Radio Network Distributed Algorithms in the Unknown Neighborhood Model

Bilel Derbel, El-Ghazali Talbi

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Radio Network Distributed Algorithms in the Unknown Neighborhood Model

Bilel Derbel — El-Ghazali Talbi

N° 6581

June

Thème COM

Rapport de recherche

Radio Network Distributed Algorithms in the Unknown Neighborhood Model

Bilel Derbel* † , El-Ghazali Talbi‡

Thème COM — Systèmes communicants
Équipes-Projets DOLPHIN

Abstract: The paper deals with radio network distributed algorithms where nodes are not aware of their one hop neighborhood. Given an n-node graph modeling a multihop network of radio devices, we give a $O(\log^2 n)$ time distributed algorithm that computes w.h.p., a constant approximation value of the degree of each node. We also provide a $O(\Delta \log n + \log^2 n)$ time distributed algorithm that computes w.h.p., a constant approximation value of the local maximum degree of each node, where the global maximum degree Δ of the graph is not known. Using our algorithms as a plug-and-play procedure, we show that many existing distributed algorithms requiring the knowledge of Δ to execute efficiently can be run with essentially the same time complexity by using the local maximum degree instead of Δ. In other words, using the local maximum degree is sufficient to break the symmetry in a local and efficient manner. We illustrate this claim by investigating the complexity of some basic problems. First, we investigate the generic problem of simulating any classical message passing algorithm in the radio network model. Then, we study the fundamental edge/node coloring problem in the special case of unit disk graphs. The obtained results show that knowing the local maximum degree allows to coordinate the nodes locally and avoid intereferences in radio networks.

Key-words: Distributed algorithms, time complexity, Radio networks.

 * Laboratoire d'Informatique Fondamentale de Lille (LIFL), Université de Lille 1 (USTL). Supported by the DOLPHIN inria project team.
 † corresponding author : bilel.derbel@lifl.fr
 ‡ Laboratoire d'Informatique Fondamentale de Lille (LIFL), Université de Lille 1 (USTL). Supported by the DOLPHIN inria project team.

Radio Network Distributed Algorithms in the Unknown Neighborhood Model

Résumé : The paper deals with radio network distributed algorithms where nodes are not aware of their one hop neighborhood. Given an n-node graph modeling a multihop network of radio devices, we give a $O(\log^2 n)$ time distributed algorithm that computes w.h.p., a constant approximation value of the degree of each node. We also provide a $O(\Delta \log n + \log^2 n)$ time distributed algorithm that computes w.h.p., a constant approximation value of the local maximum degree of each node, where the global maximum degree Δ of the graph is not known. Using our algorithms as a plug-and-play procedure, we show that many existing distributed algorithms requiring the knowledge of Δ to execute efficiently can be run with essentially the same time complexity by using the local maximum degree instead of Δ. In other words, using the local maximum degree is sufficient to break the symmetry in a local and efficient manner. We illustrate this claim by investigating the complexity of some basic problems. First, we investigate the generic problem of simulating any classical message passing algorithm in the radio network model. Then, we study the fundamental edge/node coloring problem in the special case of unit disk graphs. The obtained results show that knowing the local maximum degree allows to coordinate the nodes locally and avoid intereferences in radio networks.

Mots-clés : Distributed algorithms, time complexity, Radio networks.

1 Introduction

1.1 Motivation and goals

We consider a multi-hop network of radio devices modeled by a graph. The nodes can communicate with their neighbors by broadcasting messages. We assume that simultaneous communications are subject to interferences. We assume that nodes are not yet organized in any specific manner, that is we consider a newly deployed network where no specific communication structure is available to nodes. Generally speaking, we want to let the nodes in the network self-organise in some accurate manner in order to be able to do more sophisticated computations. In this context, we focus on what knowledge about the network is required in order to organize the network distributively. In other words, how can we organise the network efficiently using only a local information. More specifically, our aim is to study the impact of two parameters, the degree of each node and the maximum degree of the graph, on designing time-efficient distributed algorithms for solving basic tasks in a radio network model. These two parameters are in fact crucial ingredients when computing many fundamental distributed structures, e.g., node/edge coloring [MW05, BKM+06], matching [BBK+04], single-hop emulation [BYGI89], broadcasting and gossiping [GPX05, ANLP89, BYII89], etc. Many past works passed away the fact that these parameters could not be available to nodes in a newly deployed network. The key challenge of this paper is to provide efficient algorithms working under the assumption that those parameters are not available.

Our interest in node degrees stems from the fact that the information available about node degrees is often used to efficiently break the symmetry in a local manner. For instance, to avoid the interferences caused by simultaneous transmissions, a common technique consists in choosing a sending probability for each node at each time slot depending on that information. In addition, the time complexity of solving many basic distributed tasks is tightly related to the degrees in the network in the sens that related lower and upper bounds are often functions of those degrees. To illustrate this claim, consider the classical radio network model where a node can hear a message from one neighbor in a given time slot if and only if no other neighbors are being sending in that time slot. Assume also that a node can not detect collisions that is it can not distinguish the scenario where no neighbor is sending from the scenario where more than one neighbor are sending. It is obvious that if a node v has to receive one message from each of its neighbors, then its degree is a lower bound on the number of time slots that are needed. Assuming that v's neighbors know v's degree, if each neighbor tries to send with a probability inversely proportional to v's degree, then we can show that the latter lower bound is almost tight. Considering the more general case where *each* node must send a message to all its neighbors, it appears that the maximum degree is an important parameter to globally coordinate the sending probabilities of nodes to attain tight performances. This example shows that even the most primitive tasks (e.g., broadcast a message to neighbors) can be affected by the knowledge of node degrees. Generally speaking, tuning the sending probabilities depending on that knowledge is often a key idea in order to coordinate node transmissions distributively and to make local decisions efficiently.

