Connecting}$

state, it records u in $v.gd$ as its guarding node and changes its state to **Treed** (line 13); If the acknowledgement message carries a **Treed** sate, v knows that u has already been guarded by other nodes, hence deletes u from its active-neighbor table $v.nb$ (line 14). Note that this algorithm only uses one-hop communications and finds an arbitrary guardian set, and may degenerate to a Steiner tree algorithm when the group members are deployed very sparsely in the network. However, these do not hinder us from deriving a constant approximation ratio for our algorithm, which exponentially improves the best-known quadratic approximation ratio proposed in [3]. We will see this in the next section.

B. Performance Analysis

The approximation ratio of T'_A obtained by **Algorithm 2** is immediate from the following theorem.

***Theorem** 2:* $\Psi(T'_A) \leq 13\Psi(T_{opt})$

Proof: Let T_I again be the multicast tree spanning the nodes in M such that $|in(T_I)|$ is minimized. The outcome of **Algorithm 2** implies that the nodes in C' are mutually independent. Therefore, any node in $in(T_I)$ can be adjacent to at most 5 nodes in C' [31]. Moreover, as any node $u \in C'$ is a member of M, u is either in $in(T_I)$ or is adjacent to certain node in $in(T_I)$. So we can get:

$$|C'| \leq 5|in(T_I)| \tag{7}$$

Let the optimal Steiner tree and approximate Steiner tree (produced by S2 of **Algorithm 2**) spanning the nodes in C' be $\acute{T}^*_S$ and $\acute{T}_S$, respectively. We have:

$$|in(T'_A)| = |nd(\acute{T}_S)| \leq 2|nd(\acute{T}^*_S)| \tag{8}$$

Since any node in C' is either in $in(T_I)$ or is adjacent to certain node in $in(T_I)$, we can find a Steiner tree spanning the nodes in C' whose number of nodes is at most $|C'|+|in(T_I)|$. Therefore, we have:

$$|nd(\acute{T}^*_S)| \leq |C'| + |in(T_I)| \tag{9}$$

Combining (7)–(9), we can get:

$$|in(T'_A)| \leq 2(|C'| + |in(T_I)|) \leq 12|in(T_I)| \tag{10}$$

The result by far gives an upper bound for $|in(T'_A)|$. Now we can apply ***Lemma** 2* and (6) (similar to the proof of **Theorem 1**) to obtain $\Psi(T'_A) \leq 13\Psi(T_{opt})$. ■

We omit the complexity analysis for this algorithm, as it is very similar to **Algorithm 1**, except that the constant length of a round in the first stage becomes shorter, as only one-hop communications are required. The tree maintenance becomes simpler, as the guardian set is not constructed in a greedy manner, the impact of member joining or leaving directly goes to the Steiner tree.

VI. MEGCOM-CAP: Constant Approximation for MEAAM with Adjustable tx Power

When the tx power of each node is adjustable, the idea of guardian set does not work anymore, as whether a node can be guarded by another is not known before an actual tx power is chosen. Even if we have a shared multicast tree for the group communication session, any node u in a multicast tree T may need to use the tx power $\max\{d^{\alpha}_{(u,v)}|v \in nb(u,T)\}$ to transmit the data packets. Therefore, different nodes may use different tx power to forward the data packets. This makes the MEAAM problem more complicated.

To design an approximation algorithm for the MEAAM problem in the adjustable-tx-power case, we use a constructive proof to show the existence of an algorithm with a constant approximation ratio, which naturally suggests one possible algorithm to construct the multicast tree. We still try to build a multicast tree whose sum of tx power of the internal nodes is minimized, because the internal nodes still have a heavier forwarding load than the degree-one nodes for any multicase tree spanning M. Let $\lambda(u,T) = \max\{d^{\alpha}_{(u,v)}|v \in nb(u,T)\}$, and let T_W be a multicast tree such that $\Theta(T_W) = \sum_{u \in in(T_W)} \lambda(u,T_W)$ is minimized. Our idea is to find a multicast tree that approximates T_W and in turn approximates T_{opt}. The trick is to identify a quantitative relation between Θ and Ψ. This is done by ***Theorem** 3*, which in turn relies on the upper bounds of the total energy consumption for transmitting and receiving data, respectively, derived in ***Lemma** 3* and ***Lemma** 4*.

