

# Optimal Initialization and Gossiping Algorithms for Random Radio Networks

Vlady Ravelomanana

ccsd-00085254, version 1 - 12 Jul 2006

Vlady Ravelomanana is with the LIPN – UMR 7030 (CNRS), Institut Galilée Université de Paris 13, France.  
E-mail : vlad@lipn.univ-paris13.fr

July 12, 2006

DRAFT

### Abstract

The initialization problem, also known as naming, consists to give a unique identifier ranging from 1 to  $n$  to a set of  $n$  indistinguishable nodes in a given network. We consider a network where  $n$  nodes (processors) are randomly deployed in a square (resp. cube)  $X$ . We assume that the time is slotted and the network is synchronous, two nodes are able to communicate if they are within distance at most of  $r$  of each other ( $r$  is the transmitting/receiving range). Moreover, if two or more neighbors of a processor  $u$  transmit concurrently at the same time slot, then  $u$  would not receive either messages. We suppose also that the anonymous nodes know neither the topology of the network nor the number of nodes in the network.

Under this extremal scenario, we first show how the transmitting range of the deployed processors affects the typical characteristics of the network. Then, by allowing the nodes to transmit at various ranges we design sub-linear randomized initialization protocols : In the two, resp. three, dimensional case, our randomized initialization algorithms run in  $O(n^{1/2} \log n^{1/2})$ , resp.  $O(n^{1/3} \log n^{2/3})$ , time slots. These latter protocols are based upon an optimal gossiping algorithm which is of independent interest.

### Keywords

Multihop networks; self-configuration in ad hoc networks; randomized distributed protocols; initialization; naming; gossiping; broadcasting; information dissemination; fundamental limits of random radio networks.

## I. INTRODUCTION

Distributed, multihop wireless networks, such as ad hoc networks, sensor networks or radio networks, are gaining in importance as subject of research [31]. Here, a network is a collection of transmitter-receiver devices, referred to as *nodes* (*stations* or *processors*).

Wireless multihop networks are formed by a group of nodes that can communicate with each other over a wireless channel. Nodes or processors come without ready-made links and without centralized controller. The network formed by these processors can be modeled by its *reachability graph* in which the existence of a directed edge  $u \rightarrow v$  means that  $v$  can be reached from  $u$ . If the power of all transmitters/receivers is the same, the underlying reachability graph is symmetric. As opposed to traditional networks, wireless networks are often composed of nodes whose number can be several orders of magnitude higher than the nodes in conventional networks [2]. Sensor nodes are often deployed inside a medium. Therefore, the positions of these nodes need not be engineered or pre-determined. This allows random and rapid deployment in inaccessible terrains and suit

well the specific needs to disaster-relief, law enforcement, collaborative computing and other special purpose applications.

As customary [3], [4], [5], [9], [17], [25], [26] the time is assumed to be slotted and nodes can send messages in synchronous *rounds* or *time slots*. In each round, every node can act either as a *transmitter* or as *receiver*. A node  $u$  acting as receiver in a given round gets a message, if and only if, exactly one of its neighbors transmits in the same round. If more than two neighbors of  $u$  transmit simultaneously,  $u$  receives nothing. That is, the considered networks do not have the ability to distinguish between absence of message and collision or conflict. This assumption is motivated by the fact that in many real-life situations, the (tiny) devices used do not always have the collision detection ability. Moreover, even if such detection mechanism is present, it may be of limited value especially in the presence of some noisy channels. Therefore, it is highly desirable to design protocols working independently of the existence/absence of any collision detection mechanisms.

We consider that a set of  $n$  nodes are initially *homogeneously scattered* in a square  $X$  of size  $|X|$  (or in a cube  $X$  of volume  $|X|$ ). As in several applications, the users of the network can move, and therefore the topology is unstable. For this reason, it is desirable for the protocols to refrain from assumptions about the network topology, or about the information that processors have concerning the topology. In this work, we assume that none of the processors have initially any topological information, except the measure (surface or volume)  $|X|$  of  $X$  where they are randomly dropped. We observe here that even if  $|X|$  is exactly known but not  $n$  then even if the order of  $n$  is known, viz.  $n = O(|X|)$ , equation such as (6) in the Theorem 2 (see below) allows us to handle the subtle changes involved in the constants hidden by the “big-Ohs” between  $O(n)$  and  $O(|X|)$ . Moreover, these assumptions are strengthened by the fact that during their deployment some nodes may be faulty with unknown probability.

Methods to achieve *self-configuration* and/or *self-organization* of networking devices appear to be amongst the most important challenges in wireless computing [2]. The initialization task is part of these methods : Before networking, each node must have a *unique identifier* (referred to as *ID* or *address*). A mechanism that allows the network to create a unique address (ID) automatically for each of its participating nodes is known

as the *address autoconfiguration* protocol. In this work, our nodes are initially *indistinguishable*. This assumption arises naturally since it may be either difficult or impossible to get interface serial numbers while on missions (see also [17], [25], [26]). Thus, the IDs self-configuration protocols do not have to rely on the existence of serial numbers.

The problem we address here is then to design a *fully distributed protocol* for the initialization problem (also known as *naming* problem). As far as we know, the initialization problem was first handled in the seminal papers of Hayashi, Nakano and Olariu [17], [25], [26] for the case when the underlying reachability graph is a complete one.

We remark that the transmitting range of each station can be set to some value  $r$  ranging from 0 to  $r_{\text{MAX}}$ . This model is commonly used in mobile computing and radio networking [7], [19], [32]. Note that such modelization is frequently encountered in many domains ranging from statistical physics to epidemiology (see for example [16] for the theory of coverage processes or [23] for percolative ingredients). The random graphs generated this way have been considered first in the seminal paper of Gilbert [14] (almost at the same time Erdős and Rényi considered the well-known  $\mathbb{G}(n, p)$  model [12]) and analysis of their properties such as connectivity and coverage have been the subject of intense studies [15], [24], [27], [28], [29], [30]. Figure 1 shows devices which have been deployed on some field in a random fashion. The depicted examples suggest that transmission ranges can play a crucial role when setting protocols at least for randomly distributed nodes. Other important parameters are the number  $n$  of active stations, the shape of the area  $X$  where the nodes are scattered and the nature of the communications to be established.

According to these observations, to design efficient protocols, we have to take into account and to exploit the structural properties of the reachability graph. In our scenario, since none of the nodes knows the number  $n$  of the processors in the network, our first task is to find distributed protocols that allow (probabilistic) counting of these nodes. We then go on to show that by setting the transmitting range parameter correctly, the network can be auto-initialized, with high probability<sup>1</sup>, on first hand in  $O(n^{1/2} \log n^{1/2})$  time slots for the two dimensional case and on the other hand, in  $O(n^{1/3} \log n^{2/3})$  steps

<sup>1</sup>Throughout this paper, an event  $\mathcal{E}_n$  is said to occur *asymptotically almost surely* if and only if the probability  $Pr[\mathcal{E}_n]$  tends to 1 as  $n \rightarrow \infty$ . We also say  $\mathcal{E}_n$  occurs *with high probability* (w.h.p. for short).

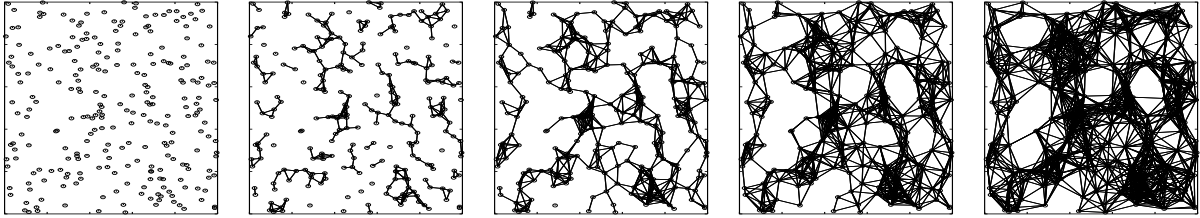


Fig. 1

A TYPICAL RADIO NETWORK IS GENERATED VIA UNIFORM DISTRIBUTION OF THE  $x$  AND  $y$  COORDINATES OF THE DEVICES AND THE TRANSMISSION RANGES OF THE NODES ARE INCREASED GRADUALLY (FROM LEFT TO RIGHT). THE TWO LAST PICTURES SHOW THAT IF THE OBTAINED GRAPH HAS MORE EDGES THAN NEEDED, THE NUMBER OF COLLIDING PACKETS IS MORE DIFFICULT TO CONTROL.

in the three-dimensional case<sup>2</sup>. As far as we know, this is the first analysis for the initialization protocols in the multihop cases (the single-hop cases have been treated in the literature in [17], [25], [26], [32]). Our algorithms are shown to take advantage of the fundamental characteristics of the network. These limits are computed with the help of fully distributed protocols and once known, an initialization algorithm is run to assign each of the  $n$  processors a distinct ID number in the range from 1 to  $n$ . Even though the protocols are probabilistic, once the IDs are attributed their uniqueness can be checked (if needed) *deterministically* by for example the use of deterministic linear algorithms such

In order to settle the initialization problem, we use a gossiping algorithm. Gossiping as the gossiping protocol for symmetric networks in [22, Section 5].

is with broadcasting [8] one of the fundamental tasks for information dissemination and Under the conditions described above, the Figures 2 and 3 summarize briefly the input represents naturally one of the most extensive studied problems in radio networks (see for and output of the distributed initialization protocols presented in this work.

instance [9], [22] and references therein). In the gossiping problem, every node is initially given a (different) message that needs to be distributed to all other nodes. Under the same assumptions as above, we design a randomized gossiping protocol that achieves its designated task w.h.p. in  $O(n^{1/2} \log n^{1/2})$ , resp.  $O(n^{1/3} \log n^{2/3})$ , time slots for the two, resp. three, dimensional settings.