Unfortunately, the nodes could even not be aware of their own degrees. This observation leads to the fact that efficient algorithms working under the assumption that no knowledge about node degrees is available are more likely to be applied in a real setting. In this paper, we will show how to overcome the absence of any such knowledge. From a more theoretical point of view, we want to understand how such a knowledge can affect our way of thinking and designing efficient distributed algorithms. The motivation behind this approach is a better understanding of the *locality nature* of radio network distributed algorithms, that is how to efficiently solve a distributed problem and make local decisions using only a local information. Such a theoretical understanding is in fact very precious to come out with new efficient algorithms in realistic applications.

1.2   Model

We consider a classical graph based radio network model. More precisely, the network is modeled by an n-node graph $G = (V, E)$. The nodes represent a set of radio devices. There is an edge between two nodes if the two nodes are in the transmission range of one another. Nodes can communicate by sending and receiving messages. For the sake of analysis, we suppose that the nodes have access to a global clock generating discrete pulses (slots), i.e., the network is synchronized (though, this assumption is not crucial for our algorithms). We assume that nodes wake up synchronously, i.e., at pulse 0 all nodes are awaken. At each pulse, a node can transmit, i.e., send a message, or stay silent. A node receives a message in a given pulse if and only if it is silent then and precisely one of its neighbors transmits. No collision detection mechanism is available to nodes in any way. Unless stated explicitly, nodes do not have any identifiers. In addition, nodes do not have any knowledge about the current number of nodes in their neighborhood. Nevertheless, we assume that nodes have an estimate $\widetilde{n}$ on the number of nodes n in the network. In practice, our assumptions can be interpreted as following: nodes know that they may have between 1 and $\widetilde{n}$ neighbors, but they do not know how many. We remark that since the running time of our algorithms depends polylogarithmically on $\widetilde{n}$, a rough estimate $\widetilde{n}$ (e.g., $\widetilde{n} = n^{O(1)}$) will affect the time complexity by only a small factor. For the sake of clarity, our algorithms are described for $\widetilde{n} = n$.

For every node v, we denote $\mathcal{N}(v)$ the set of v's neighbors and $d_v = |\mathcal{N}(v)|$ the degree of v. We denote $\mathcal{N}^+(v) = \mathcal{N}(v) \cup \{v\}$. The maximum degree of the network is formally defined as $\Delta = \max\{d_v \mid v \in V\}$. The local maximum degree of a node v is formally defined as $\Delta_v = \max\{d_u \mid u \in \mathcal{N}^+(v)\}$.

We say that an event occurs *with high probability* (w.h.p., for short) if the probability that the event occurs is at least $1 - 1/n^\alpha$ for a positive constant α which can be made arbitrarily large by setting the corresponding parameters to large enough values.

1.3   Methodology and results

The first result of the paper is a $O(\log^2 n)$ time distributed algorithm that computes w.h.p., a constant approximation value of the degree of each node. The idea of the algorithm is based on the claim that if the nodes send a message with the same probability p then the probability that a given node v hears a message from its neighbors is maximized for $p = 1/d_v$. Thus, to decide on an estimate value of its degree, a node v changes its sending probability periodically and tries to detect the sending probability for which it hears the most often messages from its neighbors.

The second result of the paper is a $O(\Delta \log n + \log^2 n)$ time distributed algorithm that computes w.h.p., a constant approximation value of the local maximum degree $\widetilde{\Delta_v}$ of each node v. The idea of the algorithm is based on the claim that if all the nodes send with the same probability p then w.h.p., for $p = 1/d_v$, a node v hears a message from each neighbor during a period of $O(d_v \log n)$ time. Thus, by combining this observation with our first algorithm, we are able to coordinate the nodes (by periodically varying their sending probabilities) so that each node can efficiently communicate its approximate degree to its neighbors.

The third result of the paper is more general, in the sens that it allows us to get rid of the knowledge of Δ in many distributed applications. It is based on the latter two results. We illustrate this for three fundamental problems: (i) the SRS (Single Round Simulation) problem which consists in simulating the rounds of a synchronous point-to-point message passing algorithm into the radio network model, (ii) the classical node coloring problem which consists in coloring the nodes such that every two neighbors have distinct colors (ii) a variant of the maximum matching problem which consists in computing a set of non adjacent edges, i.e., strong (or distance-2) matching[1]. For these three problems, the best existing algorithms use the knowledge of Δ to

---

[1]This problem is tightly related to the strong edge coloring problem which consists in coloring the edges in such away that two edges at distance 1 or 2 have distinct colors. The edge coloring problem is itself tightly related to the channel assignment problem.

break the symmetry and execute efficiently. Based on our algorithms and using $\widetilde{\Delta_v}$ instead of Δ locally for each node v, we derive new algorithms with essentially[2] the same time complexity than the best previous algorithms. More precisely, we obtain respectively $O(\Delta^2 \log n + \log^2 n)$[3], $O(\Delta \log n + \log^2 n)$, $O(\log^3 n)$ time algorithms (w.h.p.). The key argument of our analysis relies on the observation that if the sending probability is set to be $1/\Delta_v$ for each node v, then w.h.p, a node will succeed sending a message to its neighbors without experiencing a collision within $O(\Delta_v \log n)$ time period.

