# Energy-Optimal Distributed Algorithms for Minimum Spanning Trees

Yongwook Choi, Maleq Khan, *Member, IEEE,* V.S. Anil Kumar, and Gopal Pandurangan, *Member, IEEE*

***Abstract*—Traditionally, the performance of distributed algorithms has been measured in terms of time and message complexity. Message complexity concerns the number of messages transmitted over all the edges during the course of the algorithm. However, in energy-constrained ad hoc wireless networks (e.g., sensor networks), energy is a critical factor in measuring the efficiency of a distributed algorithm. Transmitting a message between two nodes has an associated cost (energy) and moreover this cost can depend on the two nodes (e.g., the distance between them among other things). Thus in addition to the time and message complexity, it is important to consider *energy complexity* that accounts for the total energy associated with the messages exchanged among the nodes in a distributed algorithm.**

**This paper addresses the minimum spanning tree (MST) problem, a fundamental problem in distributed computing and communication networks. We study energy-efficient distributed algorithms for the Euclidean MST problem assuming random distribution of nodes. We show a non-trivial lower bound of $\Omega(\log n)$ on the energy complexity of any distributed MST algorithm. We then give an energy-optimal distributed algorithm that constructs an optimal MST with energy complexity $O(\log n)$ on average and $O(\log n \log\log n)$ with high probability. This is an improvement over the previous best known bound on the average energy complexity of $\Omega(\log^2 n)$. Our energy-optimal algorithm exploits a novel property of the giant component of sparse random geometric graphs. All of the above results assume that nodes do not know their geometric coordinates. If the nodes know their own coordinates, then we give an algorithm with $O(1)$ energy complexity (which is the best possible) that gives an $O(1)$ approximation to the MST.**

***Index Terms*—Distributed Algorithm, Minimum Spanning Tree, Energy-Efficient, Random Geometric Graph, Percolation**

## I. Introduction

Emerging technologies such as ad hoc wireless networks and sensor networks operate under inherent resource constraints. Consider a sensor network, an ad hoc wireless network in a geographic area built up of a large number of inexpensive devices called as sensors. Sensors have energy constraints due to the limited battery power of the sensor nodes; this severely constrains the amount of computation the nodes can do and the distance to which they can communicate. A distributed algorithm which consumes a relatively large amount of power may not be suitable in such a resource-constrained network. The topology of these networks can change frequently due to mobility or node failures. Communication cost and running time are even more crucial in such a dynamic setting. A distributed algorithm that runs on such devices should have as little communication as possible and should run as fast as possible (i.e., requiring a small number of communication rounds) and use as little energy as possible. Hence it becomes critical to design energy-efficient distributed algorithms that operate on these networks.

Traditionally, the performance of distributed algorithms has been measured in terms of running time and message complexity. In fact, in standard distributed computing literature, these are the two most widely used measures [23]. Message complexity concerns the total number of messages transmitted over all the edges during the course of the algorithm. However, in many settings, we require a more accurate and relevant measure of performance. A good example is ad hoc wireless networks, where energy is a very critical factor for measuring the efficiency of a distributed algorithm. Transmitting a message between two nodes has an associated cost (energy), and moreover this cost can depend on the two nodes (e.g., the distance between them among other things). Thus in addition to the traditional time and message complexity, it is also relevant to consider *energy complexity* that accounts for the total energy associated with the messages exchanged among the nodes in a distributed algorithm.

This paper addresses the minimum spanning tree (MST) problem, which is an important primitive in many applications in wireless networks, e.g., broadcasting, data aggregation and topology control [24, 27, 5]. Data aggregation paradigms commonly use trees to schedule the transmission of data from all nodes in the graph at a source [18]; minimum cost spanning trees help optimize the energy usage in this process. Various topology control algorithms also use MSTs to construct well connected subgraphs with provable cost relative to the optimum [24]. Motivated by ad hoc wireless and sensor networks, we focus on the Euclidean MST problem. Our goal in this paper is to study distributed algorithms for the Euclidean MST problem that have low energy complexity. We show energy complexity lower bounds on the performance of any distributed MST. We also study distributed *approximation*

Manuscript received August 30, 2008; revised January 31, 2009. M. Khan and V.S. Anil Kumar were supported in part by NSF Nets Grant CNS-0626964, NSF HSD Grant SES-0729441, CDC Center of Excellence in Public Health Informatics Grant 2506055-01, NIH-NIGMS MIDAS project 5 U01 GM070694-05, DTRA CNIMS Grant HDTRA1-07-C-0113 and NSF NETS CNS-0831633. G. Pandurangan was supported in part by NSF Award CCF-0830476. A short version of this paper appeared in the proceedings of the 20th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), Munich, Germany, 2008, 188-190.

Y. Choi and G. Pandurangan are with the Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA. e-mail: {ywchoi, gopal}@cs.purdue.edu

M. Khan and V.S.Anil Kumar are with the Network Dynamics and Simulation Science Laboratory, Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061, USA. e-mail: {maleq,akumar}@vbi.vt.edu.

0000–0000/00$00.00 © 2009 IEEE

algorithms for MST that give better energy complexity at the price of some additional information about the coordinates of the nodes and obtain a constant factor approximation to the MST. Because of space limitations, we have omitted several proofs, which are available in the full version [4].

## II. Model and Problem

**Network Model.** We assume that the network is modeled as a weighted undirected graph $G = (V, E, w)$ where $V$ is the set of the nodes (vertices), $E$ is the set of the (bi-directional) communication links between them, and $w(e)$ is the weight of the edge $e \in E$. The weight $w(u, v)$ represents the energy associated with transmitting a message between $u$ and $v$. We assume that the transmission at power level $P(u)$ is received all nodes $v$ with $w(u, v) \leq P(u)$, and the nodes have capability to set the power level adaptively within some range - recent advances in technology, such as cognitive radios have made this fairly easy to achieve with low overhead [1].

The graph $G$ has the following underlying geometry. The nodes are set of $|V| = n$ points distributed *randomly* in a unit square and two nodes are connected if they are within distance $r$ of each other, where $r$ is a transmission range induced by the maximum power level allowed. The graph thus induced is a *random geometric graph* [22]. This is a standard graph model that has been widely used in the literature for modeling ad hoc (sensor) networks. Without loss of generality, we assume that $r$ is chosen such that $G$ is connected (cf. Section V). We assume the Radio Broadcast Model (RBN) for interference [23], in which the transmission from a node $u$ to its neighbor $v$ is successful, provided no other neighbor $w \neq u, w \in N(v)$ transmits at the same time. Most of our results will extend to other interference models, e.g. Tx-Rx [2].

In an ad hoc wireless sensor network, the total energy required in a distributed algorithm typically depends on the time needed, the number of messages exchanged, and the radiation energy needed to transmit the messages over a certain distance [3, 28]. The radiation energy $w(u, v)$ needed to transmit a message from node $u$ to node $v$ is typically proportional to some power of the distance $d(u, v)$ [16, 19]. We assume that $w(u, v) = a \cdot (d(u, v))^\alpha$ for some constants $a$ and $\alpha$ (which is referred to as the path loss exponent) - for concreteness, we use $\alpha = 2$ in most of the paper; many of our results can be generalized to other functions that model other path loss and fading models. Formally, the *energy complexity* of the distributed algorithm is defined as $\sum_{i=1}^{m} w_i$, where $w_i$ is the weight of the edge that connects the nodes exchanging the $i$th message, and $m$ is the total number of messages exchanged by the algorithm.