Finally, it is shown that our *sub-linear* algorithms, both the gossiping and the initialization protocols, are *asymptotically optimal* since they achieved their designated tasks

<sup>2</sup>In this paper, log is referred to as the natural logarithm.

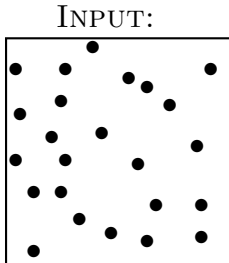


Fig. 2

$n$  INDISTINGUISHABLE AND IDENTICAL PROCESSORS RANDOMLY PLACED IN THE SQUARE  $X$ . THE ONLY KNOWLEDGE REQUIRED IS THE SIZE  $|X|$  OF THE SUPPORT.

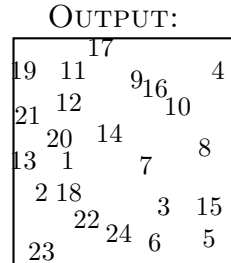


Fig. 3

EACH OF THE  $n$  PROCESSORS IS ASSIGNED A UNIQUE ID RANGING FROM 1 TO  $n$ . THESE IDS CAN SERVE AS IP ADDRESS (HERE  $n = 24$ ).

(w.h.p.) in  $O(D \log n) = O(D\Delta)$  time slots, where  $\Delta$  is the maximum degree of the underlying network and  $D$  represents its eccentricity (hop-diameter).

**Outline of the paper.** The remainder of this paper is organized as follows. Section 2 first presents a randomized protocol **SEND** which is a distributed algorithm for sending information in our settings. We then analyze this protocol. In Section 3, we discuss how to set correctly the transmission range of the nodes. Section 3 also offers results about the relationship between the transmission range  $r$ , the number of active nodes  $n$ , the size of  $X$ , the maximum degree  $\Delta$  and the hop-diameter  $D$  of the wireless networks. These results and the use of the procedure **SEND** allow us to build a broadcasting protocol named simply **BROADCAST**. Section 3 ends with the design and analysis of a protocol called **SFR** (**SFR** stands for “*search for range*”) which serves to find (in a distributed way) the appropriate transmission range. More precisely, by varying the transmission range, the protocol **SFR** gives us ideas on the orders of magnitude of the fundamental characteristics of the network. Section 4 presents the randomized gossiping protocol specifically intended for random wireless networks. This Section is organized as follows : We first present

a randomized algorithm that colors the nodes of the underlying graph in such a way that every pair of two processors  $(u, v)$  at distance at most 2 hops from each other are assigned different colors. Even “greedy”, this latter algorithm is shown to color the graph in polylogarithmic time slots (w.r.t.  $n$ ) using  $O(\Delta) = O(\log n)$  colors. This efficient coloring protocol treats the direct and hidden terminal problems. Once obtained, the 2-hop coloration leads to a natural scheduling of the communications in order to gossip in  $O(D\Delta)$  rounds. In its turn, the gossiping protocol is used to initialize the network. This is easily done by means of a simple ranking argument. Section 4 ends with the proofs of correctness and optimality of both algorithms (the gossiping and initialization protocols).

## II. THE BASIC PROTOCOL FOR SENDING INFORMATION

First of all, no deterministic protocol can work correctly in the networks when processors are anonymous. This can be easily checked : conflict between two processors absolutely identical can not be solved deterministically. Therefore, this impossibility result implies the use of randomness (see [5]). Since our processors do not have identifiers, our first task is to build a basic protocol for the nodes which compete locally to access the unique channel of communication in order to send a given message. This can be achieved by organizing a flipping coin game between them. Recall also that if the transmission/receiving range is set to a value  $r$ , only neighbors of distance at most  $r$  are able to communicate when conflicts are absent. In [29], Penrose proved that there exists a common radius of transmission to achieve the connectivity of the reachability graph.

In the following procedure we have to take into account this parameter as well as the duration  $T$  of the trials :

**Procedure** SEND( $msg, T, r$ )

**For**  $i$  from 0 to  $T$  **do**

With probability  $1/2^i$  send  $msg$  to every neighbor

(★ that is to all processors within distance at most  $r$  ★)

**end.**

Note that  $r$  is here a parameter which can be tuned to a precise value. Again, it should be clear now that only neighbors within distance at most  $r$  can receive the message when

there is no conflict. Therefore, we have the following definition :

*Definition 1:* Given a transmission radius  $r$  and a set of  $n$  nodes uniformly independently scattered on a square  $X$  of size  $|X| = O(n)$ , a random graph is defined by adding edge between any pair of nodes  $(x, y)$  such that the Euclidean distance between  $x$  and  $y$  is less than or equals to  $r$ . Denote by  $r_{\text{CON}}$  the transmission range required to have a connected graph. For a fixed radius of transmission  $r$ , let  $d_v$  be the degree of any given node  $v$  which depends on  $r$ , i.e.  $d_v \equiv d_v(r)$ .

*Theorem 1:* Let  $r \geq r_{\text{CON}}$  be the current transmission range of the processors. Suppose that each of the  $d_v$  neighbors of  $v$  starts the execution of **SEND**( $msg, T, r$ ) at the same round. Let  $P(T, d_v)$  be the probability that  $v$  will receive the message  $msg$  at least once between the time  $t = 0$  and  $t = T$ . Then, there exists a function  $f(T, d_v) = O\left(\frac{d_v}{2^T}\right) + O\left(\frac{1}{\sqrt{d_v}}\right)$  such that  $P(T, d_v)$  satisfies

$$0.8111 + f(T, d_v) \leq P(T, d_v) \leq 0.8113 + f(T, d_v). \quad (1)$$

*Proof:* The assumption that the reachability graph is connected insures that for all processor  $v$ , the degree of  $v$  verifies  $d_v > 0$ . We have  $P(T, d_v) = 1 - \prod_{i=0}^T \left(1 - \frac{d_v}{2^i} \left(1 - \frac{1}{2^i}\right)^{d_v}\right)$  since only one of the  $v$  neighbors can succeed :  $v$  and the other  $d_v - 1$  nodes are kept silent. For any given  $i_1$  and for all  $i \geq i_1$ , we have  $\left(1 - \frac{d}{2^i} \left(1 - \frac{1}{2^i}\right)^d\right) \leq \left(1 - \frac{d}{2^i} \exp\left(-\frac{d}{2^i} \left(1 + \frac{1}{2^i}\right)\right)\right)$ . So, if  $2^t \gg d$  by choosing  $i_1 = \lceil \frac{1}{2} \log_2 d \rceil$  after standard calculus we obtain

$$1 - P(t, d) \leq \exp\left(-\sum_{m \geq 1} \frac{1}{m} \sum_{i=i_1}^t \frac{d^m}{2^{im}} \exp\left(-\frac{dm}{2^i} \left(1 + O\left(\frac{1}{\sqrt{d}}\right)\right)\right)\right).$$

Using Mellin transform asymptotics methods (see [13] and [20, p. 131]), for any  $m \geq 1$  we get

$$\left| \sum_{i=i_1}^t \frac{d^m}{2^{im}} \exp\left(-\frac{dm}{2^i} \left(1 + O\left(\frac{1}{\sqrt{d}}\right)\right)\right) - \frac{m!}{m^{m+1} \log 2} \right| \leq \frac{10^{-5}}{m^m \log 2} + O\left(\frac{1}{\sqrt{d}}\right) + O\left(\frac{d^m}{2^{tm}}\right), \quad (2)$$

where the  $10^{-5}$  term is due to the fluctuation in the Fourier series involved in the Mellin asymptotic calculations [13]. Next, since  $\frac{m!}{m^{m+2}} \leq \frac{e^{-m}}{m}$  if  $m \geq 7$  we have

$$\sum_{m=1}^6 \frac{m!}{m^{m+2}} \leq \sum_{m=1}^{\infty} \frac{m!}{m^{m+2}} \leq \sum_{m=1}^6 \frac{m!}{m^{m+2}} + \sum_{m=7}^{\infty} \frac{e^{-m}}{m}.$$

Since

$$\sum_{m=7}^{\infty} \frac{e^{-m}}{m} = -\frac{1}{60e^6} \left( 60e^6 \ln(1 - 1/e) + 60e^5 + 30e^4 + 20e^3 + 15e^2 + 12e + 10 \right),$$

it comes

$$0.18869\dots \leq \exp\left(-\sum_{m=1}^{\infty} \frac{m!}{m^{m+2} \ln 2}\right) \leq 0.18879\dots \quad (3)$$

Similarly for any  $x \in [0, 1]$  and  $d \geq 1$ , we have  $(1-x)^d \leq e^{-dx}$ . Therefore,  $\left(1 - \frac{d}{2^i} \left(1 - \frac{1}{2^i}\right)^d\right) \geq 1 - \frac{d}{2^i} \exp\left(-\frac{d}{2^i}\right)$ . This time we get

$$\exp\left(-\sum_{m \geq 1} \frac{1}{m} \sum_{i=i_1}^t \frac{d^m}{2^{im}} \exp\left(-\frac{dm}{2^i}\right)\right) \leq 1 - P(t, d).$$

Using this latter, after similar computations as for (2) and (3), we get the inequalities (1). ■

In [5], Bar-Yehuda *et al.* have designed a randomized procedure called **DECAY** to send information and whose probability of success is greater than  $\frac{1}{2}$  (see for instance [5, pp 108–109]). In our procedure **SEND**, the proof of Theorem 1 (see also [13]) shows that by changing the basis of the coin flipping game, viz. replacing the probability  $1/2^i$  in the algorithm by  $1/a^i$  for any constant  $a > 1$ , the probability of success of the  $T$  trials can be made arbitrary close to 1 (after similar logarithmic number of time slots satisfying  $a^T \gg d$ ).