From a theoretical point of view, the benefits of our results are twofold. First, they show that the knowledge of the degree of nodes affects the running time by only a polylogarithmic factor. Second, they show that the knowledge of Δ does not deeply affect the computations since in many distributed tasks it can be replaced by the local maximum degree. From a more practical point of view, we provide algorithms working in a very primitive computation model since we even do not assume that nodes are aware of their surrounding neighborhoods. We believe that such algorithms could be very useful in real life radio and wireless applications.

1.4 Outline

In Section 2, we describe and analyze algorithm INITNETWORK for computing an estimate value of the degree of each node. In Section 3, we describe and analyze algorithm LOCALMAXDEGREE for computing an estimate value of the local maximum degree of each node. In Section 4, we apply the latter algorithms for the SRS, coloring and matching problems. Finally, in Section 5, we conclude the paper and raise some open problems.

2 An algorithm for computing the degree of nodes

2.1 Description of the algorithm

Input: An n-node graph G. ($k = \Theta(\log n)$, and $\widetilde{\beta} \simeq 0.09$).
Output: $\widetilde{d_v}$: an approximate value of d_v.

for $i = 1$ **to** $\log n$ **do**
$\quad c_i := 0;\ p_i := 1/2^i;$
$\quad$ **for** $j = 1$ **to** k **do**
$\quad\quad v$ sends a ping message with probability p_i;
$\quad\quad$ **if** *v hears a ping message* **then** $c_i := c_i + 1$;
$\quad$ **if** $c_i \geqslant \widetilde{\beta} k$ **then** $\widetilde{d_v} := 2^i;$

Figure 1: Algorithm INITNETWORK: code for a node v

Algorithm INITNETWORK works in $\log n$ phases. At each phase, a node sends a message for $k = \Theta(\log n)$ rounds with a decreasing probability $p_i = p_{i-1}/2$. At each phase, each node counts the number of times it hears a message. If this number exceeds a threshold $\widetilde{\beta} k$ then the node decides that its degree must be order of $1/p_i$. As we will show in our analysis, the main reason that makes the algorithm work is that for every node v if all its neighbors are sending with the same probability $1/d_v$, then the expected number of times v hears a message is order of k. Thus, by varying geometrically the sending probability at each phase from $1/2$ to $1/n$, each node is likely to experience a phase where it hears about $\Theta(k)$ messages. Thus, each node can decide on an approximate value of its degree.

---

[2]Here, "essentially" means "up to a polylogarithmic factor"
[3]As we will precise in Section 4, many improved bounds can be derived using our SRS algorithm.

2.2   Analysis

First, we will extensively use the following helpful inequalities in our analysis:

Fact 1 $\forall x \geqslant 1$ *and* $|t| \leqslant x$, *we have*

$$e^t \left(1 - \frac{t^2}{x} \right) \leqslant \left(1 + \frac{t}{x} \right)^x \leqslant e^t$$

In addition, we will use the following well known inequalities to bound the probability that the sum of random Bernoulli trials deviates from its expectation:

Fact 2 (Chernoff and Hoeffding's Inequalities [MU05, MR95, Hoe63]) *Let* $X_1, \cdots, X_k$ *be* k *independent Bernoulli trials,* $\mathbb{P}(X_i = 1) = p = 1 - \mathbb{P}(X_i = 0)$. *Let* $X = \sum_{i=1}^k X_i$, *and* $\mathbb{E}(X) = \mu$.

The Chernoff's Inequality tells us that for any ϵ *verifying* $0 < \epsilon < 1$, *it holds that*

$$\mathbb{P}(X < (1 - \epsilon)\mu) < e^{-\mu\epsilon^2/2}$$

The Hoeffding's Inequality tells us that for any $\epsilon > 0$, *it holds that*

$$\mathbb{P}(X - \mu \geqslant k\epsilon) \leqslant e^{-2k\epsilon^2}$$

Let us denote $c_i(v)$ the value of the counter c_i of node v at the end of a given phase i. Then,

Lemma 1 *Given a node* $v \in V$, *we have* $\mathbb{E}(c_i(v)) = k \cdot d_v \cdot p_i \cdot (1 - p_i)^{d_v}$.

Proof. Let $ping(v, i, j)$ the event "node v hears a ping at a step j of phase i". For node v to hear a message, it must stay silent and exactly one node in $\mathcal{N}(v)$ must send a message. Since each node decides (independently) with probability p_i, we get:

$$
\begin{aligned}
\mathbb{P}(ping(v, i, j)) &= \sum_{u \in \mathcal{N}(v)} (1 - p_i) \cdot p_i (1 - p_i)^{d_v - 1} \\
&= d_v p_i (1 - p_i)^{d_v}
\end{aligned}
$$

Since the sending probabilities are independent during the k trials, the lemma holds. ∎

Lemma 2 *There exist positive constants* $\nu, \beta, \epsilon > 0$ *such that the following holds:*

1. *For any node* v *and a phase* i *such that* $i - 1 \leqslant \log d_v \leqslant i$, *we have* $\mathbb{E}(c_i(v)) \geqslant \beta \cdot k$.

2. *For any node* v, *if* $i \notin [\log(d_v) - 2, \log(d_v) + 4]$ *then we have* $\mathbb{E}(c_i(v)) \leqslant ((1 - \epsilon)\beta - \nu) \cdot k$.

Proof. By a simple verification, and for $i - 1 \leqslant d_v \leqslant i$ we have $\mathbb{E}(c_i(v)) \geqslant k/(4e)$. Thus, for $\beta = 1/(4e) \simeq 0.091$, the first property of the lemma holds.

