Brief Announcement: Wake Up and Join Me! An Energy-Efficient Algorithm for Maximal Matching in Radio Networks

Varsha Dani
varshadani@gmail.com
Ronin Institute
Montclair, NJ, U.S.A.

Aayush Gupta*
aayush@cs.unm.edu
Department of Computer Science
University of New Mexico
Albuquerque, NM, U.S.A.

Thomas P. Hayes*
hayes@cs.unm.edu
Department of Computer Science
University of New Mexico
Albuquerque, NM, U.S.A.

Seth Pettie†
seth@pettie.net
Department of EECS
University of Michigan
Ann Arbor, MI, U.S.A.

ABSTRACT

We consider networks of small, autonomous devices that communicate with each other wirelessly. Minimizing energy usage is an important consideration in designing algorithms for such networks, as battery life is a crucial and limited resource. Working in a model where both sending and listening for messages deplete energy, we consider the problem of finding a maximal matching of the nodes in a radio network of arbitrary and unknown topology.

We present a distributed randomized algorithm that produces, with high probability, a maximal matching. The maximum energy cost per node is $O(\log^2 n)$, and the time complexity is $O(\Delta \log(n))$. Here n is any upper bound on the number of nodes, and Δ is any upper bound on the maximum degree; n and Δ are parameters of our algorithm that we assume are known *a priori* to all the processors. We note that there exist families of graphs for which our bounds on energy cost and time complexity are simultaneously optimal up to polylog factors, so any significant improvement would need additional assumptions about the network topology.

We also consider the related problem of assigning, for each node in the network, a neighbor to back up its data in case of eventual node failure. Here, a key goal is to minimize the maximum *load*, defined as the number of nodes assigned to a single node. We present an efficient decentralized low-energy algorithm that finds a neighbor assignment whose maximum load is at most a polylog(n) factor bigger that the optimum.

CCS CONCEPTS

• **Theory of computation** → **Distributed algorithms**.

KEYWORDS

Distributed Algorithms; Energy-Aware Computation; Radio Networks; Maximal Matching; Sensor Networks

ACM Reference Format:
Varsha Dani, Aayush Gupta, Thomas P. Hayes, and Seth Pettie. 2021. Brief Announcement: Wake Up and Join Me! An Energy-Efficient Algorithm for Maximal Matching in Radio Networks. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing (PODC '21), July 26–30, 2021, Virtual Event, Italy.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3465084.3467950

1 INTRODUCTION

For networks of small computers, energy management and conservation is often a major concern. When these networks communicate wirelessly, usage of the radio transceiver to send or listen for messages is often one of the dominant causes of energy usage. Moreover, this has tended to be increasingly true as the devices have gotten smaller; see, for example, [3, 10, 13]. Motivated by these considerations, Chang *et al.* [4] introduced a theoretical model of distributed computation in which each send or listen operation costs one unit of energy, but local computation is free. Over a sequence of discrete timesteps, nodes choose whether to sleep, listen, or send a message of $O(\log n)$ bits. A listening node successfully receives a message only when exactly one of its neighbors has chosen to send in that timestep; otherwise it receives no input.

It is not uncommon for research on sensor networks to make assumptions about the topology of the network, such as assuming the network is defined by a unit disk graph, or that each node is aware of its location using GPS. However, we will be interested in the more general setting where we make almost no assumptions about the network topology. We will assume that communication takes place via radio broadcasts, and that there is an arbitrary and unknown undirected graph G whose edges indicate which pairs of nodes are capable of hearing each other's broadcasts. We will, however, assume that each node is initialized with shared parameters n and Δ, which are upper bounds on, respectively, the total number of nodes, and the maximum degree of any node. By designing algorithms to operate without pre-conditions on, or foreknowledge of,

*Supported in part by NSF CAREER award CCF-1150281.
†Supported in part by NSF grant CCF-1815316.

PODC '21, July 26–30, 2021, Virtual Event, Italy

ACM ISBN 978-1-4503-8548-0/21/07.
https://doi.org/10.1145/3465084.3467950

the network topology, we potentially broaden the possible applications of our algorithms, and, by extension, of sensor networks. For instance, we can imagine a network of small sensors scattered rather haphazardly from an airplane passing over hazardous terrain; the sensors that survive their landing are unlikely to be placed predictably or uniformly.

In this model, Chang *et al.* [4] presented a polylog-energy, polytime algorithm for the problem of one-to-all broadcast. A later paper, by Chang, Dani, Hayes and Pettie [5], gave a sub-polynomial energy, polynomial-time algorithm for the related problem of breadth-first search.

In the present work, we will be concerned with another fundamental problem of graph theory, namely to find large sets of pairwise disjoint edges, or *matchings.* The problem of finding large matchings has been thoroughly studied in a wide variety of computational models dating back more than a century, to König [11]. For a fairly comprehensive review of past results, we recommend Duan and Pettie [8, Introduction].

