Brief Announcement: Energy-Optimal Distributed Algorithms for Minimum Spanning Trees

Yongwook Choi* Purdue University
Maleq Khan† Virginia Tech
V.S. Anil Kumar† Virginia Tech
Gopal Pandurangan* Purdue University

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-constraint radio or 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, and design energy-efficient distributed algorithms for energy-constraint networks.

This paper addresses the minimum spanning tree (MST) problem, an important problem in distributed computing. 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 a 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 $\Omega(\log^2 n)$. All 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.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems – *Computations on discrete structures*; G.2.2 [**Discrete Mathematics**]: Graph Theory – *Graph algorithms*

General Terms: Algorithms, Theory

Keywords: Distributed Algorithm, Energy-Efficient, Minimum Spanning Tree, Distributed Approximation Algorithm

*Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA. E-mail: ywchoi@purdue.edu, gopal@cs.purdue.edu

†Network Dynamics and Simulation Science Laboratory, Virginia Bioinformatics Instit., Virginia Tech, Blacksburg, VA 24061, USA. E-mail: {maleq, akumar}@vbi.vt.edu. Maleq Khan and V.S. Anil Kumar were supported in part by NSF Award CNS-0626964.

SPAA'08, June 14–16, 2008, Munich, Germany.
ACM 978-1-59593-973-9/08/06.

1. 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) and 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. 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 (the graph induced is also known as a *random geometric graph* [11]). G can be considered as a weighted unit disk graph. This is a standard geometric graph model that has been widely used in the literature for modeling communications networks in a plane, e.g., wireless and ad hoc (sensor) networks. Without loss of generality, we assume that r is chosen such that G is connected.

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 ith message, and m is the total number of messages exchanged by the algorithm. The weight of an edge is a function of the distance between the two points. Although our results are generalizable to a wide variety of weight functions, for concreteness, we assume that $w(u, v)$ is proportional to some fixed power (denoted as α) of the distance between u and v, i.e., $w(u, v) = a(d(u, v))^{\alpha}$ where $d(u, v)$ is the (Euclidean) distance between u and v, and a is some fixed constant. The motivation for this comes from energy requirements in a radio (wireless) communication paradigm: to transmit a message over a distance r, the required *energy* is (proportional to) r^{α}, where typically α is 2 [7]. In this paper, unless otherwise stated, we assume $\alpha = 2$. (Our results can be generalized to other values of α as well.)

Distributed Computing Model. Each node in G hosts a processor with limited initial knowledge. Each node has unique identity numbers and at the beginning of computation, each vertex v accepts as input its own identity number. Nodes do not even 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). In a one-directional communication, when a node u can send 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 as *local broadcasting* and is a feature of radio and wireless networks. We assume that a node can receive messages from more than one neighbor in the same time step. In this model, we ignore "collisions" between messages; such issues do arise in real-world radio and wireless networks. However, the collisions can easily be resolved by a constant-factor increase in energy complexity of a distributed algorithm [10].

MST Problem. In the above model, we study the Euclidean MST problem: given a network $G = (V, E, w)$, find a tree T spanning V such that $\sum_{(u,v)\in T} d(u,v)$ is minimized. We actually solve a generalized version of the above problem: find a tree T spanning V such that $\sum_{(u,v)\in T} d^{\alpha}(u,v)$ is minimized where α 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 [3]) 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. Note that α plays a role in determining the approximation ratio guarantee (for exact algorithms, it does not matter). Although our results can be generalized to any α, for quality of an MST, we focus on $\alpha = 1$ (the Euclidean MST) and $\alpha = 2$.

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 in broadcasting [10, 1] and data aggregation [8, 10]. In wireless networks, an MST can be used as a communication tree to minimize energy consumption since it minimizes $\sum_{(u,v)\in T} d^{\alpha}(u,v)$.

2. 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 $O(\log n \log\log n)$ 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 $\Omega(\log^2 n)$ [10]. This bound was obtained in [10] for a natural implementation of the classical algorithm of Gallager, Humblet, and Spira (henceforth called as GHS algorithm) [5]. 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 [10]).

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 [5] uses $O(n \log n + |E|)$ messages. It was shown in [10] that this algorithm requires $\Omega(\log^2 n)$ energy on average under random distribution; in contrast 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 [4, 12]. But these algorithms require much more messages than GHS algorithm, and consequently require a lot more energy. The distributed algorithm of [9, 10] requires only $O(\log n)$ energy, but it gives an $O(\log n)$-approximation to the MST. The work of [10] raised the question of whether there exists a distributed algorithm of $O(\log n)$ energy complexity and this paper answers this in the affirmative.

