Brief Announcement: Near-Optimal BFS-tree Construction in Radio Networks

Mohsen Ghaffari
MIT, CSAIL
32 Vassar Street
Cambridge, MA 02139
ghaffari@mit.edu

Bernhard Haeupler
Microsoft Research
1065 La Avenida
Mountain View, CA 94043
haeupler@cs.cmu.edu

ABSTRACT

We present the first improved construction of a *Breadth First Search tree* (BFS-tree) in the *radio network model*.

Computing a BFS-tree, or the hop-distance to a source, was one of the first problems solved in the radio network model. Particularly, more than 20 years ago, Bar-Yehuda et al. showed how to compute a BFS-tree in $O(D \log^2 n)$ rounds in any n-node network of diameter D. Since then this BFS-tree algorithm has been used extensively for constructing a substrate over which communications can be coordinated efficiently. However, the $O(D \log^2 n)$ dependence on the diameter has become a running time bottleneck in many of these applications, and no faster construction was found. Recently, trying to circumvent this barrier, approximate variants of BFS-trees were introduced which can be used in a similar manner but be computed faster. Still, the question whether exact BFS-tree could be computed faster remained open.

Here we present a simple randomized distributed algorithm that computes a BFS-tree in $O(D \log n \log\log n + \frac{\log^3 n}{\log\log n})$ rounds, with high probability. This running time is optimal up to an $O(\log\log n)$ factor for most values of D, in particular for all $D \in [\log^2 n, n^{1-\varepsilon}]$ for any constant $\varepsilon > 0$.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems—*Computations on Discrete Structures*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Network Problems*

Keywords

Wireless networks; Radio Networks; Breadth First Search

1. INTRODUCTION

Wireless networks are inherently broadcast networks, yet interference of simultaneous transmissions make the efficient use of the shared medium a hard problem in multi-hop networks. This is particularly true in distributed settings in which the participants are unaware of the network topology. The radio network model [4] was introduced to capture these characteristics and has been intensely studied since then (see [12] for a survey). In this model the network is given by an undirected graph $G = (V, E)$ with $n = |V|$ nodes and diameter D. Nodes communicate in synchronous rounds in which each node can decide to listen or to send a packet. A listening node u receives a packet sent by a neighbor v if and only if v is the only neighbor of u sending. If on the other hand multiple neighbors of u send simultaneously, then the transmissions interfere and form a collision at u. We are concerned with the standard setting in which such collisions cannot be detected, that is, distinguished from no neighbor sending.

It was observed very early on that for many distributed communication problems it is useful to first structure the radio network. In particular, Bar-Yehuda et al. [3] showed how to compute a BFS-tree from a given node in $O(D \log^2 n)$ time. They then used this BFS-tree to efficiently solve several multi-message communication problems on top of it, such as, broadcasting k messages from a source to all nodes in a network. Many subsequent works, such as [5–7, 9, 10], also applied this strategy. Interestingly, many protocols used on top of the BFS-trees have an $O(D \log n)$ dependency on the diameter, making the simple BFS-tree computation the running-time bottleneck of quite sophisticated algorithms. Despite this, the twenty years old $O(D \log^2 n)$ BFS-tree construction of [3] remained the best known. Recently, in an attempt to circumvent this predicament, in [8] the authors introduced a relaxation of BFS-trees, called *pseudo BFS-trees*, and showed that this relaxation is good enough to be used in many applications. The paper [8] also contains a sophisticated recursive algorithm that computes a pseudo BFS-tree in near-optimal time, namely, in $O(D \log \frac{n}{D} + \log^{2+\varepsilon} n)$ rounds, for any constant $\varepsilon > 0$. However, the question whether a BFS-tree or exact node distances to the source could be computed in less than $\Theta(D \log^2 n)$ rounds remained open.

Our Contribution: We give the first improved BFS-tree construction:

Theorem 1.1. *Assume a radio network $G = (V, E)$ with n nodes, diameter D and a designated source node $s \in V$. There is a randomized distributed algorithm that computes a BFS-tree rooted at s in $O(D \log n \log\log n + \frac{\log^3 n}{\log\log n})$ rounds, with high probability.*

PODC'14, July 15–18, 2014, Paris, France.
ACM 978-1-4503-2944-6/14/07.
http://dx.doi.org/10.1145/2611462.2611507.