Given a node v, let $f(i) = \mathbb{E}(c_i(v))$. Then, we first show that $f(i)$ is maximized for $i = \log(d_v + 1)$, i.e., $d_v = 2^i - 1$. In fact,

$$
\begin{aligned}
f'(i) &= k d_v \left(-\frac{\ln(2) \cdot 2^i}{2^{2i}} \left(1 - \frac{1}{2^i} \right)^{d_v} + \frac{d_v \ln(2) \cdot 2^i}{2^i \cdot 2^{2i}} \left(1 - \frac{1}{2^i} \right)^{d_v - 1} \right) \\
&= \frac{\ln(2) \cdot 2^i \cdot k d_v}{2^{2i}} \left(1 - \frac{1}{2^i} \right)^{d_v - 1} \left(\frac{d_v + 1}{2^i} - 1 \right) \\
&= \frac{\ln(2) \cdot k d_v}{2^i} \left(1 - \frac{1}{2^i} \right)^{d_v - 1} \left(\frac{d_v + 1}{2^i} - 1 \right)
\end{aligned}
$$

Hence, $f'(i) \geqslant 0 \iff i \leqslant \log(d_v + 1)$.

Now, a simple verification shows that for $i = \log(d_v) - 4$ we have $\mathbb{E}(c_i(v)) \leqslant 4 \cdot e^{-4} \cdot k \simeq 0.073 \cdot k$ and for $i = \log(d_v) + 4$ we have $\mathbb{E}(c_i(v)) \leqslant e^{-1/16}/16 \cdot k \simeq 0.058 \cdot k$.

Since $f(i) = \mathbb{E}(c_i(v))$ increases up to $\log(d_v + 1)$ and then decreases, and $\log(d_v) \leqslant \log(d_v + 1) \leqslant \log(d_v) + 1$, the second property of the lemma holds for $\nu \simeq 0.01$ and $\epsilon = 0.001$.

∎

We say that a node decides on the value of $\widetilde{d}_v$ in a given phase i of the algorithm if the condition $c_i \geqslant \widetilde{\beta} k$ is true. Note that $\widetilde{\beta}$ in algorithm INITNETWORK is set to be $(1 - \epsilon)\beta$ where ϵ and β are the constants from the previous lemma.

Lemma 3 *Given a node v, v decides on the value of $\widetilde{d}_v$ during at least one phase i verifying $i \in [\log(d_v) - 2, \log(d_v) + 4]$, with probability at least $1 - 1/n^{\Theta(1)}$.*

Proof. For such a phase i, we have $d_v = \Theta(1/p_i)$. Thus, the Chernoff's inequality tells us

$$
\begin{aligned}
\mathbb{P}\left(c_i(v) \leqslant (1 - \epsilon)\mathbb{E}(c_i(v))\right) &\leqslant \exp\left(-\frac{\epsilon^2}{2} \cdot k \cdot d_v \cdot p_i \cdot (1 - p_i)^{d_v}\right) \\
&\leqslant \exp\left(-\frac{\epsilon^2}{2} \cdot k \cdot d_v \cdot p_i \cdot \exp(-p_i d_v) \cdot \left(1 - \frac{(p_i d_v)^2}{d_v}\right)\right) \\
&\leqslant \exp\left(-\Theta(\epsilon^2 k)\right)
\end{aligned}
$$

Thus for $k \geqslant \log^{\Theta(1)}(n)/\epsilon^2$, we have $\mathbb{P}\left(c_i(v) \leqslant (1 - \epsilon)\mathbb{E}(c_i(v))\right) \leqslant 1/n^{\Theta(1)}$.

Now we take $i - 1 \leqslant \log(d_v) \leqslant i$ (Note that there always exists such a phase i). Thus, we have $\mathbb{E}(c_i(v)) \geqslant \beta k$. Thus $\mathbb{P}\left(c_i(v) \leqslant (1 - \epsilon)\beta k\right) \leqslant 1/n^{\Theta(1)}$. To resume, for phase i such that $i - 1 \leqslant \log(d_v) \leqslant i$ we have:

$$
\mathbb{P}\left(c_i(v) \geqslant (1 - \epsilon)\beta k\right) \geqslant 1 - 1/n^{\Theta(1)}
$$

Thus, the lemma holds.

∎

Now, we have to prove that w.h.p., a node v cannot experience a phase not in the range of $\log d_v$ in which it hears about $\Theta(k)$ messages. This is the purpose of the next lemma.

Lemma 4 *Given a node v and a phase i verifying $i \notin [\log(d_v) - 2, \log(d_v) + 4]$, v does not decide on the value of $\widetilde{d}_v$ at phase i with probability at least $1 - 1/n^{\Theta(1)}$.*

Proof. Take a phase i verifying the condition of the lemma. Hence, by a previous lemma $\mathbb{E}(c_i(v)) \leqslant ((1 - \epsilon)\beta - \nu)k$. Thus, by noting that $(1 - \epsilon)\beta - \mathbb{E}(c_i(v))/k \geqslant \nu > 0$ and using the Hoeffding's inequality, we have

$$
\begin{aligned}
\mathbb{P}(c_i(v) \geqslant (1 - \epsilon)\beta k) &= \mathbb{P}\left(c_i(v) - \mathbb{E}(c_i(v)) \geqslant \left((1 - \epsilon)\beta - \frac{\mathbb{E}(c_i(v))}{k}\right)k\right) \\
&\leqslant \exp\left(-2k\left((1 - \epsilon)\beta - \frac{\mathbb{E}(c_i(v))}{k}\right)^2\right) \\
&\leqslant \exp\left(-2k\nu^2\right)
\end{aligned}
$$

Recall that ν is a constant. Thus for $k \geqslant \dfrac{\ln(n^{\Theta(1)})}{2\nu^2} = \Theta(\log n)$, the lemma holds.