***Lemma** 3:* For any multicast tree T spanning the nodes in M, the total energy consumption of transmitting all data packets using T in a group communication session is at most $2k \cdot \Theta(T)$.

Proof: Each internal node in $in(T)$ must transmit every data packet sent by each member node. So the total energy consumption for transmitting data packets by the internal nodes in T is:

$$\sum\nolimits_{u \in in(T)} \lambda(u,T) \cdot \sum\nolimits_{v \in M} p(v) = k \cdot \Theta(T). \tag{11}$$

Any degree-one node in T has a unique neighboring node in T. Therefore, there exists a function $f : lv(T) \to in(T)$ such that $f(u)$ is the unique internal node adjacent to $u : \forall u \in lv(T)$. Let $Z = \{f(u)|u \in lv(T)\}$. Clearly, for any degree one node $u \in T$, we have $\lambda(u,T) \leq \lambda(f(u),T)$. The total energy consumption for transmitting data packets by the degree-one nodes in T is:

$$\begin{aligned} & \sum\nolimits_{u \in lv(T)} \lambda(u,T) \cdot p(u) \\ = \; & \sum\nolimits_{v \in Z} \sum\nolimits_{u \in \{w|w \in lv(T), f(w)=v\}} \lambda(u,T) \cdot p(u) \\ \leq \; & \sum\nolimits_{v \in Z} \sum\nolimits_{u \in lv(T)} \lambda(v,T) \cdot p(u) \\ \leq \; & k \cdot \sum\nolimits_{v \in Z} \lambda(v,T) \\ \leq \; & k \cdot \Theta(T). \end{aligned} \tag{12}$$

Combining (11) and (12), the lemma follows. ■

***Lemma** 4:* For any multicast tree T spanning the nodes in M, the total energy consumption of receiving all data packets using T in a group communication session is at most $\Psi(T_{opt}) + k \cdot \Theta(T)$.

Proof: Based on our assumption that $\forall v \in T$, $\varepsilon_r \leq$

$\lambda(v,T)$, we can get:

$$\begin{aligned} |in(T)| \cdot k \cdot \varepsilon_r &= k \cdot \sum\nolimits_{v \in in(T)} \varepsilon_r \\ &\leq k \cdot \sum\nolimits_{v \in in(T)} \lambda(v,T) \\ &= k \cdot \Theta(T). \end{aligned}$$

The total energy consumption for receiving all the data packets by the nodes in T is $k \cdot (|nd(T)| - 1) \cdot \varepsilon_r$, which is no more than the total energy consumption $\Psi(T)$ for realizing a group communication session, because $\Psi(T)$ is the sum of the energy consumption for receiving and transmitting all data packets. Note that T can be any tree spanning the nodes in M, so we also have:

$$k \cdot [|nd(T_{opt})| - 1] \cdot \varepsilon_r \quad \leq \quad \Psi(T_{opt}).$$

Combining the above two inequalities with $|lv(T)| \leq |M| \leq |nd(T_{opt})|$, we can get:

$$\begin{aligned} & k \cdot (|nd(T)| - 1) \cdot \varepsilon_r \\ = \; & k \cdot [|lv(T)| + |in(T)| - 1] \cdot \varepsilon_r \\ \leq \; & k \cdot [|nd(T_{opt})| - 1] \cdot \varepsilon_r + k \cdot |in(T)| \cdot \varepsilon_r \\ \leq \; & \Psi(T_{opt}) + k \cdot \Theta(T), \end{aligned} \tag{13}$$

hence the claimed upper bound follows. ■

***Theorem** 3:* Let T_W be a multicast tree spanning the nodes in M such that $\Theta(T_W)$ is minimized. For any multicast tree T, if $\Theta(T) \leq \rho \cdot \Theta(T_W)$, then $\Psi(T) \leq (3\rho + 1) \cdot \Psi(T_{opt})$

Proof: Since the total energy consumption for transmitting data packets in a group communication session by the internal nodes in T_{opt} is $k \cdot \Theta(T_{opt})$ (according to (11)), we have:

$$k \cdot \Theta(T_{opt}) \leq \Psi(T_{opt}).$$

According to the definition of T_W, we get:

$$\Theta(T_W) \leq \Theta(T_{opt}).$$

Then, combining ***Lemma** 3* and ***Lemma** 4* with the above two inequalities, we have:

$$\begin{aligned} \Psi(T) &\leq 3k \cdot \Theta(T) + \Psi(T_{opt}) \\ &\leq 3\rho k \cdot \Theta(T_W) + \Psi(T_{opt}) \\ &\leq 3\rho k \cdot \Theta(T_{opt}) + \Psi(T_{opt}) \\ &\leq (3\rho + 1)\Psi(T_{opt}), \end{aligned} \tag{14}$$

hence the theorem follows. ■

Remark: This theorem is itself very important. It shows that, if we can find a tree T that approximates T_W within a constant ratio, then T is also a constant approximation to T_{opt}.