This round complexity is near-optimal, considering the lower bound of $\Omega(D \log \frac{n}{D} + \log^2 n)$ [1, 11][1]. Particularly, it is within $O(\log \log n)$ factor of the lower bound for $D \in [\log^2 n, n^{1-\varepsilon}]$ for any constant $\varepsilon > 0$.

2. THE DECAY PROTOCOL

We first present the *decay protocol* of [2]. This protocol was initially used for spreading one message from the source to all nodes, in $O(D \log n + \log^2 n)$ rounds. It is also the base for the $O(D \log^2 n)$ round BFS-tree construction of [3].

Decay can be view as a coordinated random exponential back-off strategy. It divides the time into *phases*, each consisting of $\log n + 1$ rounds. In the k^{th} round of each phase, each node that has a message to send transmits it with probability $2^{-(k-1)}$.

Lemma 2.1. *(**Decay's Progress Lemma** [2]) Consider a node v and a phase r. If there is at least one neighbor u of v that has a message to send in phase r, then with probability at least $1/8$, node v receives a message in this phase (although not necessarily from u).*

Proof. Suppose that the number of neighbors of v that have a message to send in phase r is x. The case $x = 1$ is taken care of by the first round of the phase. Suppose $x \geq 2$ and let $k^* = \lceil \log_2 x \rceil$. In round k^* of phase r, the probability that v receives a message from one of its neighbors is at least $x \cdot 2^{-k^*+1}(1 - 2^{-k^*+1})^{x-1} \geq 1/4$. □

The method of [3] for computing a BFS is as follows: divide time into *epochs*, each having $10 \log n$ decay phases. The source sends a message in the round of the first epoch. When a node v receives a message for the first time in epoch r, it records r as its estimate $\tilde{d}_v$ of its distance d_v from the source. Moreover, v sends a message in every epoch $r' \geq r + 1$. From Lemma 2.1 we get that in any epoch a node, with high probability, receives a message if at least one of its neighbors is sending in this epoch. It follows directly that for every node $\tilde{d}_v = d_v$ with high probability.

3. THE ALGORITHM

We explain an algorithm for computing a BFS in networks with diameter at most $d = \frac{100 \log^2 n}{\log^2 \log n}$ using $T = \Theta(\frac{\log^3 n}{\log \log n})$ rounds. A pseudocode is presented in Algorithm 1. Constructing a BFS-tree in a networks with larger diameter can be done by essentially just repeating this algorithm.

We first explain the concept of a *wave*, which computes distance upper bounds with some weak probabilistic guarantees. By pipelining logarithmically many such waves and taking the best/lowest upper bound we obtain the correct distance, with high probability.

3.1 One Wave

Similar to [3] we divide time into epochs. However our epochs contain only $40 \log \log n$ phases of decay each. The wave starts at the source s, with the source sending a message in all rounds of the first epoch. Whenever a node v receives a message for the first time, say in epoch i, v records $\tilde{d}_v = i$ as its *guess about distance* d_v of v from the source, computed in this wave. Furthermore, node v sends the message containing $\tilde{d}_v$ starting with the next epoch and for $\frac{5 \log n}{\log \log n}$ epochs.

[1]We note that this lower bound was presented for the single-message broadcast problem but it also applies to the BFS construction.

Lemma 3.1. *For each node v with distance $d_v \leq \frac{100 \log^2 n}{\log^2 \log n}$ from the source, we have the following three properties: (1) a deterministic guarantee that $\tilde{d}_v \geq d_v$, (2) with probability at least $1 - \frac{1}{\log^2 n}$, we have $\tilde{d}_v = d_v$, (3) with probability at least $1 - \frac{1}{n^2}$, we have $\tilde{d}_v \leq d_v + \frac{\log n}{\log \log n}$.*

Proof. The first property is trivial as the wave proceeds at most one hop in each epoch. For the other properties, first notice that a node v stopping sending messages after $\frac{5 \log n}{\log \log n}$ epochs does not have any effect, with high probability. This is because after $\frac{5 \log n}{\log \log n}$ epochs, which has $5 \log n$ decay phases, any neighbor of u has with high probability received a message.