The main goal of the present work is to present a polylog-energy, polynomial-time distributed algorithm that computes a maximal matching in the network graph. The term maximal here indicates that the matching intersects every edge of the graph, and therefore cannot be augmented without first removing edges. It is well-known that the cardinality of any maximal matching is within a factor two of the largest (or maximum) matching.

THEOREM 1.1. *Let G be any graph on at most n vertices, of maximum degree at most Δ. Then there is a distributed algorithm with the following properties:*

(1) *The algorithm always terminates in $O(\Delta \log n)$ timesteps, at which point each node knows its partner in a matching, M.*
(2) *With high probability, M is a maximal matching and every node used energy at most $20C \log^2 n$.*

Observe that the per-node energy use is polylog(n), which obviously can't be improved by more than a polylog factor. Moreover, the time complexity bound, $O(\Delta \log n)$, is also nearly optimal, when one considers that G could contain a clique of size Δ, in which case, in order for all the nodes in that clique to get even one chance to send a message and have it received by the other nodes in the clique, there must be at least Δ timesteps, since our model does not allow a node the possibility to receive two or more messages in a single round. To put this another way, when Δ is small, a high degree of parallelism is possible, which our algorithm exploits; but, when Δ is large, there exist graphs for which this parallelism is impossible.

1.1 Application: Neighbor Assignment

One possible motivation for finding large matchings, apart from their intrinsic mathematical interest, comes from the desire to **back up data in case of node failures**. Suppose we had a perfect matching (that is, one whose edges contain every node) on the n nodes of our network. Then the matching could be viewed as pairing each node with a neighboring node that could serve as its backup device. This would ensure that each device has a load of one node to back up, and that each node is directly adjacent to its backup device.

Since perfect matchings are not always available, we consider a more general scheme, in which each node is assigned one of its neighbors to be its backup device, but we allow for loads greater than one. Such a function can be visualized as a directed graph, with a directed edge from each node to its backup device. In this case, each node has out-degree 1, and load equal to its in-degree. We would like to minimize the maximum load over all vertices.

In Section 3, we will show that, if one is willing to accept a maximum load that is $O(\log n)$ times the optimum, this problem can be simply reduced to the maximal matching problem. In light of our main result, this means that, if there exists a neighbor assignment with polylog(n) maximum load, then we can find one on a radio network, while using only polylog(n) energy.

1.2 Related Work

Multi-hop radio network models have a long history, going back at least to work in the early 1990's by Bar-Yehuda, Goldreich, Itai [1, 2] among others. The particular model of energy-aware radio computation we are using was introduced by Chang et al. [4].

A recent result by Chatterjee, Gmyr, and Pandurangan [6] considered the closely related problem of Maximal Independent Set in another model, called the "Sleeping model." Although it has some interesting similarities to our work, there are several important differences. Firstly, we note that although matchings of G are nothing more than independent sets on the line graph of G, in distributed computing, we cannot just convert an algorithm designed to run on the line graph of G into an algorithm to run on G. Secondly, we note that the Sleeping model is based on the CONGEST model, and so, when a node is awake, it is allowed to send a different message to each of its neighbors at a unit cost. By contrast, in our model, one node can only send one message in a timestep, and it may collide with messages sent by other nodes.

Moscibroda and Wattenhofer [12] considered the problem of finding a Maximal Independent Set in a radio network. Their work also has some interesting similarities to ours, although they are assuming a unit-disk topology, and listening for messages is free in their model. On the other hand, their algorithm works even when the nodes wake up asynchronously at the start of the algorithm.

2 MATCHINGS

A *matching* is a subset of the edges of a graph G, such that no two of the edges share an endpoint. We say a matching is *maximum* if it has at least as many edges as any other matching for G. We say a matching is *maximal* if it is not contained in a larger matching for G. Equivalently, a matching is maximal if every edge of G shares at least one endpoint with an edge from the matching.

For $\alpha > 1$, we say a matching is α-approximately maximum if its cardinality is at least $1/\alpha$ times the cardinality of a maximum matching. It is an immediate consequence of the definitions that any maximal matching is 2-approximately maximum.

Our matching algorithm can be thought of as a distributed and low-energy version of the following greedy, centralized algorithm. Randomly shuffle the m edges. Then, processing the edges in order, accept each edge that is disjoint from all previous edges. Note that this always results in a maximal matching.

To make this into a distributed algorithm, we make each node, in parallel, try to establish contact with one of its unmatched neighbors to form an edge. Since a node can only receive a message successfully if exactly one of its neighbors is sending, we limit the probability for each node to participate in a given round, by setting a participation rate that is, with high probability, at most the inverse of the maximum degree of the residual graph induced by the unmatched nodes. It turns out that this can be accomplished using a set schedule, where the participation rate is a function of the amount of elapsed time.

The main technical obstacle in the analysis is proving that the maximum degree of the graph decreases according to schedule (or faster). This is achieved by noting that, if not, the first vertex to have its degree exceed the schedule would have to have been failed to be paired by our algorithm, despite going through a long sequence of consecutive rounds in which its chance to be paired was relatively high.