In this paper, we only suggest a straightforward algorithm that achieves a constant approximation ratio. In fact, we will show that, if we apply the distributed Steiner tree algorithm given in S2 of **Algorithm 1** to span M directly, the resulting tree $\ddot{T}_S$ approximates T_W within a constant ratio.

***Lemma** 5:* Let $\ddot{T}_S$ be a 2-approximation Steiner tree in G spanning M, we have: $\Theta(\ddot{T}_S) \leq 48\Theta(T_W)$.

Proof: For any multicast tree T spanning the nodes in M, we denote by $\zeta(T)$ the sum of the weights (tx power) of the edges in T. For any node u in $in(T_W)$, we denote by $T_W^{(u)}$ the Euclidean minimum spanning tree of the nodes in $\{u\} \cup nb(u, T_W)$. According to [12], we have:

$$\zeta\left(T_W^{(u)}\right) \leq c \cdot \lambda(u, T_W), \quad 6 \leq c \leq 12. \tag{15}$$

Let G' be the graph constructed by superposing the $T_W^{(u)}$'s for all $u \in in(T_W)$. Let T_z be an arbitrary spanning tree of G'. We can get:

$$\begin{aligned} \zeta(T_z) &\leq \sum\nolimits_{u \in in(T_W)} \zeta\left(T_W^{(u)}\right) \\ &\leq c \cdot \sum\nolimits_{u \in in(T_W)} \lambda(u, T_W) \\ &= c \cdot \Theta(T_W). \end{aligned} \tag{16}$$

Let $\ddot{T}_S^*$ be the minimum Steiner tree in G spanning the nodes in M. Since $\ddot{T}_S$ is a 2-approximation Steiner tree and T_z is also a tree spanning M, we have:

$$\zeta(\ddot{T}_S) \leq 2\zeta(\ddot{T}_S^*) \leq 2\zeta(T_z). \tag{17}$$

Note that the weight of any edge in $\ddot{T}_S$ can be counted at most twice in $\Theta(\ddot{T}_S)$, so

$$\Theta(\ddot{T}_S) \leq 2\zeta(\ddot{T}_S). \tag{18}$$

The claimed approximation ratio follows by combining all these inequalities. ■

Finally, using Theorem 3 and Lemma 5 we can easily get:

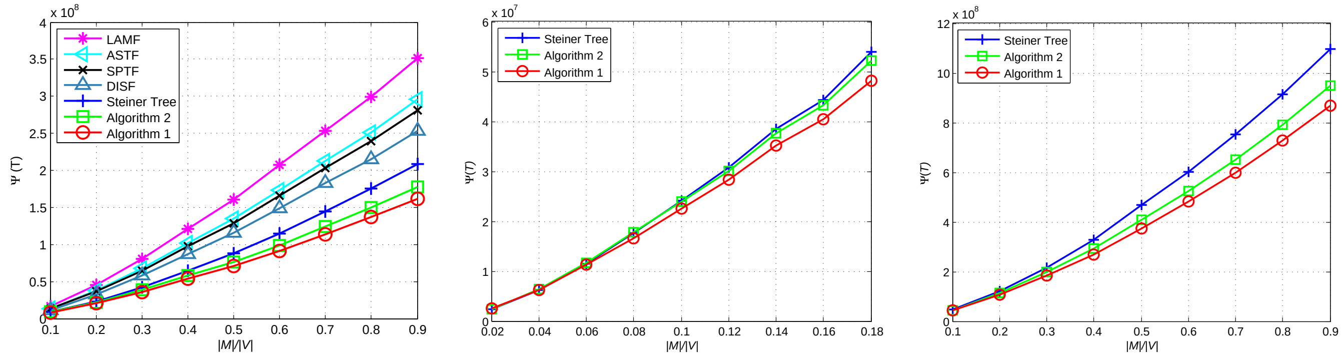
***Theorem** 4:* Our distributed Steiner tree algorithm on M approximates the MEAAM problem under the adjustable tx power case with a constant approximation ratio of 145.