∎

Theorem 1 *In $O(\log^2 n)$ time and with high probability, algorithm INITNETWORK outputs a constant approximation of the degree of any node.*

Proof. The running time is clear from the description of the algorithm. The expected message complexity is too. Using the two previous lemmas, a node v decides *only* at a phase i verifying $i \in [\log(d_v) - 2, \log(d_v) + 4]$ with probability at least $(1 - 1/n^{\Theta(1)})^k$. Thus all nodes decide on such a phase with probability at least :

$$\prod_{v \in V} \left(1 - \frac{1}{n^{\Theta(1)}}\right)^k = \left(1 - \frac{1}{n^{\Theta(1)}}\right)^{k \cdot n}$$

$$\geqslant 1 - \frac{1}{n^{\Theta(1)}}$$

If a node v decides on a phase i such that $i \in [\log(d_v) - 2, \log(d_v) + 4]$ we have that $\widetilde{d_v} = \Theta(d_v)$ which terminates the proof. ∎

3   An algorithm for computing the local maximum degree

3.1   Description of the algorithm

Input: An n-node graph G and a parameter k.
Output: $\widetilde{\Delta_v}$: an approximate value of Δ_v.

run algorithm INITNETWORK and compute $\widetilde{d_v}$;
$\widetilde{\Delta_v} := \widetilde{d_v}$; $i := 0$;
while $i \leqslant \Theta(\log(\widetilde{d_v}))$ **do**
    $i := i + 1$; $p_i := 1/2^i$;
    for $j = 1$ ***to*** k/p_i **do**
        v sends message "$\widetilde{d_v}$" with probability p_i;
        if *v hears a message "$\widetilde{d_w}$" from node w* **then**
            $\widetilde{\Delta_v} := \max \left\{ \widetilde{\Delta_v}, \widetilde{d_w} \right\}$;

Figure 2: Algorithm LOCALMAXDEGREE: code for a node v

The idea of algorithm LOCALMAXDEGREE depicted in Fig. 2 is to coordinate the sending probabilities in a local way so that each node can send the estimate value of its degree to all neighbors within a short period of time. The crucial observation is that given a node v, if all its neighbors are sending with probability $1/d_v$, then v will receive a message from each neighbor with high probability within $O(d_v \log n)$ time. The main difficulty is that a neighbor u of v does not know the degree of v. Thus, neighbor u can not a priori set its sending probability to the required value. Moreover, node u may have other neighbors and thus it is also required to change its sending probability according to those neighbors. The key idea of our algorithm is to increase geometrically the sending probability p_i each $\Theta((1/p_i) \log n)$ time phase i beginning from $p_i = 1/2$ until reaching $p_i = \Theta(1/d_v)$. Thus, at a phase i, nodes with degree order of p_i will hear, w.h.p., a message from neighbors having a larger degree.

3.2   Analysis

The following theorem (and its proof) formalizes the intuition given in the previous paragraph.

Theorem 2 *In $O(\Delta \log n + \log^2 n)$ time and with high probability, algorithm LOCALMAXDEGREE outputs a constant approximation of the local maximum degree of any node.*

Proof. Let us consider a phase i and a node v verifying $d_v = \Theta(1/p_i)$. From the initialization step the condition of the while loop is still true for all nodes having a degree at least order of $\Theta(1/p_i)$.

Let u a node having a degree at least order of $\Theta(1/p_i)$ and v a neighbor of u verifying $d_v = \Theta(1/p_i)$. Assuming that algorithm INITNETWORK outputs the estimate value of the degree of each node, we denote "$(u \longrightarrow v)^{(i)}$" the event "node v hears a message from neighbor u at a given iteration of the for loop" and "$(u \longrightarrow v)^{(*,i)}$" the event "node v hears a message from neighbor u during the for loop". Thus, we have that:

$$
\begin{aligned}
\mathbb{P}\left((u \longrightarrow v)^{(i)}\right) &\geqslant p_i (1 - p_i)^{d_v} \\
&\geqslant p_i \cdot e^{-p_i d_v} \cdot \left(1 - d_v p_i^2\right) \\
&\geqslant \Theta(\frac{1}{d_v}) \cdot e^{-\Theta(1)} \cdot \left(1 - \Theta(\frac{1}{d_v})\right) \\
&\geqslant \Theta(\frac{1}{d_v})
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\mathbb{P}\left(\neg (u \longrightarrow v)^{(*,i)}\right) &\leqslant \left(1 - \Theta\left(\frac{1}{d_v}\right)\right)^{\frac{k}{p_i}} \\
&\leqslant \left(1 - \Theta\left(\frac{1}{d_v}\right)\right)^{\Theta(kd_v)} \\
&\leqslant e^{-\Theta(k)}
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\mathbb{P}\left(\neg (u \longrightarrow v)^{(*,i)}, (u,v) \in E, d_v = \Theta(1/p_i), d_u > \Theta(1/p_i)\right) &\leqslant \sum_{d_v=\Theta(1/p_i), d_u>\Theta(1/p_i)} \mathbb{P}\left(\neg (u \longrightarrow v)^{(*,i)}\right) \\
&\leqslant n \cdot \Theta(\frac{1}{p_i}) \cdot e^{-\Theta(k)} \\
&\leqslant \frac{1}{n^{\Theta(1)}}
\end{aligned}
$$

The last inequality holds for $k = \Theta(\log n)$.

