Adv. Appl. Prob. **42,** 559–576 (2010)
Printed in Northern Ireland
© *Applied Probability Trust* 2010

LOW DEGREE CONNECTIVITY OF AD-HOC NETWORKS VIA PERCOLATION

EMILIO DE SANTIS,[*]
FABRIZIO GRANDONI [**] AND
ALESSANDRO PANCONESI,[***] *Sapienza University of Rome*

Abstract

Consider the following classical problem in ad-hoc networks. Suppose that n devices are distributed uniformly at random in a given region. Each device is allowed to choose its own transmission radius, and two devices can communicate if and only if they are within the transmission radius of each other. The aim is to (quickly) establish a connected network of low average and maximum degree. In this paper we present the first efficient distributed protocols that, in poly-logarithmically many rounds and with high probability, set up a connected network with $O(1)$ average degree and $O(\log n)$ maximum degree. Our algorithms are based on the following result, which is a nontrivial consequence of classical percolation theory. Suppose that each device sets up its transmission radius in order to reach the K closest devices. There exists a universal constant K (independent of n) such that, with high probability, there will be a unique giant component (i.e. a connected component of size $\Theta(n)$). Furthermore, all remaining components will be of size $O(\log^2 n)$. This leads to an efficient distributed probabilistic test for membership in the giant component, which can be used in a second phase to achieve full connectivity.

Keywords: Networks; connectivity; percolation

2010 Mathematics Subject Classification: Primary 68M10; 82B43
Secondary 60F10

1. Introduction

In this paper we study a geometric random graph model that has interesting applications to wireless networking. We suppose that n points distributed uniformly at random within the unit square are given. Each point v is connected via a directed arc to the *closest* $k(v)$ points, according to the Euclidean distance, where $k(v)$ is a positive integer value. Given this directed graph, we define an undirected graph G with the same vertex set as follows: $vw \in E(G)$, i.e. vw is an edge of G, if and only if there is a directed arc from v to w and vice versa. Henceforth, we will also refer to the points as *nodes* or *devices*.

The question that we study in this paper is how to determine the values of the $k(v)$s in order to meet two conflicting goals: G should be connected, but its average degree should be as small as possible. Moreover, the maximum degree should also be small.

Received 4 September 2006; revision received 10 June 2009.

A preliminary version of this work was published in *Algorithms – ESA 2007* (Proc. 15th Annual Europ. Symp. Algorithms, 2007; Lecture Notes Comput. Sci. **4698**), Springer, Berlin, pp. 206–217.

[*] Postal address: Department of Mathematics, Sapienza University of Rome, Piazzale Aldo Moro 2, 00185 Rome, Italy. Email address: desantis@mat.uniroma1.it

[**] Current address: DISP, Tor Vergata University of Rome, Via del Politecnico 1, 00191 Rome, Italy.
Email address: grandoni@disp.uniroma2.it
This work was done when the author was at the Department of Computer Science of Sapienza University.

[***] Postal address: Department of Computer Science, Sapienza University of Rome, Via Salaria 113, 00198 Rome, Italy. Email address: ale@di.uniroma1.it

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

More precisely, we think of the points as wireless devices capable of setting their own transmission range (by modifying their own transmission power) and able to communicate along the edges of the graph G. We are looking for efficient distributed algorithms to decide the values $k(v)$. In this paper we give two such algorithms that, with high probability, set up a connected network G and, moreover, ensure that (a) the expected degree of a node is constant and (b) the maximum degree is $O(\log n)$ or $O(\log^2 n)$. The numbers of communication rounds required for our algorithms are $O(\log^3 n)$ and $O(\log^2 n)$. We adopt the standard definitions of the distributed computing literature, spelled out in subsequent sections.