**Distributed Computing Model.** Each node in $G$ hosts a processor with limited initial knowledge. Each node has unique identity numbers. The nodes do not know the weights of its incident edges (or equivalently the distance to its neighbors). We assume that the communication is synchronous and occurs in discrete time steps. The energy associated with a bi-directional communication between neighbors $u$ and $v$ is $\Theta(w(u, v))$, i.e., if $u$ wants to send a message to $v$ and $v$ replies back to $u$ then the cost associated with this bi-directional communication is $2w(u, v)$. A message is of size $O(\log n)$ bits and a node can send one message per time step. In a one-directional communication, when a node $u$ sends a message to a distance $d \leq r$, we assume that any node within distance $d$ can receive the message. The cost associated with this message is (proportional to) $d^2$. This is called *local broadcasting* and is a feature of radio and wireless networks. For simplicity, we assume that there are no collisions, i.e., each message is transmitted successfully in one attempt. We leave the consideration of a realistic physical interference model as a future work as discussed in Section VIII.

**MST Problem.** Formally, the Euclidean MST problem is, given a network $G = (V, E, w)$, to find a tree $T$ spanning $V$ such that $\sum_{(u,v)\in T} d(u, v)$ is minimized. In fact, We consider a generalized version of the above problem: Find a tree $T$ spanning $V$ such that the cost = $\sum_{(u,v)\in T} d^\alpha(u, v)$ is minimized where $\alpha$ is a (small) positive number. The motivation for this objective function comes from energy requirements in a wireless communication paradigm as mentioned earlier. It can easily be shown (e.g., using Kruskal's algorithmic construction) that the MST which minimizes $\sum_{(u,v)\in T} d(u, v)$ also minimizes $\sum_{(u,v)\in T} d^\alpha(u, v)$ for any $\alpha > 0$. In the rest of the paper, we use the terms *cost* and *quality* interchangeably. Although our results can be generalized to any $\alpha$, we focus on $\alpha = 1$ (the Euclidean MST) and $\alpha = 2$. (Note that for the energy complexity, we assume $\alpha = 2$ always.)

Computing an MST by a distributed algorithm is a fundamental task, as the following distributed computation can be carried over the best backbone of the communication graph. Two important applications of MST are broadcasting and data aggregation. In wireless networks, an MST can be used as a communication tree to minimize energy consumption since it minimizes $\sum_{(u,v)\in T} d^\alpha(u, v)$. In data aggregation, the idea is to combine the data coming from different sources enroute to eliminate redundancy, minimize the number of transmissions, and thus save energy. Common aggregate functions are minimum, maximum, average, etc [18]. One popular paradigm for computing such aggregates is to construct a (directed) tree rooted at the sink where each node forwards its (locally) *aggregated* data collected from its subtree to its parent [13]. For such cases, MST is the optimal data aggregation tree [15]. It was shown in [5, 27] that broadcasting based on MST consumes energy within a constant factor of the optimum.

## III. Our Results

We show tight upper and lower bounds on the energy complexity of distributed MST algorithms. We first show that $\Omega(\log n)$ is a lower bound on the energy complexity of any distributed MST algorithm. In fact, we show that this is the lower bound for constructing any spanning tree in the network. We then give a distributed algorithm that constructs an optimal MST with $O(\log n)$ energy complexity on average and $O(\log n \cdot \log \log n)$ energy complexity with high probability (whp). (Throughout the paper, "whp" means with probability tending to 1 as $n \to \infty$, where $n$ is the number of nodes in the network.) The previous best known bound on the average energy complexity for distributed MST in this model was $\Omega(\log^2 n)$ [15]. This bound was obtained in [15] for a natural implementation of the classical algorithm of Gallager, Humblet, and Spira (henceforth called as GHS algorithm) [9]. All the above results assume that nodes do not know their geometric coordinates. If nodes know their own coordinates, then we give an algorithm with $O(1)$ energy complexity that gives an $O(1)$ approximation to the MST. We note that $\Omega(1)$ is a lower bound on the energy complexity of any distributed MST algorithm (even with nodes knowing their coordinates) since any algorithm has to communicate at least once using the tree edges of an MST. For an instance specified by the set $V$ of nodes, we denote this lower bound as $LMST(V) = \sum_{(u,v)\in MST(V)} (d(u,v))^2$, where $MST(V)$ denotes the minimum Euclidean spanning tree on $V$. If the nodes are distributed uniformly at random, it is well-known that $\sum_{(u,v)\in MST(V)} (d(u,v))^2 = \Omega(1)$ (e.g., see [15]).

**Related Work.** This paper is inspired by the prior work of Khan et al. [15] on distributed construction of spanning trees in wireless and sensor networks. We refer to this work and the references therein for more background on energy-efficient distributed algorithms for wireless sensor networks, in particular, on the connections to other related work on wireless sensor and ad hoc networks, and on the models and problems (exact and approximate MST) addressed in the present paper. It is shown in [15] that although message complexity of a distributed algorithm directly influences the energy complexity, algorithms that have optimal message complexity are not necessarily energy optimal. The message-optimal GHS algorithm [9] uses $O(n \log n + |E|)$ messages; however, this algorithm requires $\Omega(\log^2 n)$ energy on average under random distribution [15]. In contrast, in this paper, we show that there is an algorithm that takes $O(\log n)$ energy on the average and this is asymptotically optimal. There are distributed algorithms that construct the MST optimally in terms of time complexity (see, e.g., [7, 23]). But these algorithms require much more messages than GHS algorithm, and consequently require a lot more energy. The distributed algorithm of [14, 15] requires only $O(\log n)$ energy, but it gives an $O(\log n)$-approximation to the MST. The work of [15] raised the question of whether there exists a distributed algorithm of $O(\log n)$ energy complexity and this paper answers this in the affirmative.

## IV. Lower Bound

We show a non-trivial lower bound of $\Omega(\log n)$ on the energy required by any distributed algorithm to construct any spanning tree of the network ($\Omega(1)$ lower bound is trivial, as mentioned in Section III). This bound holds under the following assumptions: (1) the model is synchronous (hence the lower bound applies to asynchronous model as well); (2) any non-empty set of processors may start the algorithm; a processor that is not started remains asleep until a message reaches it and can be awakened spontaneously at any time; (3) no assumption is made on the size of the messages; this assumption only strengthens our bound; (4) nodes do not have any information on their geometric coordinates.

Our lower bound is based on a classical lower bound due to Korach et al. [17], which shows that $\Omega(n \log n)$ messages are needed by any distributed algorithm for constructing a spanning tree (or equivalently, leader election) in a complete network. Thus we have the same assumptions that were made in [17]. Among the assumptions, the second one is crucial. In fact, if all nodes that spontaneously start the algorithm are awakened at the same time, an $O(n)$ upper bound can be shown by using a technique similar to the one in [8]. Thus, under this assumption, the Korach et al. lower bound does not hold, and consequently, our $\Omega(\log n)$ energy lower bound will not hold either.