In the next Section, we turn on the problem of finding suitable values of transmission range whenever the only *a priori* knowledge of the processors is  $|X|$ .

### III. TRANSMISSION RANGES AND CHARACTERISTICS OF THE NETWORK

The aim of this Section is to provide randomized distributed algorithms that allow the nodes in the network to find the right transmission range such that the reachability graph is at least connected. To this end, we need to know the relationships between the transmission range  $r$ , the number of processors  $n$  and the measure  $|X|$  of the support. Other fundamental characteristics of the graph, such as the minimum (resp. maximum) degree  $\delta$  (resp.  $\Delta$ ) and the hop-diameter  $D$  are also of great interest when designing wireless protocols (see [5]). Moreover, the limits of the randomly generated network of

processors help when designing algorithms for such network. We refer here to [14], [15], [24], [30], [34], [35] for works related to of random networks. Two distinct paragraphs are treated in this Section.

- The first one concerns the characteristics of the reachability graph in the superconnectivity regime, i.e. when the radius of transmission of the nodes grows much more faster than the one required to achieve the connectivity of the graph.
- The second subsection is devoted to the design and analysis of a distributed protocol, called **SFR**, that will allow the nodes to approximate the aforementioned characteristics.

#### A. Fundamental limits of a random network in the superconnectivity regime

For several reasons, we follow the Miles's model [24]. In this model, a large number  $n$  of devices are dropped in some area  $X$ . As  $n \rightarrow \infty$  but  $n = O(|X|)$ , the graph generated by the transmitting devices can be well approximated with a Poisson point process (see for instance [16]). First of all, this extreme independence property allows penetrating analysis. Next, since Poisson processes are *invariant* if their points are independently translated (the translations being identically distributed following some bivariate distribution : direction and distance), the results can take their importance for *moving nodes* and therefore, they are well suited to cope with randomly deployed mobile devices. Third, due to Poisson processes properties, if with probability  $p$ , such that  $p \times n = O(|X|)$ , some nodes are *faulty* or intentionally *asleep* (e.g. to save batteries to design energy-efficient algorithms [26]), our results remain valid. In this latter scenario, the number of nodes  $n$  is simply replaced by  $n' = p \times n$ .

Among other results, Penrose [29] proved (with our notations) that if  $n/|X| = O(1)$  and  $X$  is a two dimensional area then :

$$\lim_{n \rightarrow \infty} Pr \left[ \pi \frac{n}{|X|} r_{\text{CON}}^2 - \log(n) \leq \omega \right] = \exp(-e^{-\omega}), \quad \omega \in \mathbb{R}. \quad (4)$$

Penrose's result tells us that by letting the radius of transmission range growing as

$$r = \sqrt{\frac{\log n + \omega(n)}{\pi n}} |X|, \quad (5)$$

for any arbitrary function of  $\omega(n)$  tending to infinity with  $n$ , the obtained graph of the network is a.a.s. connected.

For our purpose, we need the following results related to the degrees of the nodes according to successive values of the transmission range :

*Theorem 2:* Denote by  $r$  the transmission range of the  $n$  nodes randomly distributed in the square  $X$  of size  $|X| = O(n)$ . Then, in the following regimes with high probability the graph is connected and we have :

(i) For fixed values of  $k$ , that is  $k = O(1)$ , if  $\pi \frac{n}{|X|} r^2 = \log n + k \log \log n + \omega(n)$ , then the graph has a.a.s. a minimum degree of  $\delta = k$ .

(ii) Let  $k \equiv k(n)$  but  $1 \ll k \ll \log n / \log \log n$ . If  $\pi \frac{n}{|X|} r^2 = \log n + k(n) \log \log n$ , then the minimum degree (resp. maximum degree) is a.a.s.  $\delta = k(n)$  (resp.  $\Delta = e \log n$ ).

(iii) If  $\pi \frac{n}{|X|} r^2 = (1 + \ell) \log n$  with  $\ell > 0$  then each node  $v$  of the graph has a.a.s.  $d_v$  neighbors with

$$-\frac{\ell \log n}{W_{-1}\left(-\frac{\ell}{e(1+\ell)}\right)} + o(\log n) \leq d_v \leq -\frac{\ell \log n}{W_0\left(-\frac{\ell}{e(1+\ell)}\right)} + o(\log n), \quad (6)$$

where  $W_{-1}$  and  $W_0$  denote the two branches of the Lambert W function<sup>3</sup> which are detailed in [10]. Moreover, each geographical point of the support  $X$  is also recovered by  $\Theta(\log n)$  disks of transmission in the case  $\pi \frac{n}{|X|} r^2 = (1 + \ell) \log n$ , with  $\ell > 0$ .

For the proof of Theorem 2, we refer to [32] where asymptotic coverage as well as connectivity properties are treated in detail for the ranges of transmission considered in the Theorem 2.

Observe that with the same assumptions as in Theorem 2 but in the 3 dimensional case (with a cube instead of a square), similar results as above hold with every occurrence of the surface “ $\pi r^2$ ” replaced by the volume “ $\frac{4}{3}\pi r^3$ ”. For example, to have each point of the cube recovered by  $\Theta(\log n)$  balls, it suffices to set the transmission radius to  $r = \sqrt[3]{\frac{3(1+\ell)\log n |X|}{4\pi n}}$ . In this case, w.h.p. the degree  $d_v$  of each node  $v$  satisfies also (6).

Throughout the rest of this paper, we mainly concentrate our attention on the results related to the 2 dimensional case, since there are direct correspondences with the 3 dimensional case such as the one mentioned above.

Next, we derive an upper-bound of the hop-diameter  $D$  in the superconnectivity regime :

<sup>3</sup>The Lambert W function is considered as a special function of mathematics on its own and its computation has been implemented in mathematical softwares such as Maple.

*Theorem 3:* Let  $D \equiv D(r)$  be the hop-diameter of the graph. Suppose that the transmission range satisfies  $r = \sqrt{\frac{(1+\ell)\log n}{\pi n} |X|}$  with  $\ell > 0$ . We then have :

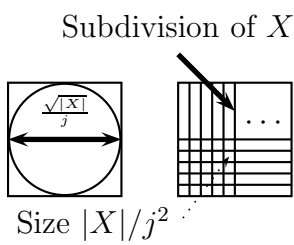
(i) If  $\ell > \frac{4-\pi}{\pi-2}$  then

$$\lim_{n \rightarrow \infty} Pr \left[ D \leq 3 \sqrt{\frac{\pi n}{(1+\ell) \log n}} + O(1) \right] = 1. \quad (7)$$

(ii) If  $\ell \leq \frac{4-\pi}{\pi-2}$  then

$$\lim_{n \rightarrow \infty} Pr \left[ D \leq 5 \sqrt{\frac{\pi n}{(1+\ell) \log n}} + O(1) \right] = 1. \quad (8)$$

*Proof:* Split the square  $X$  into  $j^2$  equal subsquares  $S_1, S_2, \dots, S_{j^2}$ . Each of the subsquares has a side  $\sqrt{|X|}/j$  and an area  $|X|/j^2$ . Choose  $j$  such that each subsquare  $S_i$  can contain entirely a circle of radius equals to  $r$  as depicted below.



That is  $\frac{\sqrt{|X|}}{2j} = r = \sqrt{\frac{(1+\ell)\log n |X|}{\pi n}}$ . So,  $j = \frac{1}{2} \sqrt{\frac{\pi n}{(1+\ell) \log n}}$ . For sake of simplicity but w.l.o.g., we suppose that  $j \in \mathbb{N}$ . By the Theorem 2 property (iii) given above, with high probability we have  $\Theta(\log n)$  nodes inside the circle.

Any pair of processors inside the same circle need *at most* 2 hops to be connected since they are at distance at most  $2r$  and since each subgraph inside such a circle is a.a.s. connected.

We claim that from two adjacent subsquares  $S_1$  and  $S_2$ , communications between any node  $a \in S_1$  and any node  $b \in S_2$  need at most (w.h.p.) :

- a) 6 hops for  $\ell > \frac{4-\pi}{\pi-2} = 0.7519\dots$  and
- b) 10 hops for  $\ell \leq \frac{4-\pi}{\pi-2}$ .