These results appear to be relevant to wireless networking. Our model of connectivity for G is the most realistic from the point of view of wireless applications since the communication primitives of standards such as IEEE 802.11 and Bluetooth rely on ack (as in acknowledgment) messages, and, therefore, a communication link really exists only when both nodes are within the transmission radius of each other. Limiting the degree of nodes can be beneficial in many ways. For instance, in security applications, nodes exchange keys and run cryptographic protocols with their neighbors (see, for instance, [12]). Limiting the degree reduces the amount of traffic and computation. Moreover, the transmission radius of v is set in order to reach its $k(v)$ closest neighbors. Hence, limiting the $k(v)$s reduces the overall transmission power and translates into longer network lifetimes. In particular, we can show that the expected power consumption required to sustain the network generated by our algorithms is of the order of the area. This is in some sense optimal, since the optimal power consumption is, with high probability, proportional to the area of the region within which the nodes are randomly distributed [2]. Probably the most important benefit is that, by bounding $k(v)$ and by setting the transmission power accordingly, interference is kept under control: the smaller the degree of a node, the lower the number of neighbors affected by a transmission and, consequently, the lower the number of possible packet collisions and corresponding retransmissions (see, for instance, [1]). Note that our high probability bound on the maximum degree ensures that not only are things good on average, but also that no node will be penalized too much.

Let us now describe our algorithms. Probably the simplest distributed algorithm we can think of is the following: set beforehand $k(v) = K$ for all nodes v, where K is a suitable constant (see [5], [6], [9], and [10] for experimental results). Unfortunately, there is no constant K which guarantees connectivity with *high probability*, i.e. with probability going to 1 as n grows. To reach this goal, K must grow as $\log n$ [11].

If points can communicate, the situation changes. Indeed, Kucera [7] gave a protocol to decide $k(v)$ for all v that sets up a connected network of expected constant degree. The result however is existential in flavor: the protocol requires nodes to explore linear size components of the network, linearly many times, making it completely impractical. Our faster protocols are based on the following insight (see Theorem 2). There is a universal constant K, independent of n, such that, if all the devices set $k(v) = K$, with high probability, the network has the following special structure:

- there is a unique giant component containing $\Theta(n)$ nodes;
- all other components have size $O(\log^2 n)$.

This says that it is possible to set up a giant component in a very simple way, a useful fact by itself (e.g. for coverage applications). It also says that there is an efficient distributed test for membership to the giant component: a node belongs to the unique giant component if and only if it belongs to a component with more than (an order of) $\log^2 n$ nodes.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

Given this, the following strategy is very natural. Devices that discover to be trapped inside small components increase their transmitting power in order to reach a device that belongs to the giant component. A node in the giant component that is contacted in this way will respond, setting its power in order to reach the calling node. We will refer to this as Algorithm A, to be described in detail in the next section. Algorithm A sets up a network in which the expected number of neighbors of each device is (upper bounded by a) constant. Furthermore, with high probability, the network is connected and its maximum degree is $O(\log^2 n)$. The number of communication rounds required is $O(\log^2 n)$. This gives an exponential speed up with respect to [7], therefore turning an existential result into a usable algorithm.

We can improve the bound on the maximum degree at the expense of an increased communication cost. Suppose that each device v belonging to a small component increases its transmitting power a bit at a time, each time checking if it has reached a node in the giant component. Nodes closer to the giant component will join it first. Nodes farther away might be able to connect to such closer nodes, rather than expanding their radius all the way to the closest node in the original giant component. In the next section we will give a precise description of this, referred to as Algorithm B. Algorithm B sets up a network in such a way that the expected number of neighbors of each device is constant. Furthermore, with high probability, the network is connected and its maximum degree is $O(\log n)$. The number of communication rounds required is $O(\log^3 n)$.

Finally, we show that our algorithms set up energy-efficient networks, in the following sense: the expected transmission power needed to sustain the network is of the same order of magnitude as that of the expected optimal power assignment.

These bounds on the expected and maximum degrees, power consumption, and the communication costs make our algorithms likely candidates for efficient, real implementations.