VII. Simulations

In this section, we present the simulation results of our algorithms. We implement a simulator using C++ and use it to study how the performance of different MEGCOM algorithms is affected by various network parameters such as the network size, the node density and the percentage of group members. In the simulations, the network nodes are randomly deployed in a square area according to a certain node density, where the node density is defined as 1 if $|V|$ nodes are deployed in a $\sqrt{|V|} \times \sqrt{|V|}$ square. The group members are selected from V following a Bernoulli distribution, and the number of packets originated from each group member is chosen from a uniform distribution $\mathcal{U}(1, 100)$. For each setting of the network parameters, we generate 100 network instances and show the mean values of the simulation results.

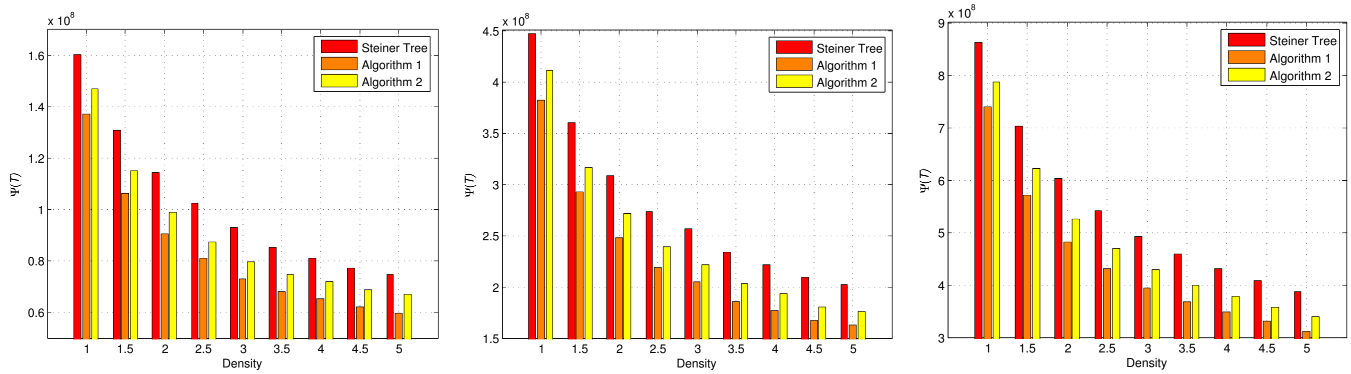
A. Fixed TX Power

We first study the performance of MEGCOM-LFP and MEGCOM-CFP under the fixed transmission power case with $\varepsilon_s = 200$, $\varepsilon_r = 20$, and the transmission range of each node is set to 2. For comparison, we build different multicast trees for the MEAAM problem in the simulations, using MEGCOM-LFP (**Algorithm 1**), MEGCOM-CFP (**Algorithm 2**), the Shortest Path Tree (SPTF) algorithm, the Steiner Tree algorithm, as well as DISF [3], ASTF [27], and LAMF [28]. In the SPTF algorithm, a root node is randomly selected from the



(a) 300-node networks (b) 500-node networks (c) 700-node networks

Fig. 5. Comparing different algorithms for building multicast trees with the network density fixed to be 2.



(a) 300-node networks (b) 500-node networks (c) 700-node networks

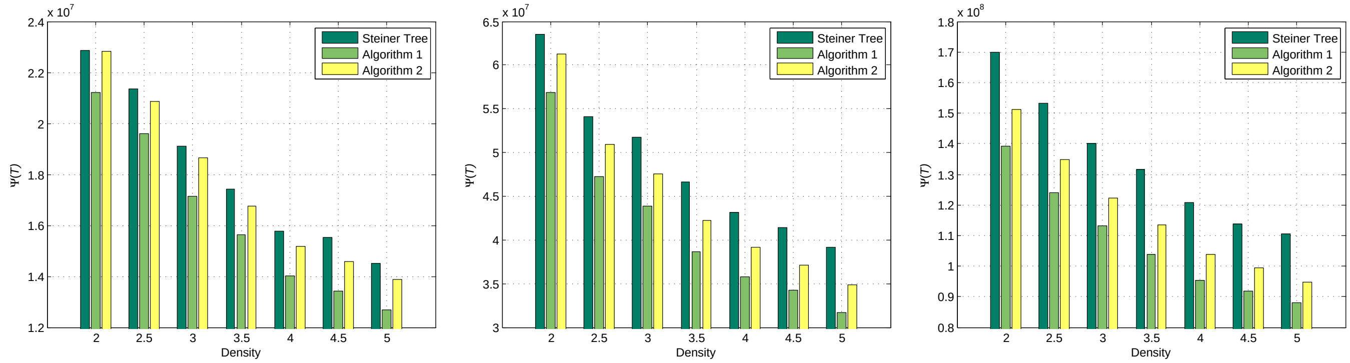
Fig. 6. Comparing different algorithms for building multicast trees with the percentage of group members fixed to be 60%.