To prove these assertions, consider adjacent subsquares as depicted in Figures 4, 5, and 6. A bit of trigonometry shows that each lens-shaped region such as  $L_1$  has a surface  $|L_1| = \frac{1}{6}(4\pi - 3\sqrt{3})r^2$ . Note that  $L_1$  represents the intersection of two disks of equal radius  $r$  whose centers are at distance  $r$ . Therefore, there is no node inside the lens-shaped region  $L_1$  with probability

$$\left(1 - \frac{|L_1|}{|X|}\right)^n = \left(1 - \frac{1}{6}(4\pi - 3\sqrt{3})\frac{(1+\ell)\log n}{n}\right)^n \leq \exp\left(-\frac{1}{6}(4\pi - 3\sqrt{3})(1+\ell)n\right). \quad (9)$$

AT MOST 6 HOPS

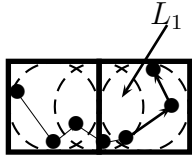


Fig. 4

HORIZONTAL TRANSMISSION.

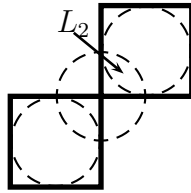


Fig. 5

DIAGONAL TRANSMISSION.

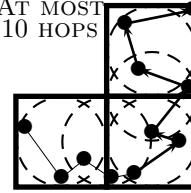
AT MOST  
10 HOPS

Fig. 6

“INDIRECT” TRANSMISSION.

Since each subsquare has at most 4 lenses of size  $|L_1|$ , none of these regions is empty with probability at least

$$\left(1 - \exp\left(-\frac{1}{6}(4\pi - 3\sqrt{3})(1 + \ell)n\right)\right)^{(4 \times j^2)} \geq \exp\left(-2 \times \frac{\pi n^{1-\frac{1}{6}}(4\pi - 3\sqrt{3})(1 + \ell)}{(1 + \ell) \log n}\right). \quad (10)$$

Hence, with probability tending to 1 as  $n \rightarrow \infty$ , in every lens-shaped region of size  $|L_1|$  there is at least a node. Thus, to transmit message between two horizontally (or vertically) adjacent subsquares, we need at most 6 hops (see Figure 4).

To prove b), we consider lenses such as  $L_2$  depicted in Figure 5. The size of such region is  $|L_2| = \frac{\pi-2}{2}r^2$  which measures the area of the intersection of two equal disks of radius  $r$  and at distance  $\sqrt{2} \times r$ . Arguing as for (10), we find that for every lens of size  $|L_2|$  to be non-empty (w.h.p.) we need that  $(1 + \ell)\frac{\pi-2}{2} > 1$ . This condition is only satisfied if  $\ell > \frac{2}{\pi-2} - 1 = 0.7519\dots$ . For values of  $\ell \leq 0.7519\dots$ , transmissions are sent horizontally then vertically (or vice-versa). Such transmissions can required up to 10 hops (cf. Figure 6). The proof of the Theorem is now easily completed by simple counting arguments. ■

In the 3 dimensional case, we have

*Theorem 4:* Suppose that  $n$  sensor nodes are randomly deployed in a cubic region of volume  $|X|$  according to the uniform distribution. If their common transmission range is set to  $r = \sqrt[3]{\frac{3(1+\ell) \log n |X|}{4\pi n}}$  with  $\ell > 11/5$ , then the diameter  $D$  of the network satisfies :

$$\lim_{n \rightarrow \infty} Pr \left[ D \leq 12 \sqrt[3]{\frac{\pi n}{6(1 + \ell) \log n}} \right] = 1. \quad (11)$$

*Proof:* See [32]. ■

### B. Exploiting the fundamental limits and adjusting the transmission range distributedly

The previous paragraph gives us almost sure characteristics of the network but we need to verify and to exchange these informations by means of distributed protocols. To this end, we need two procedures. The first one is the **BROADCAST** protocol. In this protocol, some processors (called *sources*) try to disseminate a given message to all the nodes in the network. It makes several calls of **SEND**. The second procedure **SFR** (for “search-for-range”) is used to adjust the correct transmission range of the nodes in order to “take control” of the main characteristics of the network. **SFR** works as follows.

Each processor starts with the maximum range of transmission. Then at each step, the transmission range is diminished gradually until the disconnection of some of the nodes. At this stage, the newly isolated nodes readjust their transmission range in order to be re-connected and use the protocol **BROADCAST** in order to disseminate a “disconnection” message (to advert all the processors in the network). A processor quits the protocol if and only if either it has been isolated once, has been reconnected and has sent the “disconnection” message or it has received the “disconnection” message containing information about the adequate transmission range.

#### B.1 The broadcasting protocol

The procedure **BROADCAST** is similar to the one in [5] except that we use **SEND** to transmit messages.

**Procedure** **BROADCAST**( $msg, \varepsilon, \Delta, r, N$ )

$k := 2\lceil \log_2 \Delta \rceil$

( $\star \Delta$  is an upper-bound of the maximum degree  $\star$ )

$\tau := \lceil \log_2 (N/\varepsilon) \rceil$

( $\star N$  is an upper-bound of the number of nodes  $\star$ )

Wait until receiving a message  $msg$

**For**  $i$  from 1 to  $\tau$  **do**

Wait until **TIME** mod  $k = 1$

( $\star$  to synchronize  $\star$ )

**SEND**( $msg, k, r$ )

( $\star$  attempt to send  $msg$   $\star$ )

**end.**

In the procedure above,  $\varepsilon > 0$  can be made arbitrarily small.  $\Delta$  is a parameter representing

the maximum degree of the network (or an upper bound of the maximum degree. This can be computed for a given value of the transmission range using Theorem 2).  $N$  is an upper-bound of the number of participating nodes. **TIME** is a protocol which allows a given node to have the current time. Following the proof of [5, Theorem 4], we have :

*Theorem 5: Bar-Yehuda, Goldreich, Itai [5].* Suppose that  $r \geq r_{\text{CON}}$  is the actual transmission range of the nodes. Assume that  $\Delta$  (resp.  $N$ ) is an upper-bound of the maximum degree (resp. the number of nodes) in the network and let  $T = 2D + 5 \times \max(\sqrt{D}, \sqrt{\log_2(N/\varepsilon)}) \times \sqrt{\log_2(N/\varepsilon)}$ . Assume that some initiators (or sources) start the procedure **BROADCAST**( $msg, \varepsilon, \Delta, r, N$ ) when **TIME** = 0. Then, with probability  $\geq 1 - 2\varepsilon$  after  $2\lceil \log_2 \Delta \rceil T$  time slots, all the nodes receive the message. Furthermore, with probability  $\geq 1 - 2\varepsilon$  all the nodes have terminated by time  $2\lceil \log_2 \Delta \rceil (T + \lceil \log_2(N/\varepsilon) \rceil)$ .

## B.2 Adjusting the transmitting range : the protocol **SFR**

The processors need to know bounds of the value of the number  $n$  of the processors. If  $p_0 = \lfloor \log_2 n \rfloor$  then  $2^{p_0} \leq n < 2^{p_0+1}$ . Thus, by setting  $R(2^p) := \sqrt{\frac{(\log(2^p)+2\log 2)|X|}{\pi 2^p}}$  if the value of  $p$  increases,  $R(2^p)$  decreases. In the protocol **SFR**, we increment  $p$  one by one, starting at a value close to the maximal transmission range of the processors. When  $p$  passes through  $p_0 - 1$ ,  $p_0$  and  $p_0 + 1$ , there will be w.h.p. some new isolated nodes. In fact, elementary calculus show that  $\sqrt{2 \times \frac{\log n |X|}{\pi n}} \leq R(2^{p_0-1})$ . We are now ready to give the protocol **SFR**. The procedure **SFR** has just one variable  $\varepsilon$  representing the tolerance parameter and this protocol is executed in parallel by each station. The details follow :

( L0) **Procedure SFR**( $\varepsilon$ )

( L1) **BEGIN**

( L2)  $R := x \mapsto \sqrt{\frac{(\log(2^x)+2\log 2)|X|}{\pi 2^x}} ;$

( L3)  $B := x \mapsto 24 \left\lceil \log x \times \left( \sqrt{\frac{2^x}{x}} + x - \log_2(\varepsilon) \right) \right\rceil ;$  ( $\star B(x)$  is the broadcast time.  $\star$ )

( L4) **DISCONNECTED** := false ;

( L5)  $p := \left\lceil \log_2(r_{\text{MAX}}) \right\rceil ;$

( L6) **REPEAT**

( L7)  $counter := 0;$

( L8)  $t := 100 \times \left( \left\lceil \log_2(p) \right\rceil + \left\lceil \log_2(2/\varepsilon) \right\rceil \right);$