2. The protocols

As is customary we will use the terms 'distributed algorithm' and 'protocol' as synonyms. The input to the protocols consists of n devices that are spread uniformly at random within the unit box. Recall that two devices are connected if each is within the transmission radius of the other. Two devices that are mutually within range are said to be *neighbors*. The neighbor relation defines a graph whose vertices are the devices and where edges connect neighbors.

We assume that this graph operates as a synchronous network, that is, computation proceeds in a sequence of discrete time units called *communication rounds*. In one communication round each device is able to send messages to and receive messages from all of its neighbors. The *running time* of a protocol is given by the number of communication rounds needed by the protocol to terminate.

We assume that the value of n is known to the devices. We also assume the following primitive. Each device v has a local variable $k(v)$. When $k(v)$ is increased by 1, this enables v to set its transmission radius in order to reach the next closest device, according to the Euclidean distance. We will also assume that incrementing $k(v)$ by 1 takes one communication round. We will discuss later how such a primitive can be realized in practice.

Algorithm A has two constant parameters K and φ, and works as follows.

Algorithm A. *Phase 1.* Every device v initially marks itself as *lacustrine* and then sets its own transmission radius in order to reach the closest $k(v) := K$ devices (all devices if $n < K$).

Phase 2. (i) Let $C(v)$ denote the connected component of device v. If $|C(v)| > \varphi \log^2 n$, v marks itself as *continental*.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

(ii) Every lacustrine device v increments $k(v)$ until $k(v) > \varphi \log^2 n$ or v reaches the next closest continental device, denoted as $s(v)$. In the latter case, device $s(v)$ responds by increasing its own transmission radius in order to reach v (if this is not already the case).

The goal of phase 2 for a node that fails the test '$|C(v)| > \varphi \log^2 n$' is to hook up to the closest continental device. The clause '$k(v) > \varphi \log^2 n$' is to make sure that the algorithm stops in all cases. Although it happens with negligible probability, without this test, the algorithm could increment $k(v)$ forever, without terminating. We now describe Algorithm B. Besides K and φ, Algorithm B has a third parameter $t > 0$, and works as follows.

Algorithm B. *Phase 1.* As in Algorithm A.

Phase 2. Repeat this phase $t \log n$ many times.

(i) Let $C(v)$ be the connected component of device v. If $|C(v)| > \varphi \log^2 n$ then v marks itself as *continental*.

(ii) Otherwise, v increments $k(v)$ by 1. If v reaches a continental node $s(v)$, $s(v)$ responds by increasing its own transmission radius in order to reach v (if this is not already the case).

2.1. Implementation issues and running time

In the algorithms there are two types of operation taking place. There are power control operations, such as those necessary to implement phase 1 and to increment $k(v)$ in phase 2, and communication rounds needed in phase 2 to explore the topology of the network.

The only task of the algorithms that requires communication is the test '$|C(v)| > \varphi \log^2 n$' in phase 2. This can be implemented with a breadth-first search in $O(\log^2 n)$ communication rounds.

For the sake of simplicity, we assume that the basic power control primitive for a device v is to increment $k(v)$ by 1, allowing v to reach the next closest device, and that this requires one round. This assumption is not really an issue for Algorithm A, whose power control operations occur before the test, in phase 1, and after it, when each lacustrine device increments its power, looking for the closest continental node (as long as $k(v) \leq \varphi \log^2 n$). The message conveyed by Algorithm A is that by paying a little extra cost in terms of communication, and without increasing the cost of the power control operations too much, it is possible to save significantly in terms of the power with respect to the baseline solution (i.e. set $K := \log n$) [11].

In phase 2 the power of every lacustrine device v is increased until the closest continental node $s(v)$ is reached. How does v know? In practice, this can be implemented by increasing vs power a little at a time, each time sending a beacon with some relevant information about the sender such as its ID and its transmission radius. With this information, a continental node that receives the beacon can adjust its own transmission radius, if necessary, and respond so that v and $s(v)$ will be aware of each other and become neighbors. Other schemes are also possible and used in practice.