Thus, with probability at least $1 - 1/n^{\Theta(1)}$, every node with degree order of $1/p_i$ receives a message from each of its neighbors having a larger degree. Since the value $\widetilde{d_u}$ is w.h.p. a constant approximation of u's degree, a simple verification shows that with probability at least $1 - 1/n^{\Theta(1)}$ all nodes with a degree order of $1/p_i$ succeeds computing their local maximum degree. More formally, let $\Delta_v^{(i)}$ denote the value computed by a node v at the end of a phase i. we have:

$$
\mathbb{P}\left((\forall v \in V, d_v = \Theta(1/p_i)), \Delta_v^{(i)} = \Theta\left(\max\left\{d_u \mid u \in \mathcal{N}^+(v)\right\}\right)\right) \geqslant 1 - \frac{1}{n^{\Theta(1)}}
$$

This allows us to easily conclude that with probability at least $1 - 1/n^{\Theta(1)}$ every node computes an approximate value of its local maximum degree.

As for the time complexity, we have $\widetilde{d_v} \leqslant \Theta(\Delta)$ for every node in the graph. Thus, the time complexity is at most:

$$
\begin{aligned}
\sum_{i=1}^{\Theta(\log \Delta)} \frac{k}{p_i} &= k \cdot \sum_{i=1}^{\Theta(\log \Delta)} 2^i \\
&= O(k \cdot 2^{\log \Delta}) \\
&= O(\Delta \log n)
\end{aligned}
$$

The last inequality holds for $k = \Theta(\log n)$. Since the time complexity of algorithm INITNETWORK is $O(\log^2 n)$ the theorem is proved. ∎

4   Applications

4.1   On simulating the message passing model

The first application of the previous algorithms is on simulating the standard synchronous point-to-point message passing model on the graph-based radio model. This problem was first studied in [ANLP89]. Paraphrasing the introduction in [ANLP89], that study was motivated by the fact that whenever a type of communication mode emerges, new algorithms have to be developed for it for all standard network operations. Thus, simulation procedures could help to convert algorithms designed for networks with the same topology but different means of communication to algorithms for the new communication mode. In particular, since designing algorithms for radio networks from scratch turns to be a hard task, the simulation of algorithms for standard message-passing systems may prove to be a plausible approach. This motivation is still of interest since many basic distributed tasks in radio networks remain unfortunately without efficient solutions compared to results that the research community have gained concerning the message passing models.

The work in [ANLP89] concentrated on round-by-round simulations where a separate phase of radio transmission is dedicated to simulate each single round of the original algorithm. More precisely, a general primitive called single-round simulation (SRS), serving as a building block in such simulations is provided. The role of this primitive is to ensure that every message passed by the original algorithm during the simulated round will be transmitted (and received) during the simulating phase. We refer to this as the SRS problem. For the *general* message passing model where a node can send a different message to all its neighbors at each pulse, a randomized distributed SRS algorithm with running time $O(\Delta^2 \log n)$[4] was given. For the so-called *uniform* message passing model where a node sends the same message to all its neighbors at each pulse, the authors gave a randomized distributed SRS algorithm with $O(\Delta \log n)$ running time[5].

The SRS algorithms in [ANLP89] use explicitly the knowledge of Δ in order to schedule the sending probability of each node. Thus, these algorithms are impracticable in the unknown neighborhood model. By using our algorithms in order to compute the approximate local maximum degree of each node, and by locally using that estimate value instead of Δ, we can prove the following:

Theorem 3 *In the unknown neighborhood model, the nodes can be initialized in $O(\Delta \log n + \log^2 n)$ time period so that the SRS problem can be solved with high probability for the uniform (resp. general) message passing passing model within $O(\Delta \log n)$ time (resp. $O(\Delta^2 \log n)$ time) on any n-node graph.*

Proof. First we consider a uniform point-to-point message passing algorithm $\mathcal{A}$. For now let us assume that d_v and Δ_v are known for each node v. Consider a round of $\mathcal{A}$. To solve the SRS problem: each node v must send a message M_v to all its neighbors during that round (the message must be received by each neighbor). Consider the following SRS algorithm: each node v sends with probability $1/\Delta_v$ the message M_v for r_v rounds (r_v is to be fixed later on depending on d_v and Δ_v).

Given a node v, let $(u \longrightarrow v)$ denote the event "v hears the message M_u in a single transmission round". Let $(u \longrightarrow v)^*$ denote the event "v hears the message M_u during the r_v transmission rounds". Thus, we have:

$$\mathbb{P}\left((u \longrightarrow v)\right) = \frac{1}{\Delta_u} \cdot \prod_{w \in \mathcal{N}^+(v) \setminus \{u\}} \left(1 - \frac{1}{\Delta_w}\right)$$

---

[4] The authors also showed that $\Omega(\Delta^2)$ is a lower bound.
[5] The authors also showed that $\Omega(\Delta \log(\Delta))$ is a lower bound.