( L9) **For**  $i$  from 1 to  $t$  **Do**

(L10)  $\text{SEND}("p", i, R(p));$

(L11)           **If** receiving a message “p” **Then**  
(L12)                      $counter := counter + 1$ ;  
(L13)           **EndIf**  
(L14)       **EndFor**  
(L15)       **If**  $counter = 0$  **Then**  
(L16)           **For**  $j$  from 1 to  $\left\lceil \log_2 \left( \frac{2}{\varepsilon} \right) \right\rceil$  **Do**  
(L17)                     **BROADCAST**(“Disconnection  $p$ ”,  $\varepsilon$ ,  $3p$ ,  $R(p - 1)$ ,  $2^{p+1}$ ) ;  
(L18)           **EndFor**  
(L19)           DISCONNECTED := true ;  
(L20)       **Else**  
(L21)           Wait for a message for  $\left\lceil \log_2 \left( \frac{2}{\varepsilon} \right) \right\rceil \times B(p - 1)$  times ;  
(L22)           **If** receiving the “disconnection message” **Then**  
(L23)                     Scan the value of  $p$  and set DISCONNECTED := true;  
(L24)           **Else**  $p := p + 1$  ;  
(L25)           **EndIf**  
(L26)       **EndIf**  
(L27)       **UNTIL** DISCONNECTED := true ;

When reaching the value of  $p_0$ , the isolated nodes – whose transmission ranges are now set to  $r = R(p_0)$  – can increase back their transmission range, viz.  $R(p_0 - 1)$ , in order to be reconnected. Next, these processors have to advert the others about upper-bounds of  $n$ ,  $\Delta$  and  $D$ , respectively given by

$$2^{p_0} \leq n < 2^{p_0+1}, \quad \Delta \leq \frac{1}{-W_0(-e^{-1}/2)} \log n < 3p_0 \text{ and } D \leq 5 \sqrt{\frac{\pi 2^{p_0}}{(p_0 + 1) \log 2}} < 12 \left\lceil \sqrt{\frac{2^{p_0}}{p_0}} \right\rceil, \quad (12)$$

where we used Theorems 2 and 3 for respectively  $\Delta$  and  $D$  with  $\ell = 1$  and the transmission range is set to

$$r = \sqrt{\frac{\log(2^{p_0-1}) |X|}{(2^{p_0-1}) \pi}}. \quad (13)$$

The message of disconnection can be sent and received correctly by means of multiple calls to the protocol **BROADCAST** but we have to give sufficient time slots — cf. (L21) — to the broadcasting processors in order to let the others be aware of the bounds given by (12). The message sent for these advertisements is represented by a special message, say “*Disconnection  $p_0$* ” which contains the right value of  $p_0$ . Taking into account (12), we

remark that the “broadcast time” given by the Theorem 5 is (with probability greater than  $1-2\varepsilon$ ) less than  $2\lceil\log_2 \Delta\rceil \times (2D+5 \times \max(\sqrt{D}, \sqrt{\log_2(N/\varepsilon)}) \times \sqrt{\log_2(N/\varepsilon)} + \lceil\log_2(N/\varepsilon)\rceil)$ . This is strictly less than  $24 \log(p_0) \times \left(\sqrt{\frac{2p_0}{p_0}} + p_0 - \log_2(\varepsilon)\right)$ . Given these descriptions, the protocol **SFR** has the following properties :

*Theorem 6:* Assume that the random deployed network is an instance satisfying (12). For any  $c > 0$  there exists a constant  $c_1 > 0$  such that with probability at least  $1 - \frac{1}{n^c}$ , the protocol **SFR**( $\frac{1}{n^{c_1}}$ ) terminates in at most  $O(D \times \log n)$  time slots. After this time, with probability at least  $1 - \frac{1}{n^c}$ , every node is aware of the upper-bounds of the values of  $n$ ,  $\Delta$  and  $D$ .

*Proof:* In lines (L9)–(L14), the inner loop is repeated  $t$  times. Consider a random picked node  $v$ . By Theorem 1 for any given node  $v$ , as soon as  $i$  in line (L9) satisfies  $2^i \gg d_v$ , the probability of success of each call of **SEND** is at least 0.8.... By Theorem 2, and under the hypothesis that the graph satisfies the almost sure properties of a random network, if  $p = p_0 = \lfloor \log_2(n) \rfloor$ ,  $d_v < 3p_0$ . Therefore, by setting  $t$  as in line (L8), we insure that if the node  $v$  is still connected, it will receive more than one message from its neighbors with probability at least  $1 - \frac{\varepsilon}{2}$ . Similarly, by repeating sufficient calls of **BROADCAST** for the just disconnected nodes (see the discussion above) and give sufficient time to them to send the message of disconnection to the others, w.h.p. we allow all the processors of the whole network to know the correct upper-bounds of  $n$  (and thus  $\Delta$  and  $D$ ). To explain the constants  $c$  and  $c_1$  involved in the result, one can always choose  $\varepsilon$  of the form  $\varepsilon = 1/n^{c_1}$  in order to obtain probabilities of failure of order  $1/n^c$ . ■

According to these results and throughout the rest of the paper, we have the following definition.

*Definition 2:* A random graph  $G$  is said *typical* if and only if

- (i) it satisfies Theorem 2 and Theorem 3 and
- (ii) for any constant  $c_1 \geq 1$ , after one invocation of the protocol **SFR**( $\frac{1}{n^{c_1}}$ ), every node of the graph  $G$  knows the same value of  $p_0$  satisfying (12).

#### IV. NAMING THE ANONYMOUS RANDOM RADIO NETWORKS

We have settle in the previous Section the problem of determining the correct transmission range for the nodes of a random network. Such network typically has the charac-

teristics (mainly maximum degree and hop-diameter) dictated by Theorems 2 and 3. We also know through the protocol **SFR** probabilistic upper-bounds of such characteristics. In [4], Bar-Yehuda *et al.* gave protocols for efficient emulation of a single-hop network with collision detection on multi-hop radio network, provided that the number of nodes, the diameter and the maximum degree of the network (or upper-bounds of them) are known. Combination of the results in [4], in [25] with the results in the previous paragraphs lead to a new initialization protocol. That is, we can emulate a complete network (with collision detection) using the methods in [4] and therefore, using any broadcasting protocol with the Nakano-Olariu algorithms in [25], it is possible to build an initialization protocol in time  $O(n \times B)$  where  $B$  represents the broadcast time and is of order of magnitude  $B = O(D \log n)$  for the networks under consideration (see for instance [5], [9], [22]).

Rather than emulating a complete network, we first plan to color the graph in a particular manner : the two-hop coloration. In this problem, the nodes of the network are colored in such a way that every pair of stations  $(u, v)$  in hop-distance at most 2 from each other are assigned different colors (codes or “channel assignment”). This specific coloration gives the graph a natural scheduling of the communications which avoids *direct and hidden terminal problems* : if any pair of nodes  $(u, v)$  at hop-distance  $\leq 2$  are allowed two codes  $c_u, c_v$  (with  $c_u \neq c_v$ ) and decide to transmit at the time slots corresponding directly to these codes then it is easily seen that such scheduling is *collision-free*.

#### A. Choosing temporary IDs

Since the nodes are supposed to be indistinguishable, our present task is to attribute them temporary distinct IDs. If an upper-bound  $N$  of  $n$  is known, this can be done simply by affecting to each node an integer uniformly picked from the range  $[1, N^3]$ . The details follows :

##### **Procedure TMPIDS( $N$ )**

Each node chooses uniformly at random

an integer ranging from 1 to  $N^3$

**end.**

The above procedure has the following property :

*Theorem 7:* Suppose that  $N$  is an upper-bound of the number of nodes  $n$  known by all the stations. After one invocation of **TMPI**Ds( $N$ ), with high probability, every station of the network has a unique ID ranging from 1 to  $N^3$  and no pair of processors share the same ID.

*Proof:* The proof of this result is a simple application of the balls and bins problem. By throwing  $n$  balls (processors) into  $N^3$  bins (temporary IDs) independently and uniformly at random, with probability greater than  $\exp\left(-O\left(\frac{n^2}{N^3}\right)\right)$  every bin contains at most one ball. ■

### B. The two-hop neighbor discovery protocol

Once the temporary IDs attributed, each node  $u$  of the graph has to discover all the nodes at distance at most 2 hops from  $u$ . The protocol called **DISCOVER** below allows to know, for any given node  $u$  of the network, their respective direct and two-hop neighbors (i.e. neighbors of neighbors). This protocol appears to be extremely useful since the processors are deployed in random fashion and they do not have any knowledge of their respective neighborhoods. In the following pseudo-code,  $N$  stills represent any known upper-bound of the number of nodes  $n$ .

#### **Procedure DISCOVER( $N$ )**

##### **Begin**

For each node  $u$ , set  $L(u) := \emptyset$ ;

(★ ——— Discovering direct neighbors ——— ★)

**For**  $k$  from 1 to  $(\log N)^3$  **do**

With probability  $\frac{1}{\log N}$  each node  $u$

transmits a message containing its (temporary) ID ;

For each node  $u$ , upon receiving any message  $m$ , store its value in a local list :

$L(u) := L(u) \cup \{m\}$ ;

##### **EndDo**

(★ ——— Discovering 2-hop neighbors ——— ★)

**For**  $k$  from 1 to  $(\log N)^3$  **do**

With probability  $\frac{1}{\log N}$  each node  $u$  sends

its list  $L(u)$  of its known direct neighbors;

**EndDo**

**End.**

*Theorem 8:* Assume that the network is typical and the transmission range is set to  $r = \sqrt{\frac{2 \log(2^{p_0})|X|}{(2^{p_0})\pi}}$  with  $p_0$  satisfying (12). The running time of **DISCOVER**( $2^{p_0+1}$ ) is  $O(\log(n)^3)$  and with high probability, after one invocation of **DISCOVER**( $2^{p_0+1}$ ) :

- (i) Every node  $u$  of the network is aware of the list of all its direct and two-hop neighbors.
- (ii) For each node  $u$ , the number of such direct and two-hop neighbors is  $O(\log(n))$ .