group members, and the shared multicast tree is constructed by aggregating the shortest paths from the root node to other group members. In the implementation of the Steiner Tree algorithm, we adopt the approximation algorithm proposed by [20]; this is the Steiner tree algorithm also used in [3] as the solution to the MEAAM problem, and used in [12] as the solution to the min-energy one-to-many multicasting problem.

In Figure 5, the network density is fixed to 1 and the network size is set to 300, 500, and 700 in Figure 5(a)-(c), respectively. The percentage of group members scales from 10% to 90% in Figure 5(a)(c), and scales from 2% to 18% in Figure 5(b).

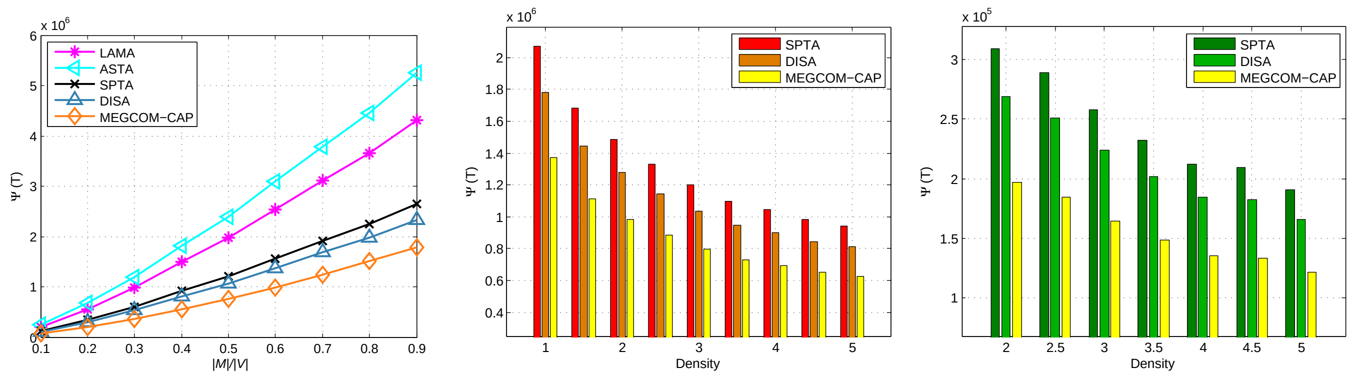
It can be seen from Figure 5(a) that the SPTF, ASTF, LAMF algorithms always consume more energy than the other algorithms. This can be explained by the fact that trees generated by these algorithms are generally based on constructing shortest paths from one group member or certain "rendezvous node" to the other nodes. As a result, these trees have a large number of internal tree nodes that are not group members. Meanwhile, we can see that the Steiner tree algorithm outperforms DISF (hence also outperforms ASTF, SPTF, LAMF), which is consistent with the fact that the linear approximation ratio of the Steiner tree algorithm for MEAAM is better than the quadratic approximation ratio of DISF. Moreover, we can also see from Figure 5(a)-(c) that **Algorithm 1** and **Algorithm 2** both outperform the Steiner tree algorithm. This phenomenon validates our analysis on approximation ratios in Section IV and Section V. The main reason for this phenomenon is that **Algorithm 1** and **Algorithm 2** both reduce the internal nodes of the shared multicast tree, whose energy consumption is the predominant part of the total energy consumption of a group communication session. Another phenomenon we observe from Figure 5 is that our algorithms perform better when the percentage of group members increases. Actually, when 90% network nodes become group members, **Algorithm 1** saves up to 40% energy cost compared with the SPTF algorithm in Figure 5(a), or 25% compared with the Steiner tree algorithm in Figure 5(c). This can be understood by the fact that more energy are conserved by our algorithms when more data packets are involved in the group communication session. Finally, Figure 5(b) reveals that when the number of group members is very small (down to about 10 nodes), different multicast algorithms perform closely on the total energy cost. However, in such a case, the energy-efficiency problem becomes less of a concern because the total energy cost achieved by any algorithm is already very low.