The situation is slightly more problematic for Algorithm B, which, in phase 2, interleaves power control operations (the increment of $k(v)$) with graph exploration. Assuming that $k(v)$ can be incremented by 1 in one round hides several implementation details, but, again, here we are mostly concerned with whether certain goals are attainable in principle, leaving the problem of finding efficient implementations to further study. In fact, increasing the level of detail would only clutter the presentation without adding any new insights.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

The message of Algorithm B is that by paying somewhat more in terms of communication with respect to Algorithm A, we can set up a network whose maximum degree is $O(\log n)$ as opposed to $O(\log^2 n)$. This is good because, in general, a smaller degree means less power consumption, less interference, etc. Thus, even though the total power consumptions of the algorithms are comparable, Algorithm B imposes a smaller load on individual vertices. The algorithms therefore illustrate a fundamental trade-off between the cost of communication and the power needed to sustain a connected network.

With our assumptions and definitions, it is clear by inspecting the algorithms that their running times are respectively $O(\log^2 n)$ and $O(\log^3 n)$.

2.2. Correctness

The mapping $s(v)$ is, for all practical purposes, well defined since, almost surely, all pairwise distances are different. We will assume this from now on.

We say that an event occurs 'with high probability' if its probability is $1 - o(1)$, where $o(1)$ is a term that goes to 0 as n, the number of devices, goes to ∞. In the analysis we will show that the parameters of the algorithms can be chosen in such a way that the $o(1)$ term goes to 0 as any inverse polynomial. Note that in principle both algorithms might fail. However, we will show the following.

1. There is a constant K, independent of n, such that, at the end of phase 1, with high probability, there is a unique giant component.
2. With high probability, all other (nongiant) components contain $O(\log^2 n)$ many points.

The first fact is potentially useful by itself because a giant component is often good enough for the applications and this shows that it can be set up with constant K, i.e. with low power. Observe also that phase 1 does not require any global information, such as the value of n.

The two facts above imply that, with high probability, the algorithms are correct. Namely, we can make the following claims.

- The parameters K and φ can be chosen in such a way that, with high probability, Algorithm A sets up a connected network in which the expected degree of a node is bounded by a constant, and the maximum degree is $O(\log^2 n)$. The choice of the parameters is independent of n.
- The parameters K, φ, and t can be chosen in such a way that, with high probability, Algorithm B sets up a connected network in which the expected degree of a node is bounded by a constant and its maximum degree is $O(\log n)$. Again, the choice of the parameters is independent of n.

In the rest of the paper we will prove the claims above.

3. Overview

Since the proof is rather involved, we first give an overview. The basic idea is to reduce our connectivity problem to site percolation in a finite box consisting of $\Theta(n)$ sites. It is known that in the supercritical phase, with high probability, there is a unique giant cluster of $\Theta(n)$ sites in the box and that its complement consists of small regions, each containing $O(\log^2 n)$ sites (see, among others, [3] and [4, p. 214]). In the following we will refer to the maximal regions in the complement of the giant cluster as *lakes*. The reduction will ensure that the unique giant cluster in the box will correspond to a unique giant component of $\Theta(n)$ points, and that the remaining

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

components of points are trapped inside lakes, each containing $O(\log^2 n)$ points. This is the situation at the end of phase 1 (with high probability). Note that the maximum degree of this giant component is bounded by K, a constant (independent of n)—a useful fact by itself.

The reduction to site percolation is achieved via several intermediate steps. The first is to replace the uniform distribution of points with a Poisson distribution, to exploit the strong independence properties of the latter. In particular, unlike the uniform distribution, the Poisson distribution ensures that the configuration of points in one region does not affect the distribution of points of any other disjoint region. There are some standard and rather general ways to connect the two settings, but here we will make use of a coupling construction that gives stronger bounds than these general tools. The configurations of points given by the mentioned Poisson processes is referred to as scenario I.