The details of the algorithms, results, and proofs of the theorems can be found in the full version of this paper [2].

3. LOWER BOUND

We show a non-trivial lower bound on the energy complexity in the following theorem ($\Omega(1)$ lower bound is trivial).

THEOREM 3.1. *Any distributed algorithm needs $\Omega(\log n)$ energy whp to construct a spanning tree.*

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; (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.

4. 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 1) is connected by setting the transmission radius $r = \sqrt{\frac{c_2 \log n}{n}}$ for some constant c_2 (cf. [6, 11]).

Our distributed MST algorithm consists of two steps, each of which uses the GHS algorithm with some modifications (described later). For constants c_1, c_2, and β (as defined above and in Theorem 4.1), our algorithm works as follows.

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 $\beta \log^2 n$, then 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.)

Our algorithm crucially depends on the following theorem.

THEOREM 4.1. *There exists a positive constant c_1 such that, if $r = \sqrt{\frac{c_1}{n}}$, then there is a unique giant component containing $\Theta(n)$ nodes whp. Furthermore, whp, all remaining components of nodes are trapped inside small regions, each of which contains at most $\beta \log^2 n$ nodes, for some positive constant β.*

The theorem is essentially similar to Theorem 1 in [13], 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 [13] each node is connected to the K closest nodes where K is some fixed constant (independent of n).

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 $\beta \log^2 n$ nodes. In the second step, the small components in each small region are merged with each others in the same small region or with the giant component, and eventually all nodes will be connected with high probability. By controlling the transmission radius in each step, we bound the energy complexity as in the following theorem.

THEOREM 4.2. *Our algorithm constructs an optimal MST using $O(\log n)$ energy on average and $O(\log n \log\log n)$ whp.*

The correctness of our algorithm follows from the connectivity of the graph and the correctness of GHS algorithm.

We now describe the modified GHS algorithm and analyze its message complexity. In the modified GHS algorithm, most of the steps are the same as those in the original GHS algorithm [5]. The difference is that each node additionally keeps a list of its neighbors that are in other fragments with their distance information. As in [5], a subtree of MST is called a *fragment*. 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.

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 fragment id so that its nodes do not need to announce new ids.

5. AN $O(1)$ ENERGY ALGORITHM

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 energy possible — cf. Section 2).

Nodes are distributed uniformly at random in a unit square with lower-left corner at $(0, 0)$ and upper-right corner at $(1, 1)$. 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, $\text{rank}(u) < \text{rank}(v)$ 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 $\text{rank}(u) < \text{rank}(v)$ or $\text{rank}(v) < \text{rank}(u)$. 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. [10]). In [10], an NNT is constructed using a different ranking: $\text{rank}(u) < \text{rank}(v)$ 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 $r = \Theta(\sqrt{\frac{\log n}{n}})$, the model used 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. The following theorem shows the time and message complexity to construct NNT.

THEOREM 5.1. *There is a distributed algorithm to construct NNT with expected energy complexity $O(1)$ and message complexity $O(n)$.*

6. REFERENCES

[1] C. Ambuhl. An optimal bound for the mst algorithm to compute energy efficient broadcast trees in wireless networks. In *32nd ICALP*, pages 1139–1150, November 2005.

[2] Y. Choi, M. Khan, V.S.A. Kumar, and G. Pandurangan. Energy-Efficient Distributed Minimum Spanning Tree Construction: Tight Bounds and Algorithms. available at `http://www.cs.purdue.edu/people/gopal/emst.pdf`.

[3] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

[4] M. Elkin. A faster distributed protocol for constructing minimum spanning tree. In *SODA*, 2004.

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

[6] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity. In *the Conf. on Decision and Control*, 1998.

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. on Wireless Communications*, 1(4):660–670, October 2002.

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

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

[10] M. Khan, G. Pandurangan, and V. Kumar. Distributed algorithms for constructing approximate minimum spanning trees in wireless networks. *IEEE Trans. on Parallel and Distributed Systems*, 2008 (in press). available at http://staff.vbi.vt.edu/maleq/papers/TPDS.pdf.

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

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

[13] E. Santis, F. Grandoni, and A. Panconesi. Fast low degree connectivity of ad hoc networks via percolation. In *ESA*, 2007.