In Figure 6, we study the impact of node density on the performance of different algorithms. This time we remove the results for the SPTF, ASTF, DISF, LAMF algorithms, as they are proved to perform worse than the other algorithms in the previous comparisons. The percentage of group members is fixed to 60%, and the node density scales from 1 to 5 with an increment of 0.5. Again, the network size is set to 300, 500, and 700 in Figure 6(a)-(c), respectively. We observe that **Algorithm 1** and **Algorithm 2** always perform better than the Steiner tree algorithm in Figure 6. In particular, **Algorithm 1** saves 15%∼20% energy compared with the



(a) $|M|/|V| = 10\%$ (b) $|M|/|V| = 20\%$ (c) $|M|/|V| = 40\%$

Fig. 7. Comparing different algorithms for building multicast trees with the network size fixed to be 500-node.



(a) 300-node networks with density 1 (b) 300-node networks with $|M|/|V| = 60\%$ (c) 300-node networks $|M|/|V| = 10\%$

Fig. 8. Comparing different algorithms under the adjustable power case.

Steiner tree algorithm. This demonstrates that the superiority of our MEGCOM algorithms persists when the node density changes. Here we use density as a rough indicator of Δ, as generating degree constrained networks is non-trivial. And even if we generated such graphs, they could hardly reflect any practical WSN deployments.

In Figure 7, we study the joint impact of node density and group-member-percentage on the energy consumption for group communication. We fix the number of network nodes to 500, and the percentage of group members is set to 10%, 20% and 40% in Figure 7(a)-(c), respectively. The node density scales from 2 to 5 with an increment of 0.5. We can see that our algorithms outperforms the Steiner tree algorithm more significantly when the percentage of group members increases. Even when the percentage of group members is small (such as in Figure 7(a)), our algorithms still show significant superiority in densely deployed networks, which is actually a prevailing deployment pattern in many applications [32], [33].

An interesting fact revealed by the simulations is that **Algorithm 1** almost always performs better than **Algorithm 2**, although the latter has a better theoretical approximation ratio than the former (see ***Theorem*** *1* and ***Theorem*** *2*). One generally considers a constant approximation ratio to be better than a logarithmic ratio. However, as an approximation ratio only serves as a metric to measure the worst case performance of an algorithm, the average performance of the algorithm in practice may not be well characterized by its approximation ratio. We can attribute this effect to the fact that the guardian set is searched from a larger solution space in **Algorithm 1** than in **Algorithm 2**, which should on average result in less internal nodes in the shared multicast tree constructed by **Algorithm 1**, hence a lower total energy consumption of a group communication session using the tree and a better performance of **Algorithm 1** in average sense.

B. Adjustable Tx Power

In this section, we compare the performance of MEGCOM-CAP with the other algorithms under the adjustable transmission power case, where the transmission power for any node u to communicate with another node v is set to $d^2_{(u,v)}$. The DISF, LAMF, SPTF, ASTF algorithms are adapted to the adjustable tx power case, and the adapted algorithms are named DISA, LAMA, SPTA and ASTA, respectively. More specifically, the SPTF algorithm is adapted by assigning each edge (u, v) a weight $d^2_{(u,v)}$, and the LAMF (ASTF) algorithms are adapted by setting the transmission power of any node u in a group communication tree T to $\max\{d^2_{(u,v)} | v \in nb(u, T)\}$, which is the minimum energy required for u to communicate with its neighboring nodes in T.

As in Section VII-A, we also study the impact of various network conditions (such as the percentage of group members and the node densities) on the performance of different algorithms, and the network settings in Fig. 8(a)-(c) are the same with those in Fig. 5(a), Fig. 6(a) and Fig. 7(a), respectively. Again, we can see from Fig. 8 that

SPTA, DISA and MEGCOM-CAP all outperform ASTF and LAMF, while MEGCOM-CAP performs the best among them. This demonstrates the superiority of MEGCOM-CAP under the adjustable tx power case.