Using the fact that $\forall w \in \mathcal{N}^+(v), \Delta_w \geqslant d_v$, we obtain

$$\mathbb{P}\left((u \longrightarrow v)\right) \geqslant \frac{1}{\Delta_u} \cdot \left(1 - \frac{1}{d_v}\right)^{d_v} \geqslant \Theta\left(\frac{1}{\Delta_u}\right)$$

Thus,

$$\mathbb{P}\left(\neg(u \longrightarrow v)^*\right) \leqslant \left(1 - \frac{1}{\Theta(\Delta_u)}\right)^{r_u} \leqslant \exp\left(-\Theta\left(\frac{r_u}{\Delta_u}\right)\right)$$

Hence, the probability that node u fails sending M_u to at least one neighbor v satisfies

$$\begin{aligned}
\mathbb{P}\left(\neg(u \longrightarrow v)^*, v \in \mathcal{N}(u)\right) &\leqslant \sum_{v \in \mathcal{N}(u)} \mathbb{P}\left(\neg(u \longrightarrow v)^*\right) \\
&\leqslant d_u \cdot \exp\left(-\Theta\left(\frac{r_u}{\Delta_u}\right)\right)
\end{aligned}$$

Thus by choosing $r_u = \Theta\left(\Delta_u \cdot \log(d_u \cdot n)\right)$, node u succeeds sending M_u to all its neighbors with probability at least $1 - 1/n^{\Theta(1)}$. Thus, the SRS algorithm succeeds for all nodes with probability at least $(1 - 1/n^{\Theta(1)})^n \geqslant 1 - 1/n^{\Theta(1)}$. (Note that the constant hidden in the Θ notation are to be tuned carefully to obtain the high probability property). Thus, w.h.p., every node u succeeds transmitting within $O(\Delta_u \log n)$ time. We remark that the SRS algorithm has the nice property that sparse nodes are not penalized by dense regions.

To conclude, remark that algorithm LOCALMAXDEGREE computes an approximation of the degree and the local maximum degree *with high probability*. Thus, by taking into account the success probability of algorithm LOCALMAXDEGREE, and by tuning the constants hidden in the Θ notation, it is easy to conclude that with high probability the SRS algorithm is successful for all nodes.

Now, Let us consider the *general* point-to-point message passing model. In other words, we consider the case where at each round of the original algorithm, a node v sends a different message to each neighbor (at most d_v messages each round). Thus, one can see that the previous SRS algorithm can be easily extended to this general case. More precisely, each node v applies the previous SRS procedure each $O(\Delta_v \log n)$ time period for each of the d_v messages to be delivered. By the same arguments it is not difficult to prove that the messages will be received within $O(d_v \cdot \Delta_v \log n)$. However, this is not sufficient to guarantee the correctness of the SRS algorithm. In fact, consider a message $M_{u,v}$ that is sent in the original message passing algorithm from node u to node v. Suppose that using the above SRS algorithm node v hears the message $M_{u,v}$ at a given period. It may be possible that another neighbor w of u also hears the same message $M_{u,v}$. That is a node may hear a message which was not destinated to him.

The problem now is as following: *In the case a node hears a message, how to decide whether the message was originally sent to him or not ?* We remark that if we make the assumptions that each node knows the identifiers of its neighbors then the question is trivial since we can concatenate the identifier of each neighbor to the message to be sent. However in the model considered in our paper, we do not make any such assumption. Thus, we make each node learn the identifiers of its neighborhood by his own. First, we recall that nodes have no identifiers in our model. We overcome this assumption by making each node select randomly an identifier from the set $\left\{1, \cdots, n^{O(1)}\right\}$. Hence, it is easy to show that with high probability each node will have a unique identifier. Now, each node must communicate its identifier to its neighbors. Using the same SRS procedure this can be done in $O(\Delta \log n)$ time with high probability. Thus, w.h.p., in $O(\Delta \log n)$ time, each node can learn the identifiers of its neighbors. By concatenating the identifier of each reception neighbors to the original messages, we can conclude that the above mentioned SRS procedure succeeds with high probability and the time bound of the theorem is straightforward. ∎

With regards to the lower bounds given in [ANLP89], the previous theorem shows that our algorithms allow to provide tight bounds (up to a log factor) for the SRS problem. In addition, one can derive many interesting corollaries. For instance, it is not difficult to prove the following:

Corollary 1 *Given a uniform (resp. general) point-to-point message passing algorithm $\mathcal{A}$ running on an n-node graph G in τ time ($\tau \leqslant n^{O(1)}$), there exists a distributed algorithm that simulates algorithm $\mathcal{A}$ in the unknown neighborhood radio network model with high probability in $O(\Delta \log n \cdot \tau + \log^2 n)$ time (resp. $O(\Delta^2 \log n \cdot \tau + \log^2 n)$ time).*

Corollary 2 *Consider a point-to-point message passing algorithm $\mathcal{A}$ running on an n-node graph G in τ time ($\tau \leqslant n^{O(1)}$). Assume that at each round of algorithm $\mathcal{A}$, each node sends the same message to a bounded number of neighbors and another message to the remaining neighbors. Then, there exists a distributed algorithm that simulates algorithm $\mathcal{A}$ in the unknown neighborhood radio network model with high probability in $O(\Delta \log n \cdot \tau + \log^2 n)$ time.*

To illustrate the previous corollaries, consider for instance the local broadcasting problem, that is the problem of letting each node broadcasts a message to its neighbors. Using our results it is obvious that this problem can be solved with high probability in $O(\Delta \log n + \log^2 n)$ time. This could be for instance compared with a randomized $O(\Delta \log^3 n)$ time protocol for the same problem in the SINR radio model[6] (not the graph based model) given very recently in [GMW08].

4.2   On computing a coloring in unit disk graphs

Node and edge coloring are one of the most important and fundamental tasks in distributed radio networks. In fact, coloring can be considered as a basic tool to initially organize unstructured wireless ad hoc and sensor networks. This is well-motivated by associating different colors with different time-slots in a time-division multiple access (TDMA) scheme. A 'good' coloring corresponds to a medium access control (MAC) layer without direct interferences, that is, no two neighboring nodes send at the same time. The following paragraphs explore some coloring related problems in the unknown neighborhood model.