3 NEIGHBOR ASSIGNMENT FUNCTIONS

Motivated by the problem of assigning nodes to backup data from their neighbors in a sensor network, we introduce the following definition. As we shall see later, it is extremely closely connected to the established concept of matching covering number.

Definition 3.1. Given graph $G = (V, E)$, a *neighbor assignment function* (NAF) is a function $f : V \to V$ such that for all $v \in V$, $\{v, f(v)\} \in E$. Equivalently, we may think of this as an oriented subgraph of G, in which each vertex has out-degree 1. The *load* of the assignment is the maximum in-degree of this digraph. Equivalently, load is $\max_{v \in V} |f^{-1}(v)|$. The *minimum NAF load* of G is the minimum load among all NAFs for G.

Note: In the case when G is bipartite, NAFs are also known as "semi-matchings." (See, for example, [7, 9].) However, since we are particularly concerned with the non-bipartite case, we preferred to introduce a different term.

In the context of backing up data, we think of the assigned node $f(v)$ as the node who will store a backup copy of v's data. Our goal for this section is to find a NAF whose load is small. In the energy-aware radio network setting, we also want to ensure that the per-node energy use is small.

There is a close connection between the load of the best NAF for a graph and the minimum number of matchings needed to cover all of its vertices, as established in the following

Definition 3.2. For a graph, G, the *matching cover number* of G, denoted $\mathrm{mc}(G)$, is the minimum integer k, such that there exists a set of k matchings of G, whose union contains every vertex of G.

THEOREM 3.3. *For every graph G, the minimum NAF-load of G equals the matching cover number of G, unless the NAF-load of G equals 1. If the NAF-load of G equals 1, the matching cover number of G can be 1 or 2.*

Song, Wang, and Yuan [14] have given an $O(n^3)$-time centralized algorithm for finding the minimum number of matchings needed to cover a graph. In light of Theorem 3.3, their result implies an $O(n^3)$ algorithm for finding the minimum-load NAF for any graph.

In the distributed and low-energy setting, it is unlikely that we can achieve such an ambitious goal. For instance, a node cannot determine its exact degree without receiving that many messages successfully, which may require linear energy. Instead, we aim for the less ambitious goal of finding a NAF whose maximum load is well within our energy budget. Our next result indicates that such a NAF can be found assuming one exists.

THEOREM 3.4. *Suppose G has a NAF with maximum load L. Then there is a distributed algorithm that, given L, will, with high probability, output a NAF with maximum load $O(L \log n)$. Its per-vertex energy usage is $O(L \operatorname{polylog} n)$.*

REFERENCES

[1] Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. Efficient emulation of single-hop radio network with collision detection on multi-hop radio network with no collision detection. *Distributed Computing*, 5(2):67–71, 1991.
[2] Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *Journal of Computer and System Sciences*, 45(1):104–126, 1992.
[3] Matthew Barnes, Chris Conway, James Mathews, and DK Arvind. Ens: An energy harvesting wireless sensor network platform. In *2010 Fifth International Conference on Systems and Networks Communications*, pages 83–87. IEEE, 2010.
[4] Yi-Jun Chang, Varsha Dani, Thomas P Hayes, Qizheng He, Wenzheng Li, and Seth Pettie. The energy complexity of broadcast. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 95–104, 2018.
[5] Yi-Jun Chang, Varsha Dani, Thomas P Hayes, and Seth Pettie. The energy complexity of BFS in radio networks. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 273–282, 2020.
[6] Soumyottam Chatterjee, Robert Gmyr, and Gopal Pandurangan. Sleeping is efficient: Mis in o (1)-rounds node-averaged awake complexity. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 99–108, 2020.
[7] Andrzej Czygrinow, Michał Hanćkowiak, Edyta Szymańska, and Wojciech Wawrzyniak. On the distributed complexity of the semi-matching problem. *Journal of Computer and System Sciences*, 82(8):1251–1267, 2016.
[8] Ran Duan and Seth Pettie. Linear-time approximation for maximum weight matching. *Journal of the ACM (JACM)*, 61(1):1–23, 2014.
[9] Nicholas JA Harvey, Richard E Ladner, László Lovász, and Tami Tamir. Semi-matchings for bipartite graphs and load balancing. *Journal of Algorithms*, 59(1):53–78, 2006.
[10] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pages 10–pp. IEEE, 2000.
[11] Dénes König. Über graphen und ihre anwendung auf determinantentheorie und mengenlehre. *Mathematische Annalen*, 77(4):453–465, 1916.
[12] Thomas Moscibroda and Roger Wattenhofer. Maximal independent sets in radio networks. In *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 148–157, 2005.
[13] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: Enabling ultra-low power wireless research. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 364–369. IEEE, 2005.
[14] Xiumei Wang, Xiaoxin Song, and Jinjiang Yuan. On matching cover of graphs. *Mathematical Programming*, 147(1):499–518, 2014.