VIII. Discussion

An important issue we would like to discuss here is multi-path fading, which is a major contributor to the unreliability of wireless communication links. In fact, considering multi-path fading in MEAAM essentially leads to another research problem, i.e., reliable group communication for wireless ad hoc networks. Compared with the large body of work on energy-efficient data transmission in wireless networks where the wireless links are assumed to be dependable (e.g., [3], [4], [6], [11]–[14], [21]), the reliable group communication problem is relatively less studied. Nevertheless, our work can be potentially extended to the unreliable link case by assigning each link a weight which is calculated by the multiplication of the one-time transmission energy and the expected retransmission times on that link. Actually, such a trick is also used in other proposals on one-to-many reliable multicasting such as [34], and the multicast tree built in [34] is exactly an approximate Steiner tree. We plan to investigate the MEAAM problem under the unreliable link case more thoroughly in the future work.

IX. Conclusion

In this paper, we have studied the Minimum-Energy All-to-All Multicasting (MEAAM) problem in multi-hop wireless networks, where the transmission power of each wireless node could be either fixed or adjustable. Since the MEAAM problem is NP-complete, we have provided a set of distributed approximation algorithms under our MEGCOM (Minimum-Energy Group COMmunication) framework. For each algorithm, we have proven its approximation ratio with respect to the optimal solution. Our approximation ratios are significantly better than those of the best-known approximation algorithms for the MEAAM problem. We have further performed extensive simulations to validate our theoretical analysis and also to confirm the energy efficiency of our MEGCOM algorithms.

References

[1] D. Powell (Guest Ed.), "Group communication," *Comm. of the ACM*, vol. 39, pp. 50–97, 1996.

[2] G. Chockler, I. Keidar, and R. Vitenberg, "Group communication specifications: A comprehensive study," *ACM Computing Surveys*, vol. 33, no. 4, pp. 427–469, 2001.

[3] W. Liang, R. Brent, Y. Xu, and Q. Wang, "Minimum-energy all-to-all multicasting in wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5490–5499, 2009.

[4] S. Huang, H. Wu, and S. Iyengar, "Multi-source broadcast in wireless networks," *IEEE Trans. Parallel and Distributed Sys.*, vol. 23, no. 10, pp. 1908–1914, 2012.

[5] L. Al-Kanj, Z. Dawy, and E. Yaacoub, "Energy-aware cooperative content distribution over wireless networks: Design alternatives and implementation aspects," *IEEE Communications Surveys & Tutorials*, 2013, In Press.

[6] J. Wang, W. Luo, Q. Feng, and J. Guo, "Parameterized complexity of min-power multicast problems in wireless ad hoc networks," *Theoretical Computer Science*, 2013, In Press.

[7] J. Luo, P. Eugster, and J.-P. Hubaux, "Route driven gossip: Probabilistic reliable multicast in ad hoc networks," in *Proc. of the 22nd IEEE INFOCOM*, 2003, pp. 2229–2239.

[8] ——, "PILOT: ProbabilistIc Lightweight grOup communication sysTem for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 2, pp. 164–179, 2004.

[9] ——, "Probabilistic Reliable Multicast in Ad Hoc Networks," *Elsevier Ad Hoc Networks*, vol. 2, no. 4, pp. 369–386, 2004.

[10] X. Yu and S. Chandra, "Designing an asynchronous group communication middleware for wireless users," in *Proc. of the 12th ACM MSWiM*, 2009, pp. 274–279.

[11] J. Wieselthier, G. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM*, 2000, pp. 585–594.

[12] P.-J. Wan, G. Calinescu, and C.-W. Yi, "Minimum-power multicast routing in static ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 507–514, 2004.

[13] W. Liang, "Approximate minimum-energy multicasting in wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 4, pp. 377–387, 2006.

[14] D. Li, Q. Liu, X. Hu, and X. Jia, "Energy efficient multicast routing in ad hoc wireless networks," *Computer Communications*, vol. 30, no. 18, pp. 3746–3756, 2007.

[15] S. J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Kluwer Mobile Networks & Applications*, vol. 7, no. 6, pp. 441–453, 2002.

[16] B. Tavli and W. Heinzelman, "Energy-efficient real-time multicast routing in mobile ad hoc networks," *IEEE Trans. on Computers*, vol. 60, no. 5, pp. 707–722, 2011.