*Proof:* The proof of Theorem 8 part (i) is closely related to the one of [32, Theorem 7]. For sake of clarity, here are the details. After the first loop of the above algorithm, the proof that every station is aware of the list of its neighbors relies on two facts, viz., **(1)** the main characteristics of the random Euclidean network and **(2)** the number of iterations  $O(\log(n))^3$  in this loop is sufficient for the nodes to send its ID *at least once* to all its neighbors. For the first point **(1)**, we have seen that for any node  $v$  of the network, if the transmission range is set to  $\sqrt{(1+\ell) \log n |X| / \pi n}$  ( $\ell > 0$ ) then the degree of  $v$  satisfies w.h.p. :

$$d_v \leq -\frac{\ell \log n}{W_0\left(-\frac{\ell}{e(1+\ell)}\right)} + o(\log n).$$

By setting  $N = 2^{p_0+1} \geq n$ , at the regime considered in the hypothesis of Theorem 8 the maximum degree of the graph is (w.h.p.) bounded by  $c \log n$  (where  $c$  is some constant satisfying e.g.  $c \geq 2 W_0(-1/2e)$  with any constant  $\ell > 2$ ). Using this latter remark, let us complete the proof of our Theorem part (i). For any distinct pair  $(i, j)$  of adjacent nodes and any time slot  $t \in [1, \log(N)^3]$ , define the random variable  $X_{i \rightarrow j}^{(t)}$  as follows :

$$X_{i \rightarrow j}^{(t)} = \begin{cases} \bullet & 1 \text{ if and only if the node } j \text{ does not receive the ID of } i \\ & \text{at time } t \in [1, \log(N)^3], \\ \bullet & 0 \text{ otherwise.} \end{cases} \quad (14)$$

In other terms, the set

$$\left\{ X_{i \rightarrow j}^{(t)}, i, j \neq i, t \in [1, \log(N)^3] \right\}. \quad (15)$$

denotes a set of random variables that counts the number of ‘‘arcs’’  $i \rightarrow j$  such that  $j$  has

never received the ID of  $i$ . Denote by  $X$  the r.v.

$$X = \sum_{i \neq j} X_{i \rightarrow j} \quad (16)$$

where  $X_{i \rightarrow j} = 1$  iff  $X_{i \rightarrow j}^{(t)} = 1$  for all  $t \in [1, \log(N)^3]$ . Now, we have the probability that  $i$  does not succeed to send its ID to  $j$  at time  $t$  :

$$\mathbb{P}\left[X_{i \rightarrow j}^{(t)} = 1\right] = \left(1 - \frac{1}{\log(N)}\right) + \frac{1}{\log N} \times \left(1 - \left(1 - \frac{1}{\log N}\right)^{d_j}\right). \quad (17)$$

Therefore, considering the whole range  $[1, \log(N)^3]$ , after a bit of algebra we obtain

$$\mathbb{P}[X_{i \rightarrow j} = 1] \leq \left(1 - O\left(\frac{1}{\log(n)}\right)\right)^{\log(N)^3} \leq \exp(-O(\log n)^2) \quad (18)$$

which bounds the probability that  $i$  has never sent its ID to  $j$  for time slots  $t$  in the range  $[1, \log(N)^3]$ . By linearity of expectations and since the number of edges is of order  $O(n \log n)$ , we then have

$$\mathbb{E}[X] \leq O(n \log(n)) \exp(-O(\log(n))^2). \quad (19)$$

Thus,  $\mathbb{E}[X] \ll 1$  as  $n \rightarrow \infty$  and using the first moment method [3], one completes the proof that after the first loop of the procedure, every station is aware of all its direct neighbors.

Using the same methods, it is easily seen that the second loop allows the nodes to know one after the other their 2-hop neighbors.

To prove **(ii)**, observe that if  $r$  is the common transmission/receiving range of the stations, the two-hop neighbors of a node  $u$  are inside a circle of radius  $2 \times r$ . Therefore, a simple application of the equation (6) in Theorem 2 permits us to deduce the assertion **(ii)** in Theorem 8. ■

### C. A two-hop coloring algorithm

We need some more basic definitions for our coloring algorithms :

*Definition 3:*

- $\Gamma(u) \stackrel{\text{def}}{=} \{\text{neighbors of a fixed node } u\}$ . Any  $v \in \Gamma(u)$  is referred to as *direct neighbors* of  $u$ .

- The set of *2-hop neighbors* is given formally by :  $\Gamma_2(u) \stackrel{\text{def}}{=} \bigcup_{v \in \Gamma(u)} \Gamma(v)$ .
- Recall that  $\Delta \stackrel{\text{def}}{=} \max_u |\Gamma(u)|$ . Similarly, define  $\Delta_2$  as  $\Delta_2 \stackrel{\text{def}}{=} \max_u |\Gamma_2(u)|$ .

To assign codes (colors) to the nodes of the network, let us consider the following simple and intuitive randomized protocol called **ASSIGNCOLOR**. As defined above,  $\Gamma(u) \cup \Gamma_2(u)$  is the set of neighbors of  $u$  at hop-distance at most 2. At the beginning of the algorithm, each vertex  $u$  possesses an initial list of colors  $p(u)$  (also referred to as *palette*) of size  $|\Gamma(u)| + |\Gamma_2(u)| + 1 = \Delta + \Delta_2 + 1$  and starts uncolored. We can assume that each node has a distinct ID (in fact, this can be effective after one invocation of **TMPIDS**) and knows its neighbors in  $\Gamma(u) \cup \Gamma_2(u)$  (by means of **DISCOVER**). Then, the protocol **ASSIGNCOLOR** proceeds in rounds. In each round, each *uncolored vertex*  $u$ , simultaneously and randomly independently picks a color, say,  $c$  from its palette. Next, the station  $u$  attempts to send this information to his direct neighbors  $\in \Gamma(u)$  and in their turn each member  $v \in \Gamma(u)$  tries to forward the information to every  $w \in \Gamma_2(u)$ . Trivially, this “two-steps” attempt succeeds iff there is no collision with the direct neighbors and also “no collision” with the 2-hop neighbors. Therefore, before attributing the color  $c$  definitely to  $u$ , every member of the set  $\Gamma(u) \cup \Gamma_2(u)$  has to sent one by one a message of reception. Note that this can be done *deterministically* as explained in great details below. Therefore,  $u$  sends a message of *confirmation* and every member  $v$  of  $\Gamma(u) \cup \Gamma_2(u)$  undergoes an *update* of its proper palette  $p(u)$  and of its own set  $\Gamma(u) \cup \Gamma_2(u)$ . Hence, at the end of such an iteration the new colored vertex  $u$  becomes passive during the rest of the protocol. Note that the protocol **ASSIGNCOLOR** is just the “2-hop version” of the coloring algorithm presented in [32, Paragraph 5.3]. Assuming that the upper-bound  $N$  of the number of nodes satisfies (12), the brief description of this procedure follows. Each of the steps below represents the *basic iteration* of the main loop of the algorithm. By allowing  $O(\log(n))^3$  iterations, the algorithm is shown to color correctly the graph.

**Basic iteration of the main-loop of ASSIGNCOLOR :**

**Step 0 :** Every node needs an initial palette of colors of size  $|\Gamma(u)| + |\Gamma_2(u)| + 1$  and a set of active neighbors and 2-hop neighbors. The upper-bound of the number of direct neighbors is set to  $\Delta := 3 \lceil \log_2(N) \rceil$ . Similarly, using formula (6) in Theorem 2, an upper-bound of the number of 2-hop neighbors is set to  $\Delta_2 := \lceil 8 \log(N) \rceil - \Delta$ . Note that the constant 8

reflects the fact that all 2-hop neighbors of  $u$  are within Euclidean distance at most twice the transmission range from  $u$ . Therefore, using (6) it yields  $-\frac{3}{W_0(-3e^{-1/4})} \sim 7.14... < 8$ .

**Step 1 :** Every node  $u$  picks a color  $c$  from its palette and tries to send it to  $\Gamma(u)$ .

**Step 2 :** If the previous step succeeds, there are no collision and every node  $v \in \Gamma(u)$  receives correctly the message. Since **DISCOVER** allows  $u$  to know its neighbors,  $u$  can rank them and in their turn, one after the other accordingly to their relative rank, they have to forward the message to the 2-hop neighbors of  $u$ , that is to any  $w \in \Gamma_2(u)$ . This phase is deterministic and is synchronized using  $\Delta$  time-slots.