*Theorem 4.1:* Any distributed algorithm needs $\Omega(\log n)$ energy WHP to construct a spanning tree.

*Proof:* Korach et al. lower bound shows that $\Omega(n \log n)$ *different* edges need to be used by any algorithm. This bound can be shown to apply for Las Vegas type randomized algorithms too.

Our model can also be viewed as complete weighted network, where the weight between any two nodes $u$ and $v$ is $w(u,v) = (d(u,v))^2$. According to the Korach et al. bound, at least $an \log n$ different edges need to be used by any distributed MST algorithm for some fixed constant $a$. To obtain a lower bound on the energy complexity, we compute the minimum energy needed to send at least one message through $n \log n$ different edges. We need the following lemma.

*Lemma 4.1:* For *every* node, WHP, at least $k/bn$ energy is needed if the node wants to communicate with its closest $k$ neighbors, for all $k > a_1 \log n$, where $a_1 < a$ is a fixed positive constant and $b$ is a suitably large constant.

*Proof:* Fix an arbitrary node $v$. Let $X$ be the random variable that denotes the total number of nodes within distance $\sqrt{k/bn}$ of $v$. $E[X] = k/b$. Using a Chernoff bound [21], for suitably large $b$, $\Pr(X \geq k) = \Pr(X \geq (1 + b - 1)k/b) \leq (e/b)^k \leq (e/b)^{a_1 \log n} = o(1/n)$.

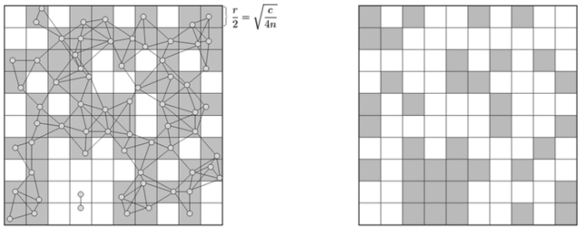
That is, WHP, the number of neighbors of $v$ within distance $\sqrt{k/bn}$ is less than $k$. Hence, if a node wants to communicate with its closest $k$ neighbors, it has to send a message to a distance of at least $\sqrt{k/bn}$ WHP. Thus, the energy needed for this is $k/bn$. By the union bound [21], this holds for every node WHP. ■

We only focus on those nodes that communicate with more than $a_1 \log n$ of its closest neighbors. Let the set of such nodes be denoted by $R$ (relevant set). We ignore the energy spent by the rest of the nodes and focus only on lower bounding the energy needed by the nodes in $R$. Since, the total number of edges used should be at least $an \log n$ and since $a_1 < a$, the nodes in $R$ need to use at least $\Omega(n \log n)$ edges, i.e., communicate with at least $\Omega(n \log n)$ (closest) neighbors. Each node in $R$ communicates with at least $k > a_1 \log n$ neighbors and by Lemma 4.1, it has to spend at least $k/bn$ energy WHP. Thus, WHP, the total energy needed is at least: $\sum_{v\in R} k/bn = 1/bn \sum_{v \in R} k \geq \Omega(\log n)$. ■

## V. An Energy-Optimal Algorithm

In this section, we give an energy-optimal distributed MST algorithm of energy complexity $O(\log n)$, matching the lower bound shown in the previous section.

We assume that graph $G$ (cf. Section II) is connected by setting the transmission radius to the $r$ value given below.



(a) giant component (b) small regions

Fig. 1. A giant component and small regions

Theorem 5.1 shows that this guarantees the connectivity of random geometric graphs.

*Theorem 5.1:* [12, 22] If $r = \sqrt{\frac{c_2 \log n}{n}}$, where $c_2$ is a constant larger than $4$, then the graph is connected whp.

Our algorithm crucially depends on the following Theorem 5.2. It essentially says that, if $r = \sqrt{\frac{c_1}{n}}$ (for some constant $c_1$), then there will be a unique giant component and other small components. Refer to Figure 1. In Figure 1(a), the giant component is shown. A maximal connected cluster of white cells in Figure 1(a) is called a *small region*. In Figure 1(b), these small regions are represented as gray cells. All small components are inside such small regions, and moreover there are not too many small components in any one small region.

*Theorem 5.2:* There is a positive constant $c_1$ such that, if $r = \sqrt{\frac{c_1}{n}}$, there is a unique giant component containing $\Theta(n)$ nodes whp. Furthermore, whp, all remaining components of nodes are trapped inside small regions each containing at most $\beta \log^2 n$ nodes, for some positive constant $\beta$.

The theorem is similar to Theorem 1 in [25], but the conditions are different. In our model two nodes are connected to each other if they are within $r = \sqrt{\frac{c_1}{n}}$ (for some constant $c_1$) of each other, whereas in [25] each node is connected to the $K$ closest nodes where $K$ is some fixed constant (independent of $n$). We prove Theorem 5.2 in Section V-B.

Our distributed MST algorithm consists of two steps, each of which uses the GHS algorithm with some modifications. For constants $c_1$, $c_2$, and $\beta$ (as defined in Theorems 5.1 and 5.2), our algorithm works as follows. (The modified GHS algorithm is described in Section V-A.)

**Step 1:**

1. Each node sets its radius to $\sqrt{\frac{c_1}{n}}$ (i.e., it communicates with nodes only within this distance).
2. Run the modified GHS algorithm.

**Step 2:**

1. Each component computes its size, the number of nodes it contains. If the size is greater than $\beta \log^2 n$, it considers itself as a giant component.
2. Each node increases its radius to $\sqrt{\frac{c_2 \log n}{n}}$.
3. Run the modified GHS algorithm on the remaining component. (The giant component does not participate but only accepts connection messages from small components.)

The main idea of our algorithm is based on the fact that, after the first step, with high probability, there will be one unique giant component and other small components, and that those small components will be trapped inside small regions, each of which contains at most $O(\log^2 n)$ nodes. In the second step, the small components in each small region are merged with each other in the same small region or with the giant component, and eventually all nodes will be connected whp. By controlling the transmission radius in each step, we bound the energy complexity as in the following theorem.

*Theorem 5.3:* Our algorithm constructs an optimal MST using $O(\log n)$ energy on average and $O(\log n \cdot \log \log n)$ energy whp.

The correctness of our algorithm immediately follows from Theorem 5.1 and the correctness of GHS algorithm. The proof of energy complexity is given in Section V-C.

### A. Modified GHS Algorithm

In the modified GHS algorithm, most of steps are the same as those in the original GHS algorithm [9]. We briefly recall the key details of the algorithm. Initially each node is considered to be a fragment (or a connected component). As the edges are added, the fragments grow by combing smaller fragments. In each "round" of the algorithm, each fragment finds its minimum length outgoing edge (MOE) and uses this edge to combine fragments. Each fragment elects its leader (this is known to every node in the fragment) to manage the combining operation. To find the MOE, the leaders of two nodes, which are adjacent to the edge added immediately in the previous step, send initiate message (relayed by the intermediate nodes) to the members of the fragment. Upon receipt of the initiate message, each node tests its adjacent edges to check if the node at the other end is in same fragment. Thus, each member node finds its outgoing edge and reports it to the leaders. Upon receipt of reports, the leaders select a new leader - the node which is adjacent to the MOE for the entire fragment and this begins a new round.