[17] K. Han, J. Luo, Y. Liu, and A. Vasilakos, "Algorithm design for data communications in duty-cycled wireless sensor networks: a survey," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 107–113, 2013.

[18] K. Han, Y. Liu, and J. Luo, "Duty-cycle-aware minimum-energy multicasting in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 910–923, 2013.

[19] K. Han, L. Xiang, J. Luo, M. Xiao, and L. Huang, "Energy-Efficient Reliable Data Dissemination in Duty-Cycled Wireless Sensor Networks," in *Proc. ACM MobiHoc*, 2013, pp. 287–292.

[20] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica*, vol. 15, pp. 141–145, 1981.

[21] Y. Wu, Z. Mao, S. Fahmy, and N. Shroff, "Constructing maximum-lifetime data-gathering forests in sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1571–1584, 2010.

[22] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OSLR," RFC 3626, October 2003, experimental.

[23] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing Protocol," RFC 3561, July 2003, experimental.

[24] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR)," RFC 4728, February 2007, experimental.

[25] F. Bauer and A. Varma, "Distributed algorithms for multicast path setup in data networks," *IEEE Trans. on Networking*, vol. 4, no. 2, pp. 181–191, 1996.

[26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. MIT Press, 2001.

[27] C.-C. Chiang, M. Gerla, and L. Zhang, "Adaptive shared tree multicast in mobile wireless networks," in *Proc. IEEE GLOBECOM*, vol. 3, 1998, pp. 1817–1822.

[28] L. Ji and M. Corson, "A lightweight adaptive multicast algorithm," in *Proc. IEEE GLOBECOM*, vol. 2, 1998, pp. 1036 –1042.

[29] X. Jia, "A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks," *IEEE/ACM Trans. on Networking*, vol. 6, no. 6, pp. 828–837, 1998.

[30] M. Imase and B. M. Waxman, "Dynamic steiner tree problem," *SIAM J. on Discrete Math.*, vol. 4, no. 3, pp. 369–384, 1991.

[31] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proc. INFOCOM*, 2002, pp. 1597–1604.

[32] A. Subramanian, H. Lundgren, T. Salonidis, and D. Towsley, "Topology control protocol using sectorized antennas in dense 802.11 wireless networks," in *Proc. IEEE ICNP*, 2009, pp. 1–10.

[33] A. P. Subramanian, H. Lundgren, and T. Salonidis, "Experimental characterization of sectorized antennas in dense 802.11 wireless mesh networks," in *Proc. ACM MobiHoc*, 2009, pp. 259–268.

[34] X. Y. Li, Y. Wang, H. Chen, X. Chu, Y. Wu, and Y. Qi, "Reliable and energy-efficient routing for static wireless ad hoc networks with unreliable links," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1408–1421, 2009.

PLACE PHOTO HERE

Kai Han received the B.S. and Ph.D. degrees in computer science from the University of Science and Technology of China, Hefei, China, in 1997 and 2004, respectively. From 2005 to 2008, he was a post-doctoral research fellow at the School of Mathematics, University of Science and Technology of China.

He is currently an associate professor at the School of Computer Science, Zhongyuan University of Technology, China. He is now a visiting fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include wireless ad hoc and sensor networks, combinatorial and stochastic optimization, as well as algorithmic game theory. He is a Member of both IEEE and ACM.

PLACE PHOTO HERE

Liu Xiang received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore. She is currently working as a quantitative research associate at Dynamic Technology Lab, Singapore. Her research interests include optimization theory and wireless sensor networks.

PLACE PHOTO HERE

Jun Luo received his BS and MS degrees in Electrical Engineering from Tsinghua University, China, and the PhD degree in Computer Science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland. From 2006 to 2008, he has worked as a post-doctoral research fellow in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the School of Computer Engineering, Nanyang Technological University in Singapore, where he is currently an assistant professor. His research interests include wireless networking, mobile and pervasive computing, distributed systems, multimedia protocols, network modeling and performance analysis, applied operations research, as well as network security. More information can be found at http://www3.ntu.edu.sg/home/junluo.

PLACE PHOTO HERE

Yang Liu received the M.Sc. and Ph.D. degrees from Northwestern Polytechnical University, Xian, China, in 2003 and 2007, respectively, both in computer science. She is currently an associate professor with the School of Information Sciences and Engineering, Henan University of Technology, Zhengzhou, China. Her research interests include distributed algorithms, algorithmic game theory, grid computing and cloud computing.