**Step 3 :** If the previous step works correctly, every member of  $\Gamma_2(u)$  received the message and in their turn they have to send it back. In order to avoid confusion, the message are specifically marked with  $u$  (the ID of the original sender). This step can be done deterministically since  $u$  can also rank its 2-hop neighbors. Thus, step 3 needs also  $\Delta_2$  rounds.

**Step 4 :** When all their messages are back, all the nodes  $v \in \Gamma(u)$  need to advert  $u$ , one by one and in order. Therefore, step 4 is done in  $\Delta$  time-slots.

**Step 5 :** Upon receiving **all** the messages from all of its direct and 2-hop neighbors, the node  $u$  has to send back a message of **confirmation** to them. This step is done again deterministically and needs  $\Delta$  time-slots (only the direct neighbors are needed to forward the confirmation message). The nodes in  $\Gamma(u) \cup \Gamma_2(u)$  update their palettes of colors by removing the color  $c$  which is now attributed to  $u$ .  $u$  becomes a *passive* node.

The corresponding pseudo-code of the algorithm is given in the following:

- ( 1) **Protocol ASSIGNCOLOR**( $N$ )
- ( 2)     Set  $\Delta := 3 \lceil \log_2(N) \rceil$  and  $\Delta_2 := \lceil 8 \log(N) \rceil - \Delta$ ;     (★ following eq. (12) ★)
- ( 3)     Each vertex  $u$  is *active* and has an initial *palette* of colors, say  $p(u) = \{c_1, c_2, \dots, c_{\Delta+\Delta_2+1}\}$  along with a set of active neighbors  $\in \Gamma(u)$  and 2-hop neighbors  $\in \Gamma_2(u)$  ;
- ( 4)     **For**  $i := 1$  to  $\log(N)^3$  **Do**
- ( 5)         For each vertex  $u$  do
- ( 6)             • Pick a color  $c$  from  $p(u)$  ;
- ( 7)             • Send a message containing  $c$  **with probability**  $\frac{1}{\Delta+|p(u)|}$  ;
- ( 8)     **If** no collision **Then**     (★ 1-hop neighbors  $\implies$  forward to  $w \in \Gamma_2(u)$ ★)
- ( 9)         Every station  $v$  in  $\Gamma(u)$  gets the message properly ;
- (10)         One by one and in order (since they are ranked by  $u$ ) every member  $v$

- (11) of  $\Gamma(u)$  sends a specific message, say “forward  $v u c$ ”;
- (12) (★ where  $v$  is the ID of the current node. Observe that this step
- (13) can be synchronized by always allowing  $\Delta$  time slots. ★)
- (14) **EndIf**
- (15) **Upon** receiving a message of the form “forward  $v u c$ ” **Do**
- (16) (★ 2-hop neighbors  $\rightarrow$  just send back twice ★)
- (17) Every member  $w$  of  $\Gamma_2(u)$ ,
- (18) one by one and in order sends back a message intended to
- (19) the member of  $\Gamma(u)$ . Such message can be of the form “back  $w u c$ ” ;
- (20) (★ This step can be synchronized by always allowing  $\Delta_2$  rounds ★)
- (21) **end**
- (22) **Upon** receiving a message of the form “back  $w u c$ ” **Do**
- (23) The node  $v \in \Gamma(u)$  sends the message back to  $u$  along with its own ID;
- (24) (★ This step needs  $\Delta$  time-slots of synchronization ★)
- (24) **end**
- (25) **If**  $u$  receives all the  $|\Gamma_2(u)| + |\Gamma(u)|$  messages **Then**
- (26)  $u$  sends a message of **confirmation** which is also forwarded
- (27) by all members of  $\Gamma(u)$  to the set  $\Gamma_2(u)$ ;
- (28)  $u$  becomes *passive*;
- (29) **EndIf**
- (30) **Upon** receiving a **confirmation message**:
- (31) every station in  $\Gamma(u) \cup \Gamma_2(u)$  removes the color  $c$  from its palette ;
- (32) (★ This step is synchronized using  $\Delta_2$  time slots ★)
- (32) **EndFor**
- (33) **End.**

*Theorem 9:* Assume that the random deployed network is typical with the transmission range set to  $r = \sqrt{\frac{2 \log(2^{p_0}) |X|}{2^{p_0} \pi}}$ . Suppose also that the nodes have distinct IDs. Then, after the execution of **ASSIGNCOLOR**( $N$ ), with probability tending to 1 as  $n \rightarrow \infty$ , every pair of nodes  $(u, v)$  s.t.  $u \in \Gamma(v) \cup \Gamma_2(v)$  have received two distinct codes (colors). Moreover, the running time of the **ASSIGNCOLOR** is  $O(\log(n)^4)$  time-slots and it uses  $O(\log(n))$  colors.

*Proof:* Although more complicated, the proof of Theorem 9 is very similar to the one of [32, Theorem 8]. Observe first that the only randomized part of the algorithm is the

temptative of  $u$  to allocate a color (cf. line 7). After this, all the steps of each iteration are deterministic. Therefore, whenever successful such attempt can be easily checked by the initiator  $u$  since  $u$  possesses the list of nodes in  $\Gamma(u)$  and in  $\Gamma_2(u)$ . Concretely, the algorithm builds a new graph in which each new edge is (virtually) added between every pair of 2-hop neighbors.

For any distinct node  $u$ , recall that  $\Gamma(u) \cup \Gamma_2(u)$  represents the set of its direct and 2-hop neighbors and denote by  $p_u$  the size of its current palette. Now, define the random variable  $Y_u$  as follows:

$$Y_u = \begin{cases} \bullet & 1 \text{ if and only if the node } u, \\ & \text{remains uncolored after the } \log N^3 \text{ steps of } \mathbf{ASSIGNCOLOR} \\ \bullet & 0 \text{ otherwise.} \end{cases} \quad (20)$$

Denote by  $\Gamma_u^{(t)}$  (resp.  $\Gamma_{u,2}^{(t)}$ ) the set of *active direct neighbors* (resp. *2-hop neighbors*) of  $u$  at any given iteration  $t$  of the algorithm. Suppose that we are in such iteration  $t$ . Independently of its previous attempts,  $u$  remains uncolored with probability

$$p_{u,t} = \left(1 - \frac{1}{(\Delta + p_u)}\right) + \frac{1}{(\Delta + p_u)} \times \left(1 - \left(1 - \frac{1}{(\Delta + p_v)}\right)^{|\Gamma_u^{(t)}| + |\Gamma_{u,2}^{(t)}|}\right), \quad (21)$$

where we used the fact that there is at least a *direct collision* due to one neighbor  $v \in \Gamma_u^{(t)}$  or a “*2-hop collision*” with one neighbor  $w \in \Gamma_{u,2}^{(t)}$ .

We have  $\forall t$ ,  $|\Gamma_u^{(t)}| \leq \Delta$ ,  $|\Gamma_{u,2}^{(t)}| \leq \Delta_2$  and  $\forall v$ ,  $1 \leq |p_v| \leq \Delta + \Delta_2 + 1$ . More importantly  $\Delta = O(\log n)$  and  $\Delta_2 = O(\log n)$ . Therefore,

$$p_{u,t} \leq 1 - \frac{1}{(\Delta + p_u)} \left(\frac{1}{\Delta}\right)^{|\Gamma_u^{(t)}| + |\Gamma_{u,2}^{(t)}|} \leq 1 - O\left(\frac{1}{\log n}\right). \quad (22)$$

Since there are  $O(\log n^3)$  iterations, there exists a constant  $\alpha$  such that with probability at most

$$\left(1 - O\left(\frac{1}{\log n}\right)\right)^{O(\log n)^3} \leq \exp(-\alpha \log n^2) \quad (23)$$

$u$  remains uncolored during the whole algorithm. Thus, the expected number of uncolored vertices at the end of the protocol **ASSIGNCOLOR** is less than

$$\mathbb{E}[Y] = \sum_u \mathbb{E}[Y_u] \leq n \exp(-O(\log(n)^2)). \quad (24)$$

After using the well known Markov's inequality (cf. [3]), the proof of our Theorem is now done. ■

#### *D. Gossiping algorithm*

The 2-hop coloring process induces a natural scheduling algorithm for the gossiping task. The gossiping protocol is very intuitive: once the graph is colored at every time-slot each node  $u$  is allowed to transmit iff its attributed color  $c(u)$  satisfies  $[\mathbf{TIME} \bmod c(u)] \equiv 0$  ( $\mathbf{TIME}$  is a function that returns the current global time-slot). In the procedure **GOSSIP** below, we start with the randomly deployed nodes and use all the procedures described previously:

#### **Procedure GOSSIP**

**Step 1:** Start estimating the main characteristics of the network:  $\mathbf{SFR}\left(\frac{1}{|X|}\right)$  ;

Such procedure permits all the nodes to have estimates of the transmission range  $r$ , the maximal degree  $\Delta$ , the hop diameter  $D$  and the number of active nodes  $n$ ;

**Step 2:** Attribute temporary IDs to the nodes:  $\mathbf{TMPIDS}(N)$ ;

( $\star N$  represents a probabilistic upper-bound of  $n \star$ )

**Step 3:** Color the graph:  $\mathbf{ASSIGNCOLOR}(N)$ ;

**Step 4:** Use the obtained color as follows:

**Repeat**  $100 \times \sqrt{\frac{N}{\log N}}$  **times**

For each node  $u$  of color  $c(u)$ :

**If**  $\mathbf{TIME} \bmod c(u) = 0$  **Then**

Transmit all known IDs;

**EndIf**

**EndRepeat**

**End.**

Since,  $\mathbf{ASSIGNCOLOR}$  attributes  $O(\log n)$  distinct colors and the hop-diameter of the graph is given  $D = O(\sqrt{n/\log n})$ , we then have the following immediate but important result:

*Theorem 10:* If the random plane network is typical, then the procedure **GOSSIP** works

in time  $O(\sqrt{n \log n})$  and for every pair of nodes  $(u, v)$ , with high probability,  $u$  has received the temporary ID of  $v$ .