In the modified GHS algorithm, each node additionally keeps a list of its neighbors that are in other fragments with their distance information. In each phase, after two or more fragments are merged, each node sends a message to its neighbors to announce its new fragment id if the id has changed. Each node updates its list when it receives those announcements from its neighbors. This modification enables each node to find its minimum outgoing edge without any additional messages — just by looking up its list and picking up the one with the minimum distance.

Let us compute the message complexity of this modified GHS algorithm. For each node, the number of messages needed to announce its new fragment id is bounded by the total number of phases. The number of initiate messages and report messages is the same as the original algorithm. Thus the total message complexity is $O(n\phi)$ where $n$ is the number of nodes and $\phi$ is the number of phases.

In the modified GHS algorithm in Step 2, two simple techniques are used to reduce the expected energy complexity. Firstly, the giant fragment does not participate but only accepts connection messages from small fragments. Secondly, when small fragments are merged with the giant fragment, small fragments change their ids. That is, the giant fragment keeps its id so that its nodes do not need to announce new ids.

### *B. Proof of Theorem 5.2*

*1) The Giant Component:* We prove the first part of Theorem 5.2. The overall proof is similar to that in [25]. The basic idea is to reduce our problem to site percolation in a finite box. In the original site percolation problem, we consider an infinite grid of cells, where each site is occupied with probability $p$, and we ask the probability at which an infinite cluster of sites emerges. It is well known that there is a critical probability $p$ (denoted by $p_o$), below which the probability that an infinite cluster exists is asymptotically 0 and above which the probability is asymptotically 1. It is also known that in the supercritical phase ($p > p_o$), with high probability, there is a unique giant cluster in the box and that its complement consists of small regions, each containing $O(\log^2 n)$ sites [11].

To do this reduction, we first replace the uniform distribution of nodes with a Poisson distribution to exploit the strong independence property of the latter. That is, a distribution of nodes in one region does not affect the distribution of nodes in any other disjoint region. There is an easy way to connect these two settings (cf. [25]), and we can safely assume that we have $n$ nodes that are generated by Poisson processes in a unit square. Here we repeat the same arguments and lemma as [25] because we need them later.

We consider two Poisson processes $P_0$ and $P_t$. Process $P_0$ has parameter $\mu_0 := n - \epsilon n$, where $\epsilon$ is a small positive constant. Process $P_t$ is built on top of $P_0$ by adding to it a new independent Poisson process $\Delta P$ with parameter $2\epsilon n$. It is well known that $P_t$ is a Poisson process with parameter $\mu_t := \mu_0 + 2\varepsilon n = n + \varepsilon n$. We then define a sequence of point processes $\{Q_i\}$ sandwiched between $P_0$ and $P_t$. Starting from $Q_0 := P_0$, $Q_{i+1}$ is given by $Q_i$ by adding one point chosen uniformly at random in $P_t - Q_i$. Our reduction to site percolation will apply simultaneously to all $Q_i$'s, showing the existence of a unique giant component for each $Q_i$ with high probability. Each $Q_i$ generates points uniformly in the box (conditioned on the given number of points). The next lemma shows that, with high probability, one of the $Q_i$ will generate exactly $n$ points. As a consequence, if something holds for all $Q_i$'s simultaneously, it also holds for the original $n$-nodes problem.

*Lemma 5.1:* [25] Let $N_0$ and $N_t$ be the Poisson variables relative to $P_0$ and $P_t$, respectively. There is a positive constant $\gamma$ such that $Pr\left(\overline{\{N_0 \le n \le N_t\}}\right) \le e^{-\gamma n}$.

We now introduce site percolation problem by subdividing the unit square into a grid of non-overlapping square cells as shown in Figure 1(a). Let $r = \sqrt{\frac{c}{n}}$ be the transmission radius where $c$ is a constant, which will be fixed. Setting the transmission radius $r = \sqrt{\frac{c}{n}}$, we get an infinite grid of cells as $n$ grows. To simplify our analysis, we define the distance between two nodes $u = (x_1, y_1)$ and $v = (x_2, y_2)$ as $\max(|x_1 - x_2|, |y_1 - y_2|)$ instead of using Euclidean distance. This simplification affects our energy complexity bounds only up to a constant factor. We set the length of a side of each cell to $\frac{r}{2}$ so that any two nodes in the neighboring cells are connected. Thus, all nodes in a cluster of occupied cells will be a connected component. The expected number of nodes in each cell is $\frac{c}{4}$. Let us define a cell to be *good* if the number of nodes inside the cell is greater than or equal to $\frac{c}{8}$.

*Lemma 5.2:* Let $p_c$ be the probability that a cell is good. Then $\lim_{c\to\infty} p_c \to 1$.

Now we establish the theorem about the giant component by showing that the largest cluster of good cells form a giant component of nodes. Clearly, the giant component also includes the nodes in any occupied cells that are connected to the largest cluster of good cells.

*Lemma 5.3:* Let $\mathcal{G}$ be the largest component when the transmission radius $r = \sqrt{\frac{c}{n}}$. For any constant $\alpha \in (\frac{1}{4}, \frac{1}{2})$, there is a $c$ s.t. for a positive constant $\gamma$, $Pr(|\mathcal{G}| \le \alpha n) \le e^{-\gamma\sqrt{n}}$.

*Proof:* Let $m \simeq \frac{4n}{c}$ be the number of cells and $C$ be the largest cluster of good cells. By Theorem 1.1 in [6], for any given constant $\delta \in (0, \frac{1}{2})$, there is a value of $p_c$ such that

$$Pr(|C| \le (1-\delta)m) \le e^{-\gamma_1\sqrt{m}}.$$

By definition, a good cell contains at least $\frac{c}{8}$ nodes. Thus, if $C$ contains at least $(1-\delta)m$ good cells, then its corresponding component contains at least $(1-\delta)cm/8$ nodes.

$$\begin{aligned} Pr(|\mathcal{G}| \le \alpha n) &\le Pr(|C| \le 2\alpha \cdot \frac{4n}{c}) + Pr(\overline{\{N_0 \le n \le N_t\}}) \\ &\le \ e^{-\gamma_1\sqrt{4n/c}} + e^{-\gamma_2 n} \le e^{-\gamma\sqrt{n}} \end{aligned}$$

for some positive constant $\gamma$ and large $n$. The first inequality follows from the fact that we do the reduction only if the condition $Pr(\{N_0 \le n \le N_t\})$ holds. ■

*2) The Small Regions:* We prove the second part of Theorem 5.2. We assume that we have chosen the constant $c_1$ so that there exists a giant component with high probability. Let us consider the complement of the largest cluster of good cells. We now show that the maximal connected clusters of cells in the complement of $C$ are small clusters. We call this maximal connected cluster a *small region*. In Figure 1(b), gray area represents these small regions. Definitely small components of nodes will be inside this small region. The two lemmas below bound the number of cells and nodes in a small region, respectively.