We next introduce a first percolation problem, scenario II, by subdividing the unit square into a grid of nonoverlapping square cells (corresponding to the sites). The area of each cell is such that the expected number of points inside it is a constant parameter α. This parameter is crucial for the whole construction. A cell is *good* if the number of points that it contains is in $[\alpha/2, 2\alpha]$.

Scenario II is a Bernoulli field, but unfortunately clusters of good cells do not translate necessarily into connected components of points. Therefore, another percolation problem, scenario III, is introduced by defining a cell i *open* if it is good and, moreover, all cells within distance D from i good. The value of D is a constant, independent of n, whose value will be set in such a way that the points belonging to a cluster of open cells will form a component of points.

The problem with scenario III is that it is not a Bernoulli field—knowing that a cell C is open or closed alters the distribution of neighboring cells. However, the dependence only involves cells at a distance at most $h = 2D$ from C—the field is h*-dependent* (see [4, p. 179] and [8]). Therefore, a new scenario IV is introduced. Scenario IV is given by a general construction of [8]. This construction translates scenario III into a Bernoulli field (i.e. scenario IV) that is stochastically dominated by scenario III. This implies in particular that if a cell is open in scenario IV then it is also open in scenario III, and that if a giant cluster of open cells exists in scenario IV, the same cluster exists in scenario III. Thus, if a giant cluster exists with probability p in scenario IV then it also exists in scenario III with at least the same probability. In turn, the construction ensures that the unique giant component of sites that, with high probability, exists in scenario III, translates into a connected component of points in scenario I, and that all other components are small. The reason to introduce scenario IV is that this is a Bernoulli field and we can invoke results in the literature to give sharp probability estimates of the events of interest.

The probability that sites are on or off in the various scenarios depends on the value of the constant K of the protocol. We will fix K in such a way that a unique giant cluster of open cells exists in scenario IV with high probability. By construction, this translates into a giant component of points $\mathcal{G}$ in scenario I. To ensure that this giant component is unique in scenario I, we make use of the definition of open cells in scenario III which ensures that points trapped inside lakes cannot connect to points in other lakes, bypassing the giant component $\mathcal{G}$.

Remark 1. By setting the radius of each point to approximately $n^{-1/2}$ we would obtain a simpler reduction to site percolation to show the emergence of a giant component. Our reduction however is independent of n, showing that a giant component can be created with no global information at all.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

3.1. Preliminaries

Fact 1. *The rate function for a Poisson random variable X with mean μ is*

$$I(x) = x \ln\left(\frac{x}{\mu}\right) - (x - \mu).$$

Let us define $x = \alpha\mu$. Then we can write

$$I(\alpha\mu) = \mu[\alpha \ln(\alpha) - (\alpha - 1)] =: \mu C(\alpha),$$

where $C(\alpha) > 0$ when $\alpha \neq 1$. We will be using the following well-known facts about the Poisson random variable:

$$\Pr(X < \alpha\mu) \leq \mathrm{e}^{-C(\alpha)\mu} \tag{1}$$

for $\alpha \in (0, 1)$ and

$$\Pr(X > \alpha\mu) \leq \mathrm{e}^{-C(\alpha)\mu} \tag{2}$$

for $\alpha \geq 1$.

As mentioned, in scenarios II, III, and IV, we consider a partition of the unit square into a grid of nonoverlapping square cells of the same size. The number of cells is $m = k^2$, where $k := \lfloor\sqrt{n/\alpha}\rfloor$ and α is a constant. This partition naturally induces a mesh, whose sites are the cells and each cell (site) has (at most) four neighbors: the cells to the left, right, top, and bottom. Let $i_{x,y}$ be the cell in position (x, y) in the grid. The *distance* between i_{x_1,y_1} and i_{x_2,y_2} is $\max\{|x_1 - x_2|, |y_1 - y_2|\}$. The $*$-*neighbors* of cell i are the cells a distance 1 from i. We call a *cluster* a connected component of cells, and a $*$-*cluster* a connected component of cells with respect to the $*$-neighborhood. We will use this distance in the mesh, while we will use the Euclidean distance when talking about points in the unit square.