*E. Initializing in time  $O(D\Delta)$*

The initialization protocol is a direct consequence of the above algorithm:

**Procedure** INITIALIZATION( $N$ )

**GOSSIP**;

  For each node  $u$ , sort all received messages;

  ID( $u$ ) := rank of  $u$  in the sorted array of temporary IDs;

**End.**

*Theorem 11:* With high probability, the procedure INITIALIZATION( $|X|$ ) gives to each of the random deployed nodes a unique identity ranging from 1 to  $n$  in  $O(\sqrt{n \log n})$  time-slots.

*Corollary 12:* The gossiping and initialization protocols above are asymptotically optimal.

*Proof:* This is an immediate consequence of the results of Kushilevitz and Mansour in [21]. Since gossiping and initializing are harder than broadcasting which requires  $\Omega(D \log(n/D))$  on graphs such as random plane networks, gossiping and initializing require at least the same time-slots. Fortunately for our graphs (w.h.p),  $D\Delta$  and  $D \log(n/D)$  are of the same order of magnitudes. ■

Note that all the results above remain valid with the  $O(\sqrt{n \log n})$  replaced by  $O(n^{1/3}(\log n)^{2/3})$  if the nodes are deployed inside a cube (instead of a square).

## V. CONCLUSION

We showed that given a randomly distributed wireless nodes with density  $n/|X|$ , when the transmission range of the nodes is set to  $r = \sqrt{(1 + \ell) \frac{\log n |X|}{\pi n}}$ : (i) the hop-diameter is  $O\left(\sqrt{\frac{n}{\log n}}\right)$ , (ii) the network is  $\Theta(\log n)$ -connected, each point of the support area is monitored by  $\Theta(\log n)$  nodes and the degrees of all nodes are  $\Theta(\log n)$ , with high probability. We showed how these results can help to conduct precise analyses in order to design protocols for the self-configuration of the network. In our settings, the nodes assume only

as *a priori* knowledge of the nodes the size (or volume) of the support  $X$ . These results illustrate how fundamental limits of random networks can help researchers and developers for the design of algorithms in the extremal scenarios and the protocols given in this paper can serve as basis for other decentralized and localized algorithms. For instance, our gossiping algorithm can be used to compute the average temperature of an area recovered by sensors.

## REFERENCES

- [1] Abramowitz and Stegun, *Handbook of Mathematical Functions*. Dover Publications, 1974.
- [2] Akyildiz, I. F., Su, W., Sankarasubramanian, Y. and Cayirci, E. Wireless sensor networks: a survey. *Computer Networks* 38: 393–422, 2002.
- [3] Alon, N., Bar-Noy, A., Linial, N. and Peleg, D. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43: 290–298, 1991.
- [4] Bar-Yehuda, R., Goldreich, O. and Itai, A. Efficient Emulation of Single-Hop Radio Network with Collision Detection on Multi-Hop Radio Network with no Collision Detection. *Distributed Computing*, 5: 67–71, 1991.
- [5] 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 Comp. and Sys. Sciences*, 45: 104–126, 1992.
- [6] Cayley, A. A Theorem on Trees. *Quart. J. Math. Oxford Ser.*, 23: 376–378, 1889.
- [7] Cheng, Y.-C. and Robertazzi T. G. Critical connectivity phenomena in multihop radio models. *IEEE Trans. on Communications*, 36: 770–777, 1989.
- [8] Chlebus, B. Randomized Communication in Radio Networks Chapter in "Handbook of Randomized Computing," Panos M. Pardalos, Sanguthevar Rajasekaran, John H. Reif, and Jose D.P. Rolim (Eds.), Kluwer Academic, New York, 2001, vol. I, pp. 401–456.
- [9] Chrobak, M., Gasieniec, L. and Rytter, W. Fast Broadcasting and Gossiping in Radio Networks. *Proc. IEEE F. of Comp. Sci. (FOCS)*, 2000.
- [10] Corless, R. M., Gonnet G. H., Hare D. E. G., Jeffrey D. J. and Knuth D. E. On the Lambert W Function. *Advances in Computational Mathematics*, 5: 329–359, 1996.
- [11] De Bruijn, N. G. *Asymptotic Methods in Analysis*. Dover, New-York, 1981.
- [12] Erdős, P. and Rényi A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:17–61, 1960.
- [13] Flajolet, P. and Sedgewick, R. *Analytic Combinatorics*. Book in preparation. Chapters are available as INRIA research reports. See <http://algo.inria.fr/flajolet/books>.
- [14] Gilbert, E. N. Random Plane Networks. *Journal of the Society for Industrial and Applied Math*, 9: 533–543, 1961.
- [15] Gupta, P. and Kumar P. R. Critical power for asymptotic connectivity in wireless networks. *Stochastic Analysis, Control, Optimization and Applications: a volume in honor of W. H. Fleming, W. M. McEneaney, G. Yin and Q. Zhang*, Birkhauser, Boston, 1998.
- [16] Hall, P. *Introduction to the Theory of Coverage Processes*. Birkhäuser, Boston, 1988.
- [17] Hayashi, T., Nakano, K. and Olariu, S. *Randomized Initialization Protocols for Packet Radio Networks*, in

- S. Rajasekaran, P. Pardalos, B. Badrinath, and F. Hsu, Eds., Discrete Mathematics and Theoretical Computer Science, SIAM Press 2000, 221–235.
- [18] Janson, S., Knuth, D. E., Luczak, T. and Pittel B. The birth of the giant component. *Random Structures & Algorithms*, 4:233–358, 1993.
- [19] Jung, E-S and Vaidya N. A Power Control MAC Protocol for Ad Hoc Networks. *Proc. of ACM Mobicom'02*, pp. 36–47, 2002.
- [20] Knuth, D. E. *The Art of Computer Programming – Sorting and Searching, vol 3*. Addison-Wesley, 1973
- [21] Kushilevitz, E. and Mansour, Y. An  $\Omega(D \log(N/D))$  Lower Bound for Broadcast in Radio Networks SIAM Journal on Computing, Vol. 27, 702 – 712, 1998.
- [22] Liu, D. and Prabhakaran, M. *On Randomized Broadcasting and Gossiping in Radio Networks* in Proceedings of COCOON'02, Lecture Notes in Computer Sciences, Vol. 2387, 340–349, 2002.
- [23] Meester, R. and Roy, R. *Continuum Percolation*. Cambridge University Press, Cambridge, 1996.
- [24] Miles, R. E. On the Homogenous Planar Poisson Point Process. *Math. Biosciences*, 6: 85–127, 1970.
- [25] Nakano, K. and Olariu, S. Randomized Initialization Protocols for Ad-hoc Networks. *IEEE Transactions on Parallel and Distributed Systems* 11: 749–759, 2000.
- [26] Nakano, K. and Olariu, S. Energy-Efficient Initialization Protocols for Radio Networks with no Collision Detection. *IEEE Transactions on Parallel and Distributed Systems* 11: 851–863, 2000.
- [27] Penrose, M. D. The longest edge of the random minimal spanning tree. *Annals of Applied Probability*, 7: 340–361, 1997.
- [28] Penrose, M. D. A strong law for the largest nearest-neighbour link between random points. *Journal of the London Mathematical Society*, 60: 951–960, 1999.
- [29] Penrose, M. D. On  $k$ -connectivity for a geometric random graph. *Random Structures & Algorithms*, 15: 145–164, 1999.
- [30] Penrose, M. D. *Random Geometric Graphs*. Oxford Studies in Probability, 2003.
- [31] Perkins, C. E. *Ad Hoc Networking*. Addison-Wesley, 2001.
- [32] Ravelomanana, V. Extremal Properties of Three Dimensional Sensor Networks with Applications. *IEEE Trans. on Mobile Computing*, 3: 246–257, 2004.
- [33] Santalo, L. *Integral Geometry and Geometric Probability*. Cambridge University Press, 2nd edition, 2003.
- [34] Santi, P. and Blough D. M. The Critical Transmitting Range for Connectivity in Sparse Wireless Ad Hoc Networks. *IEEE Trans. Mob. Comp.*, 2: 1–15, 2003.
- [35] Shakkottai, S., Srikant, R. and Shroff N., Unreliable Sensor Grids: Coverage, Connectivity and Diameter. *to appear in Ad hoc Networks journal, 2004*. Previous version was published in the Proceedings of IEEE INFOCOM 2003.
- [36] Xue, F. and Kumar, P. R. The number of neighbors needed for connectivity of wireless networks. *Wireless Networks*, 10: 169 – 181, 2004.