For the second property, consider a node v and a shortest path P_v from the source s to node v of length d_v. This path has the from $s = w_0, w_1, \ldots w_{d_v} = v$ with length d_v. If in each epoch, the message makes exactly one hop of progress along P_v, we have $\tilde{d}_v = d_v$. Consider one hop w_j to w_{j+1} of this path and suppose that w_j is sending in an epoch. Let us say the j^{th} hop *failed in this epoch* if w_{j+1} does not receive a message (from anyone) in the epoch that w_j is sending. The probability that the j^{th} hops fails (in a given epoch) is at most $(1 - \frac{1}{8})^{40 \log \log n} \leq \frac{1}{\log^5 n}$. Hence, a union bound over all $d_v \leq \frac{100 \log^3 n}{\log \log n}$ hops of path P_v shows that none of them fails—i.e., v will receive a message after d_v epochs—with probability at least

$$1 - \frac{100 \log^2 n}{\log^2 \log n} \cdot \frac{1}{\log^5} \geq 1 - \frac{1}{\log^2 n},$$

thus completing the proof of the second property.

For the third property, note that if we have $\tilde{d}_v \geq d_v + \frac{\log n}{\log \log n}$, it means that we have had at least $k = \frac{\log n}{\log \log n}$ hop failures along some path P_v (note that each hop might fail more than once). These failures have happened during the epochs of this wave, which, counting rather pessimistically, is at most $\frac{100 \log^2 n}{\log^2 \log n} \cdot \frac{\log n}{\log \log n} = O(\log^3 n)$. The probability of having k hop failures is thus as at most

$$\binom{O(\log^3 n)}{k} \left(\frac{1}{\log^5 n}\right)^k \leq \left(\frac{O(\log^3 n)}{\frac{\log n}{\log \log n}} \cdot \frac{1}{\log^5 n}\right)^{\frac{\log n}{\log \log n}} \leq \left(\frac{1}{\log^2 n}\right)^{\frac{\log n}{\log \log n}} = \frac{1}{n^2}.$$

□

3.2 Pipelining Waves

We now explain how to strengthen Lemma 3.1 by having many waves that are pipe-lined. The source starts one new *wave* every $\frac{10 \log n}{\log \log n}$ epochs and repeats this for $\frac{10 \log n}{\log \log n}$ waves. We show in Lemma 3.2 that the waves will always be well-separated and they will not interfere with each other. Each node v takes its final guess d'_v about its distance from the source to be the smallest guess $\tilde{d}_v$ that it has had over all the $\frac{10 \log n}{\log \log n}$ waves. We show in Lemma 3.3 that, with high probability, we have $d'_v = d_v$ for every node v. Each node in the network is done with the transmissions of the last wave

Algorithm 1 BFS run at node u

```
if u = source then
    for i = 1 to 10 log n / log log n do
        TRANSMIT 0 for one epoch
        LISTEN for 10 log^2 n / log log n epochs
else
    active ← 0
    d̃_u ← ∞
                                  ▷ each iteration of this loop is one epoch
    for i = 1 to 200 log^2 n / log^2 log n do
        rec = false
        for j = 1 to 40 log log n do
            for k = 1 to log n + 1 do
                if active > 0 then
                    With probability 2^{-(k-1)}, TRANSMIT d̃_u
                            Otherwise LISTEN for one round
                else
                    LISTEN for one round
                    if received d' then
                        d̃_u ← min{d̃_u, d' + 1}
                        rec ← true           ▷ first time reception
        if rec = true then
                                  ▷ transmit for 5 log n / log log n epochs
            active ← 5 log n / log log n
            rec ← false
        else if active > 0 then
            active ← active − 1
```

at most at the end of epoch $\frac{10\log n}{\log\log n}\cdot\frac{10\log n}{\log\log n}+\frac{100\log^2 n}{\log^2\log n}=\frac{200\log^2 n}{\log^2\log n}$, which means that after at most $T=\Theta(\frac{\log^3 n}{\log\log n})$ rounds, thus proving the round complexity claim.