*Lemma 5.4:* Let $|S|$ be the number of cells in region $S$. For any small region $S$ and some positive $\gamma$, $Pr(|S| = k) \le e^{-\gamma\sqrt{k}}$.

*Proof:* It follows from the result in the supercritical phase for site percolation [11]. ■

*Lemma 5.5:* Let $Z_i$ be the random variable representing the number of nodes in cell $i$ and $S$ a small region. For large $n$, there is a positive $\gamma$ s.t. $Pr(\sum_{i\in S} Z_i > h) \le e^{-\gamma\sqrt{h}}$.

*Proof:* $Pr(\sum_{i\in S} Z_i > h)$

$$\begin{aligned} &= \sum_{k\le\frac{2h}{c}} Pr(\sum_{i\le k} Z_i > h) Pr(|S| = k) \\ &\quad + \sum_{k>\frac{2h}{c}} Pr(\sum_{i\le k} Z_i > h) Pr(|S| = k) \\ &\le \sum_{k\le\frac{2h}{c}} Pr(\sum_{i\le k} Z_i > h) + \sum_{k>\frac{2h}{c}} Pr(|S| = k) \\ &\le \sum_{k\le\frac{2h}{c}} Pr(\sum_{i\le\frac{2h}{c}} Z_i > h) + \sum_{k>\frac{2h}{c}} Pr(|S| = k) \\ &\le \frac{2h}{c} e^{-\gamma_1 h} + \sum_{k>\frac{2h}{c}} e^{-\gamma_2\sqrt{k}} \le e^{-\gamma\sqrt{h}}. \end{aligned}$$

The second last inequality follows from the large deviation principle and Lemma 5.4. ■

The following lemma completes the proof of Theorem 5.2. Let us consider the event $\mathcal{E}$: with the transmission radius $r = \sqrt{\frac{c}{n}}$, there is a unique giant component containing at least $\alpha n$ nodes and all remaining components of nodes are trapped inside small regions, each of which contains at most $\beta \log^2 n$ nodes.

*Lemma 5.6:* When $n$ is large, for every $\alpha \in (\frac{1}{4}, \frac{1}{2})$ and a positive constant $d$, $Pr(\overline{\mathcal{E}}) \leq n^{-d}$.

*Proof:* By Lemma 5.3, the probability that there is no component with at least $\alpha n$ nodes is at most $e^{-\gamma_1 \sqrt{n}}$. By Lemma 5.5 and union bound, the probability that there exists a small region with more than $\beta \log^2 n$ nodes is at most $ne^{-\gamma_2 \sqrt{\beta \log^2 n}} = n^{1-\gamma_2\sqrt{\beta}}$. Thus, $Pr(\overline{\mathcal{E}}) \leq e^{-\gamma_1\sqrt{n}} + n^{1-\gamma_2\sqrt{\beta}} \leq n^{-d}$ for some positive constant $d$ and large $n$ by choosing $\beta$ appropriately. ■

### C. Energy Complexity Analysis

We first give a high probability analysis. In Step 1, the total number of phases in the modified GHS algorithm is $O(\log n)$, and consequently the total number of messages is $O(n \log n)$. Sending one message requires $O(\frac{1}{n})$ energy since the transmission radius is $O(\sqrt{1/n})$. Therefore, the total energy required in Step 1 is $O(\log n)$. At the beginning of Step 2, each fragment needs to compute its size. This can be done with one broadcast and one convergecast, which need $O(n)$ messages and consequently $O(1)$ energy in total.

We now compute the energy required by the modified GHS algorithm in Step 2. It is shown that the number of nodes in a small region is at most $O(\log^2 n)$ whp. Thus, the number of fragments in a small region is at most $O(\log^2 n)$, and each small fragment just needs to connect only with other small fragments in the same small region or the giant fragment. Therefore, the total number of phases in the modified GHS algorithm is at most $O(\log \log n)$ whp. Thus the total number of messages needed in Step 2 is $O(n \log \log n)$ whp. The energy needed for each message is $O(\frac{\log n}{n})$ since we increased the transmission radius to $O(\sqrt{\frac{\log n}{n}})$. Thus, the total energy required in Step 2 is $O(\log n \cdot \log \log n)$. Therefore, the overall energy complexity is $O(\log n \cdot \log \log n)$ whp.

Now we show that the expected energy complexity is $O(\log n)$. The expected energy required by the modified GHS algorithm in Step 1 and the computation of each fragment's size in Step 2 is clearly $O(\log n)$. Thus it suffices to show that the expected energy required by the modified GHS algorithm in Step 2 is $O(\log n)$. This can be shown from the following lemma.

*Lemma 5.7:* In the modified GHS algorithm used in Step 2, the expected number of messages needed by all nodes in any one small region is a constant.

*Proof:* Let $S$ be a small region containing a node $v$ at the end of Step 1. Let also $N_v$ be the total number of messages needed by $v$ during the modified GHS algorithm in Step 2. Then, $N_v$ is bounded by $c \log F_S$ where $c$ is some constant and $F_S$ is the number of fragments in $S$ at the end of Step 1. Let $N_S$ be the number of messages needed by all nodes in $S$, that is, $N_S = \sum_{v \in S} N_v$. As in Lemma 5.5, $Z_i$ represents the number of nodes in cell $i$. Thus $F_S$ is bounded by $\sum_{i \in S} Z_i$, which is the total number of nodes in $S$. The following inequalities complete the proof by showing that $\mathbf{E}[N_S]$ is bounded by some constant.

$$\begin{aligned} \mathbf{E}[N_S] &\leq \mathbf{E}[(\sum_{i \in S} Z_i) c \log F_S] \leq c\mathbf{E}[(\sum_{i \in S} Z_i) \log (\sum_{i \in S} Z_i)] \\ &\leq c \sum_h h \log h \cdot Pr(\sum_{i \in S} Z_i \geq h) < \infty. \end{aligned}$$

The bound on $Pr\left(\sum_{i \in S} Z_i \geq h\right)$ follows from Lemma 5.5. ■

By Lemma 5.7, it follows that the expected energy required to connect all nodes in one small region is $O(\frac{\log n}{n})$. Since there are at most $O(n)$ small regions, the required energy for all nodes in all small regions is $O(\log n)$. The energy needed by all nodes in the giant fragment is $O(\log n)$ since there are at most $O(n)$ messages for accepting connection requests from small fragments. Therefore, the total expected energy required is $O(\log n)$. This completes the proof of Theorem 5.3.

## VI. An $O(1)$ Approximation Algorithm with $O(1)$ Energy Complexity

We showed that the lower bound on energy complexity for distributed construction of any spanning tree, hence also MST, is $\Omega(\log n)$. However, if some additional information such as coordinates of the nodes is given to the nodes, a more energy-efficient algorithm can be developed. We present a distributed algorithm to construct a spanning tree assuming that each node knows its own coordinates. This spanning tree gives a constant approximation to MST, and the energy complexity of the algorithm is also constant (this is the best possible energy complexity — cf. Section III).