4.2.1   Node coloring

A correct node coloring is an assignment of a color to each node in the graph, such that any two adjacent nodes have a different color. In [MW05], the authors provide an algorithm that produces a correct coloring with $O(\Delta)$ colors in time $O(\Delta \log n)$[7] with high probability in a unit disk graph. That algorithm require that Δ is known to all nodes. In the conclusion of [MW05], it was asked whether we can get rid of this parameter. Using our algorithms, we answer in the positive to the latter question:

Theorem 4 *In the unknown neighborhood model, there exists a distributed algorithm that produces a $O(\Delta)$ node coloring with high probability within $O(\Delta \log n + \log^2 n)$ time on any n-node unit disk graph.*

Proof. The proof is based on reconsidering the coloring algorithm from [MW05] and replacing parameter Δ used there by the local maximum degree ($\Theta(\widetilde{\Delta_v})$) computed by our algorithm. By tuning the constants used in the algorithm of [MW05] and using the same arguments, we can prove the theorem. ∎

---

[6]Signal-to-Interference-plus-Noise-Ratio

[7]The authors also showed that $\Omega(\Delta)$ is a lower bound.

4.2.2 Partial strong edge coloring

A correct edge coloring is an assignment of a color to each edge in the graph, such that any two adjacent edges have a different color. One can find many variants of the edge coloring problem. For instance, in [BKM$^+$06], the authors considered the problem of computing a strong (or a distance-2) edge coloring, that is the problem of assigning distinct colors to any pair of edges between which there is a path of length at most two. The algorithms described there are based on computing an approximate maximum strong matching by an algorithm originally described in [BBK$^+$04]. Computing a strong maximum matching consists in computing a set, with maximum cardinality, of edges mutually at distance at least 2. In [BKM$^+$06], it is proved that a $O(1)$-approximate solution to the strong maximum matching can be computed in $O(\rho \log n)$ time for unit disk graphs, where ρ denotes the time it takes to compute the *active degree* of each node[8]. Following the same terminology than in [BKM$^+$06], a node is said to be *active* in a given round if it decides to transmit. Then, the active degree of a node is the number of its neighbors that are active in the current round. Using our algorithms, we can show that $\rho = O(\log^2 n)$. Thus, we obtain the following:

Theorem 5 *In the unknown neighborhood model, there exists a distributed algorithm that produces a $O(1)$-approximate strong matching with high probability in $O(\log^3 n)$ time on any n-node unit disk graph.*

5 Concluding remarks and open problems

In this paper, we have shown that computing in the unknown neighborhood model is up to a polylogarithmic factor similar to computing in the known neighborhood model. We have also shown that, for many distributed tasks, computing the local maximum degree of each node is sufficient to overcome the need to know the global parameter Δ. Many questions remain however open. For instance, we have not considered the energy efficiency measure in this paper. It would be interesting to derive new algorithms which are both time and energy efficient. Moreover, the applications given at the end of the paper raise other specific questions. For instance, we remark that the coloring algorithms for the unknown neighborhood model works under the assumption that the nodes wake up synchronously. It will be very nice to prove that our local maximum degree algorithm can be adapted to run under the weaker asynchronous wake up assumption which will allow us to derive pure asynchronous algorithms. Another field of research is to extend our algorithms to non graph based models and/or multi-channel radio networks. Our result are in fact a first step to understand the impact of the degree knowledge in setting up a uniform theory and practice of distributed algorithms in the unknown neighborhood model.

References

[ANLP89] N. Alon, A. Bar Noy, N. Linial, and D. Peleg. On the complexity of radio communication. In 21st *annual ACM Symposium on Theory of computing (STOC'89)*, pages 274–285, New York, NY, USA, 1989. ACM.

[BBK$^+$04] Hari Balakrishnan, Christopher L. Barrett, V. S. Anil Kumar, Madhav V. Marathe, and Shripad Thite. The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1069–1079, 2004.

[BKM$^+$06] Christopher L. Barrett, V. S. Anil Kumar, Madhav V. Marathe, Shripad Thite, and Gabriel Istrate. Strong edge coloring for channel assignment in wireless radio networks.

---

[8]No bound for ρ was given in [BKM$^+$06]. One should also remark that the model considered therein assumes that a node can hear collisions which is actually stronger than the assumptions of our model.

In 4th *annual IEEE international conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, page 106, Washington, DC, USA, 2006. IEEE Computer Society.

[BYGI89]   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. In 3rd *International Workshop on Distributed Algorithms (WDAG'89, DISC nowadays)*, pages 24–32, London, UK, 1989. Springer-Verlag.

[BYII89]   Reuven Bar-Yehuda, Amos Israeli, and Alon Itai. Multiple communication in multi-hop radio networks. In 8th *ACM Symposium on Principles of distributed computing (PODC'89)*, pages 329–338, 1989.

[GMW08]   Olga Goussevskaia, Thomas Moscibroda, and Roger Wattenhofer. Local Broadcast in the Physical Interference Model. In *Technical Report*, March 2008.

[GPX05]   Leszek Gasieniec, David Peleg, and Qin Xin. Faster communication in known topology radio networks. In 24th *annual ACM symposium on Principles of distributed computing (PODC'05)*, pages 129–137, New York, NY, USA, 2005. ACM.

[Hoe63]   W. Hoeffding. Probability inequalities for sums of bounded random variables, 1963.

[MR95]   R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[MU05]   Michael Mitzenmacher and Eli Upfal. *Probability and Computing, Randomized Algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[MW05]   Thomas Moscibroda and Roger Wattenhofer. Coloring Unstructured Radio Networks. In 17th *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'05), Las Vegas, Nevada, USA*, July 2005.