Lemma 3.2. *For any i, and any two nodes u and v within distance at most $\frac{3\log n}{\log\log n}$ hops from each other, node v is done with its transmissions in the i^{th} wave before the $(i+1)^{th}$ wave arrives at u.*

Proof. The i^{th} wave starts at the source at round $\frac{10(i-1)\log n}{\log\log n}$ and, according to Lemma 3.1, it arrives at a node v after at most $d_v+\frac{\log n}{\log\log n}$ epochs, with high probability. After the wave is received the nodes v will then send for $\frac{5\log n}{\log\log n}$ additional epochs. After that, that is, at epoch

$$End_v^i \le \frac{10(i-1)\log n}{\log\log n}+d_v+\frac{6\log n}{\log\log n},$$

node v will be done with transmissions of wave i. On the other hand, the $(i+1)^{th}$ wave arrives at u no earlier than after $\frac{10i\log n}{\log\log n}+d_u$. That is, in epoch

$$\begin{aligned} Start_u^i &\ge \frac{10i\log n}{\log\log n}+d_v-\frac{3\log n}{\log n} \\ &= \frac{10(i-1)\log n}{\log\log n}+d_v+\frac{7\log n}{\log\log n} > End_v^i. \end{aligned}$$

□

Lemma 3.3. *With high probability, for each node v, we have $d'_v = d_v$ at the end.*

Proof. First note that in each wave, we have the deterministic guarantee that $\tilde{d}_v \ge d_v$. Hence, we have the deterministic guarantee that $d'_v \ge d_v$. We say a wave failed at v if in that wave $\tilde{d}_v > d_v$. If $d'_v > d_v$, it should be the case that all waves failed at v. Given property (2) of Lemma 3.1, the probability of one wave failing at v is at most $\frac{1}{\log^2 n}$. From Lemma 3.2, we know that the waves do not interfere and thus, their failures are independent. Therefore, the probability that $d'_v > d_v$ or equivalently that all waves failed at v is at most $(\frac{1}{\log^2 n})^{\frac{10\log n}{\log\log n}} \le \frac{1}{n^5}$. □

4. REFERENCES

[1] Alon, N., Bar-Noy, A., Linial, N., and Peleg, D. A lower bound for radio broadcast. *Journal of Computer and System Sciences 43*, 2 (1991), 290–298.

[2] Bar-Yehuda, R., Goldreich, O., and Itai, A. 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 (1992), 104–126.

[3] Bar-Yehuda, R., Israeli, A., and Itai, A. Multiple communication in multi-hop radio networks. *SIAM Journal on Computing 22*, 4 (1993), 875–887.

[4] Chlamtac, I., and Kutten., S. On broadcasting in radio networks: Problem analysis and protocol design. *IEEE Transactions on Communications 33*, 12 (1985), 1240–1246.

[5] Chlebus, B., Kowalski, D., and Radzik, T. Many-to-many communication in radio networks. *Algorithmica 54*, 1 (2009), 118–139.

[6] Christersson, M., Gasieniec, L., and Lingas, A. Gossiping with bounded size messages in ad hoc radio networks. In *Proc. Colloquium on Automata, Languages and Programming* (2002), 377–389.

[7] Gasieniec, L., Peleg, D., and Xin, Q. Faster communication in known topology radio networks. In *Proc. ACM Symp. on Principles of Distributed Computing* (2005), 129–137.

[8] Ghaffari, M., and Haeupler, B. Fast structuring of radio networks for multi-message communications. In *The Proceedings of the International Symposium on Distributed Computing (DISC)* (2013).

[9] Ghaffari, M., Haeupler, B., and Khabbazian, M. Randomized broadcast in radio networks with collision detection. In *Proceedings of the ACM Symposium on Principles of Distributed Computing* (2013), PODC, pp. 325–334.

[10] Khabbazian, M., and Kowalski, D. Time-efficient randomized multiple-message broadcast in radio networks. In *Proc. ACM Symp. on Principles of Distributed Computing* (2011), 373–380.

[11] Kushilevitz, E., and Mansour, Y. An $\Omega(D\log(N/D))$ lower bound for broadcast in radio networks. *SIAM Journal on Computing 27*, 3 (1998), 702–712.

[12] Peleg, D. Time-efficient broadcasting in radio networks: A review. In *The Proceedings of The International Conference on Distributed Computing and Internet Technologies* (2007), pp. 1–18.