Let the nodes are distributed uniformly at random in a unit square with lower-left corner at $(0, 0)$ and upper-right corner at $(1, 1)$ (see Figure 2a). Each node $v$ knows its coordinates $(x_v, y_v)$. We define the *ranks* of the nodes as follows: for any two nodes $u$ and $v$, rank($u$) $<$ rank($v$) iff $(x_u + y_u < x_v + y_v)$ or $(x_u + y_u = x_v + y_v$ and $y_u < y_v)$. Assuming that no two nodes have the same coordinates, for any pair of nodes $u$ and $v$, either rank($u$) $<$ rank($v$) or rank($v$) $<$ rank($u$). To build the spanning tree, each node, except the node with the highest rank, is connected to the nearest node of higher rank. It is easy to see that in such a construction, the resulting graph is a single connected component with no cycle, i.e., a tree. This tree is called *nearest neighbor tree* (NNT) (cf. [15]). In [15], an NNT is constructed using a different ranking: rank($u$) $<$ rank($v$) iff $(x_u < x_v)$ or $(x_u = x_v$ and $y_u < y_v)$, which also gives us constant approximation and constant energy complexity. However, in that ranking, there are few nodes that need to go far away to find the nearest node of higher rank. As a result, it is not suitable for the unit disk graph model with $r = \Theta(\sqrt{\frac{\log n}{n}})$ that we are using in this paper. With our modified ranking scheme, we show that every node finds the nearest node of higher rank within distance $r = \Theta(\sqrt{\frac{\log n}{n}})$ with high probability (see Lemma 6.3 below). To achieve our goal with this modified ranking of the nodes requires an entirely different technique to prove the bounds as given below.

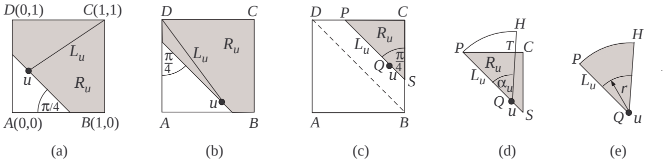


Fig. 2. (a), (b), (c): The potential region $R_u$ marked by dark color and the potential distance $L_u$ for an arbitrary node $u$. (d): A pie slice with area equal to the area of the potential region shown in (c); $\alpha_u$ is the potential angle for $u$.

Consider an arbitrary node $u$ as shown in Figure 2. The straight line $x+y = x_u+y_u$, which passes through $u$ making equal angles with both axes, divides the unit square into two regions. The region in the half plane $x+y > x_u+y_u$, the dark region in the figure, is called the *potential region* for $u$, denoted by $R_u$. Any node in $R_u$ has higher rank than $u$, and $u$ is connected to the nearest node in $R_u$. The area of $R_u$ is called the *potential area* for $u$, denoted by $A_u$. The distance to the farthest point in $R_u$ from $u$ is called the *potential distance* for $u$, denoted by $L_u$. Now, as shown in Figure 2(d), consider a pie slice with angle $\alpha_u$ (in radian) of the circle with center $u$ and radius $L_u$ such that the area of the pie slice equals the potential area for $u$, i.e., $\frac{1}{2}\alpha_u L_u^2 = A_u$. that is, $\alpha_u = \frac{2A_u}{L_u^2}$. Angle $\alpha_u$ is called the *potential angle* for $u$.

*Lemma 6.1:* For any $u$, the potential angle $\alpha_u \geq \frac{1}{2}$ radian.

*Proof:* For a node $u$ with $x_u+y_u \geq 1$, i.e., $u$ is in triangle $BCD$ at some point $Q$ as shown in Figure 2(c): $A_u = \triangle PSC$ and $L_u \leq PS$. Thus, $\alpha_u = \frac{2A_u}{L_u^2} \geq \frac{1}{2}$. If $x_u+y_u < 1$, i.e., $u$ is in triangle $ABD$, $A_u \geq \triangle BCD$, $L_u \leq BD$, and thus $\alpha_u \geq \frac{1}{2}$. ■

*Lemma 6.2:* If $d_u$ denotes the distance from $u$ to the nearest node in the potential region $R_u$, $E[d_u^2] \leq \frac{2}{n\alpha_u}$.

*Proof:* Consider Figure 2(d). By construction, the area of the pie slice $PQH$ with angle $\alpha_u$ and radius $L_u$ is equal to area of the potential region $R_u$, which is region $PSC$. Thus, the areas of regions $PTH$ and $TQSC$ are equal. Now remove the nodes from region $TQSC$ and place them in region $PTH$ uniformly at random. In this process, we are only moving some nodes away from $u$. Thus if $d'_u$ denotes the distance from $u$ to the nearest node in the pie slice $PQH$, we have $d_u \leq d'_u$. Now we compute $E[d'^2_u]$.

Consider the region in the pie slice $PQH$ within distance $r$ from $u$ as shown in Figure 2(e). By uniform distribution, the probability that a particular node resides in a particular region is equal to the area of that region since the area of the unit square $ABCD$ is 1. Thus, the probability that there is at least one node, other than $u$, within distance $r$ from $u$ in the pie slice, is given by $F(r) = 1 - \left(1 - \frac{1}{2}\alpha_u r^2\right)^{n-1}$. Then, the density function

$$\begin{aligned}
f(r) &= \frac{d}{dr}F(r) = (n-1)\alpha_u r\left(1 - \tfrac{1}{2}\alpha_u r^2\right)^{n-2} \\
E[d_u^2] &\leq E[d'^2_u] = \int_0^{L_u} r^2 f(r)\,dr \\
&= \frac{2}{n\alpha_u}\left\{1 - nx^{n-1} + (n-1)x^n\right\} \leq \frac{2}{n\alpha_u}
\end{aligned}$$

where $x = 1 - \frac{1}{2}\alpha_u L_u^2$. The last inequality follows from the fact that $0 \leq x \leq 1$. ■

Using the above lemmas, in the following theorem, we show that expected sum of the squared edge lengths of the NNT is constant. It is well-known that the expected sum of the squared edges of MST is $\Theta(1)$ [26] when the nodes are distributed in a unit square uniformly at random. Thus NNT gives us a constant approximation.

*Theorem 6.1:* The expected sum of the squared edge lengths of the NNT, $E\Bigg[\sum\limits_{e\in\text{NNT}} |e|^2\Bigg] = O(1)$.

*Proof:* $E\Bigg[\sum\limits_{e\in\text{NNT}} |e|^2\Bigg] = E\Bigg[\sum\limits_{u\in V} d_u^2\Bigg] = \sum\limits_{u\in V} E\Big[d_u^2\Big] = nE\Big[d_u^2\Big] \leq 4$, by Lemma 6.1 and 6.2. ■

Using a slightly different technique, we can show that the expected sum of the edge lengths (i.e., the case of Euclidean MST, in contrast to the sum of the *squared* edge lengths) of the NNT, $E\big[\sum_{e\in\text{NNT}} |e|\big] = O(\sqrt{n})$. For MST, $E\big[\sum_{e\in\text{MST}} |e|\big] = \Theta(\sqrt{n})$ [26]. Thus, in this case, we also have constant approximation to MST.

In the following lemma, we show that all nodes find the nearest node in their potential regions within distance $\Theta(\sqrt{\frac{\log n}{n}})$ with high probability. That is, with high probability, the NNT can be constructed in unit disk graph, where the transmission radius for each node is $\Theta(\sqrt{\frac{\log n}{n}})$.

*Lemma 6.3:* Simultaneously for all $u \in V$, $d_u \leq c\sqrt{\frac{\log n}{n}}$ with probability at least $1 - \frac{1}{n^{c^2/8-1}}$.

*Proof:* Let $r = c\sqrt{\frac{\log n}{n}}$. Consider an arbitrary node $u$. If $L_u < r$, then $\Pr\{d_u \leq r\} = 1$. Assume that $L_u \geq r$. Now, $\Pr\{d_u \leq r\}$ is larger or equal to the probability that there is at least one node in $R_u$ within distance $r$ from $u$. Thus,

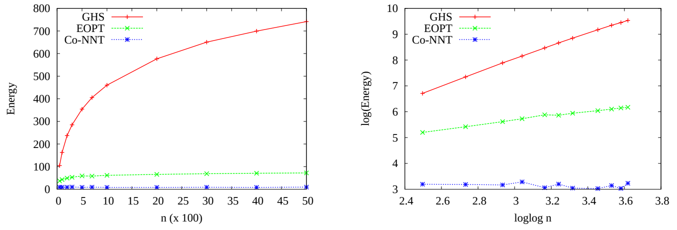
$$\begin{aligned}
\Pr\{d_u \leq r\} &\geq 1 - \left(1 - \tfrac{1}{2}\alpha_u r^2\right)^{n-1} \\
&\geq 1 - e^{\frac{1}{2}\alpha_u r^2(n-1)} \geq 1 - \frac{1}{n^{c^2/8}}.
\end{aligned}$$

Using the union bound, $d_u \leq r$ holds simultaneously for all $u \in V$ with probability at least $1 - \frac{1}{n^{c^2/8-1}}$. ■

In the following theorem, we show that we can devise a distributed algorithm to construct NNT with constant energy complexity.

*Theorem 6.2:* There is a distributed algorithm to construct NNT with expected energy complexity $O(1)$ and message complexity $O(n)$.

*Proof:* Consider the following algorithm. Assume that each node $u$ knows its potential distance $L_u$ and the number of nodes $n$ — node $u$ can locally compute exact $L_u$ from its coordinates and a rough approximation for $n$ will work; the bounds on energy and messages hold as long as approximate value for $n$ is $\Theta(n)$. To find the nearest node in the potential region $R_u$, each node $u$ transmits a *request* message containing its coordinates $(x_u, y_u)$ to distance $r_i = \sqrt{\frac{2^i}{n}}$ in rounds $i = 1, 2, \ldots, m = \lceil \lg nL_u^2 \rceil$. Any node $v$ within distance $r_i$ can hear the message and replies back to $u$ if $\text{rank}(u) < \text{rank}(v)$, i.e., $v$ is in $R_u$. If $u$ gets back replies from one or more nodes, it selects the nearest node among them and sends a *connection* message to it, and stops exploration; otherwise, $u$ continues to



(a) (b)

Fig. 3. Energy consumed by three different algorithms: GHS, our optimal algorithm EOPT and our approximtion Co-NNT. Figures (a) and (b) show the plots in normal and logscale, respectively.

the next phase, $(i+1)$st phase. If $u$ does not find any node in $R_u$ within distance $L_u$ (it happens only to the highest ranked node), it terminates anyway.

Node $u$ needs the first transmission with probability $p_i = 1$. For $i \geq 2$, $u$ needs the $i$th transmission only if there is no node within distance $r_{i-1}$ in $R_u$. That is, the probability that $u$ needs $i$th transmission is

$$p_i \leq \left(1 - \tfrac{1}{2}\alpha_u r_{i-1}^2\right)^{n-1} \leq \left(1 - 2^{i-3}/n\right)^{n-1} \leq e^{-2^{i-4}}.$$

The expected number of nodes in $R_u$ within distance $r_i$ is at most $\frac{1}{2}\pi r_i^2 n = 2^{i-1}\pi$. Thus, the expected number of replies $u$ receives is at most $2^{i-1}\pi$. Additionally, $u$ sends at most one *request* message and one *connection* message in each phase. That is, the expected number of messages in phase $i$, $E[M_i] \leq 2 + 2^{i-1}\pi$. Therefore, the expected number of messages for all $n$ nodes is at most

$$n\sum_{i=1}^{m} p_i E[M_i] = n(2+\pi) + n\sum_{i=2}^{m}(2+2^{i-1}\pi)e^{-2^{i-4}} = O(n).$$

The expected energy is at most

$$n\sum_{i=1}^{m} p_i E[M_i] r_i^2 = 2(2+\pi) + \sum_{i=2}^{m}(2+2^{i-1}\pi)2^i e^{-2^{i-4}} = O(1).$$

■

## VII. Experimental Results

We performed the simulation of our algorithms to understand their empirical performance. For comparison, we also simulated the original GHS algorithm. Our experimental setup is the following: a) the number of nodes varies from 50 to 5000, and b) the nodes are uniformly randomly distributed in a unit square. We measured the total energy used by each algorithm – GHS, our energy-optimal algorithm (EOPT), and our approximation algorithm (Co-NNT). The input to GHS and EOPT algorithm must be a connected graph to obtain an MST. Thus, we set the radius to $1.6\sqrt{\frac{\ln n}{n}}$ for GHS and the increased radius of EOPT. We set the initial radius of EOPT to $1.4\sqrt{\frac{1}{n}}$ to have a giant component after the first step.

Figure 3(a) shows the energy consumed by each algorithm. Observe that both EOPT and Co-NNT significantly improve on the energy usage, relative to GHS. Analytically, we know that the expected energy complexities for GHS, EOPT, and Co-NNT are $O(\log^2 n)$, $O(\log n)$, and $O(1)$, respectively. We can observe these results from our experimental data in Figure 3(b): Let energy $W = c\log^b n$, that is, $\log W = \log c + b\log\log n$. Thus if we plot $\log W$ vs. $\log\log n$, we have a straight line with slope $b$, the power of $\log$. In Figure 3(b), we see the slopes are about $2$, $1$, and $0$ for GHS, EOPT, and Co-NNT, respectively, which conform with our analytical results.

In our experiments, we also observe that Co-NNT gives a very close approximation to MST. The sum of the edges of Co-NNT for 1000 and 5000 nodes are $22.9$ and $50.5$, and that of MST are $20.8$ and $46.3$, repectively. The sum of the squared edges of both Co-NNT and MST are constants (independent of $n$), which are $0.68$ and $0.52$, respectively.

## VIII. Concluding Remarks and Further Work

This work addresses the energy complexity of distributed algorithms, a measure that is very relevant to energy-constrained ad hoc wireless and sensor networks. The main goal of this work is to understand lower and upper bounds on the energy complexity of the minimum spanning tree problem, a fundamental communication primitive. We showed that, without coordinate information, the lower bound on energy complexity to construct any spanning tree, hence also MST, is $\Omega(\log n)$. We then present an algorithm that matches this lower bound. With coordinate information, the best known lower bound is $\Omega(1)$ (which is trivial). We then showed a constant energy algorithm that gives a constant approximation to the MST. An important open question is whether there is an energy-optimal algorithm to construct an (exact) MST when the coordinates are given to the nodes.

In this paper, the main focus has been on transmission energy in an abstract model of the energy complexity. However, this alone does not fully capture the energy needed, as it ignores the energy requirements for receiving and staying awake, which can be significant. As part of future work, we plan to extend our results by incorporating a more accurate energy model (e.g., [20]).

In this paper, we assume the unit disk model, and have ignored the complexity of wireless interference. This can be handled in various ways. By combining the contention resolution protocol of [15] along with the results of this paper, we can get distributed algorithms for spanning tree construction, with an increase in the running time by an $O(n\log n)$ factor and in the energy usage by a constant factor, in the Radio Broadcast Model (RBN) of interference. However, it has been shown that the RBN model has several limitations, and the *Physical Interference model based on SINR constraints* has been proposed to rectify some of these [10]; our future goal is to develop energy efficient distributed algorithms in the above model.

**Acknowledgement** We would like to thank the referees for their valuable comments.

## References

[1] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50:2127–2159, 2006.

[2] H. Balakrishnan, C. Barrett, V. S. Anil Kumar, M. Marathe, and S. Thite. The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc networks. *IEEE Journal on Selected Areas in Communication*, 22(6):1069–1079, 2004.

[3] M. Cheng, M. Cardei, X. Cheng, L. Wang, Y. Xu, and D. Du. Topology control of ad hoc wireless networks for energy efficiency. *IEEE Transactions on Computers*, 53(12):1629–1635, December 2004.

[4] Y. Choi, M. Khan, V.S.A. Kumar, and G. Pandurangan. Energy-optimal distributed algorithms for minimum spanning trees. Technical report, Dept. of Computer Science, Purdue University, 2009.

[5] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the complexity of computing minimum energy consumption broadcast subgraph. In *18th STACS*, pages 121–131, June 2001.

[6] J. Deuschel and A. Pisztora. Surface order large deviations for high-density percolation. *Probability Theory and Related Fields*, 104(4):467–482, 1996.

[7] M. Elkin. A faster distributed protocol for constructing minimum spanning tree. In *SODA*, pages 352–361, 2004.

[8] Greg N. Frederickson and Nancy A. Lynch. Electing a leader in a synchronous ring. *J. ACM*, 34(1):98–115, 1987.

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

[10] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric sinr. In *Proceedings of MOBIHOC*, pages 100–109, 2007.

[11] G. Grimmet. *Percolation*. Springer-Verlag, 1999.

[12] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity. In *Proceedings of the Conference on Decision and Control*, 1998.

[13] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *ICDCS*, July 2002.

[14] M. Khan and G. Pandurangan. A fast distributed approximation algorithm for minimum spanning trees. *Distributed Computing*, 20(6):391–402, April 2008.

[15] M. Khan, G. Pandurangan, and V.S.A. Kumar. Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(1):124–139, January 2009.

[16] L. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243(1-2):289–305, July 2000.

[17] E. Korach, S. Moran, and S. Zaks. Optimal lower bounds for some distributed algorithms for a complete network of processors. *Theoretical Computer Science*, 64:125–132, 1989.

[18] B. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *International Workshop on Distributed Event-Based Systems*, July 2002.

[19] X. Li. Algorithmic, geometric and graphs issues in wireless networks. *Journal of Wireless Communications and Mobile Computing*, 3(2):119–140, 2003.

[20] R. Min and A. Chandrakasan. Top five myths about the energy consumption of wireless communication. In *Mobile Computing and Communications Review*, 2003.

[21] M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[22] S. Muthukrishnan and G. Pandurangan. The bin-covering technique for thresholding random geometric graph properties. In *SODA*, 2005.

[23] D. Peleg. *Distributed Computing: A Locality Sensitive Approach*. SIAM, 2000.

[24] P. Santi. *Topology Control in Wireless Ad hoc and Sensor Networks*. John Wiley & Sons, Inc., New York, NY, 2005.

[25] E. Santis, F. Grandoni, and A. Panconesi. Fast low degree connectivity of ad-hoc networks via percolation. In *ESA*, pages 206–217, 2007.

[26] M. Steele. Asymtotics for euclidian minimal spanning trees on random points. *Probability Theory and Related Fields*, 92:247–258, 1992.

[27] P. Wan, G. Calinescu, X. Li, and O. Frieder. Minimum-energy broadcasting in static ad hoc wireless networks. *Wireless Networks*, 8(6):607–617, November 2002.

[28] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *INFOCOM*, April 2001.

**Yongwook Choi** is a Ph.D. candidate in the Department of Computer Science at Purdue University. He received his B.S. in Electrical Engineering and M.S. in Electrical Engineering and Computer Science from Seoul National University in 2002 and 2004, respectively. His research interests include analysis of algorithms, information theory, string algorithms, data compression, and distributed network algorithms.

**Maleq Khan** is currently at the Network Dynamics and Simulation Science Laboratory at Virginia Bioinformatics Institute, Virginia Tech. He received a B.S. in Computer Science and Engineering from Bangladesh University of Engineering and Technology, an M.S. in Computer Science from North Dakota State University, and a Ph.D. in Computer Science from Purdue University. His research interests are in distributed algorithms, randomized algorithms, wireless ad hoc networks, and theoretical computer science in general. Maleq Khan, together with Gopal Pandurangan, received the Best Student Paper Award at the 20th International Symposium on Distributed Computing (DISC 2006). He is a member of IEEE.

**V.S. Anil Kumar** is currently at the Dept. of Computer Science and the Virginia Bioinformatics Institute at Virginia Tech. His interests are in the broad areas of algorithms, combinatorial optimization, probabilistic techniques and distributed computing, and their applications to wireless networks, epidemiology, and the modeling, simulation and analysis of social and infrastructure networks.

**Gopal Pandurangan** is an assistant professor of Computer Science at Purdue University. He has a B.Tech. in Computer Science from the Indian Institute of Technology at Madras (1994), a M.S. in Computer Science from the State University of New York at Albany (1997), and a Ph.D. in Computer Science from Brown University (2002). His research interests are in design and analysis of algorithms, distributed computing and communication networks (especially sensor networks and peer-to-peer networks), Web/Internet algorithmics, and computational biology. On these subjects he has authored or co-authored over 50 publications in refereed journals and conferences. In the 2006 International Symposium on Distributed Computing, he and Maleq Khan won the Best Student Paper Award for their paper titled “A Fast Distributed Approximation Algorithm for Minimum Spanning Trees”. Pandurangan is a member of IEEE.