When a device v is inside the transmission radius of another device u, we say that u *captures* v. As remarked, if u captures v and vice versa, then the two devices can communicate directly and they become *neighbors* in the graph connected by the edge uv.

A *giant* cluster is a cluster of open cells which contains at least δm cells for a given constant $\delta \in (0, 1]$. Assuming a unique giant cluster (an event that we will show happening with high probability), a *lake* is a maximal $*$-cluster in the complement of the giant cluster. A *giant component* is a connected component of points (devices) of linear (in n) size in the network of devices defined by the neighbor relation. With $|X|$ we denote either the number of cells of X or the number of points of X, depending on whether X is a cluster or a component, respectively.

4. Emergence of a giant component

In this section we show that after phase 1 of the algorithm there is a giant component $\mathcal{G}$ containing $\Theta(n)$ points with high probability.

As outlined previously, we consider four different scenarios. We consider two Poisson processes, P_0 and P_t. Process P_0 has parameter $\mu_0 := n - \varepsilon n$, where ε is a small positive constant, say $\varepsilon = \frac{1}{4}$. Process P_t is built on top of P_0 by adding to it a new independent Poisson process ΔP with parameter $2\varepsilon n$. It is well known that P_t is a Poisson process with parameter $\mu_t := \mu_0 + 2\varepsilon n = n + \varepsilon n$. Then we define a sequence of point processes $\{Q_i\}$ sandwiched between P_0 and P_t. Starting from $Q_0 := P_0$, Q_{i+1} is given by Q_i by adding one point chosen uniformly at random in $P_t - Q_i$.

As we will see, our reduction to site percolation will apply simultaneously to all the Q_is, showing the existence of a unique giant component in scenario I for each Q_i with high

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

probability. Each Q_i generates points uniformly distributed in the box (conditioned on the given number of points). The next lemma shows that, with high probability, one of the Q_i will generate exactly n points. As a consequence, if something holds for all the Q_is of scenario I simultaneously, it also holds for the original n-points problem. We denote by N_0 and N_t the number of points of the Poisson processes P_0 and P_t, respectively.

Lemma 1. *Let N_0 and N_t be the Poisson variables relative to P_0 and P_t. There is a positive constant γ (independent of n) such that* $\Pr(\overline{\{N_0 \leq n \leq N_t\}}) \leq \mathrm{e}^{-\gamma n}$.

Proof. Apply the large deviation inequalities (1) and (2) respectively to the events $\{N_0 > n\}$ and $\{n < N_t\}$.

We now define scenario II. Let us subdivide the unit square into a grid of $m = k^2$ nonoverlapping square cells, where $k := \lfloor\sqrt{n/\alpha}\rfloor$ and α is a positive constant. Note that $m = \Theta(n)$ and that the expected number of points in a cell is (roughly) α. The parameter α plays a crucial role in the whole proof. This parameter should be thought of as a large constant. Its value will be fixed later.

Definition 1. A cell is *good* if the number of points in the cell given by both P_0 and P_t is in $[\alpha/2, 2\alpha]$. The cell is *bad* otherwise.

Note that if a cell is good then its number of points is in $[\alpha/2, 2\alpha]$ for all the Q_is of scenario I. Scenario II is a Bernoulli field because points are generated by Poisson processes, so a cell is good or bad independently of other cells, and the probability of being good is the same for all cells.

Lemma 2. *Let p_α be the probability that a cell is good. Then* $\lim_{\alpha\to\infty} p_\alpha = 1$.

Proof. Apply the standard large deviation principle to the Poisson random variables corresponding to the number of points given by P_0 and P_t in the cell considered.

We would like to show that large connected clusters of good cells in scenario II give rise to large connected components of points in scenario I. This however is not true. This motivates the next scenario, scenario III. In scenario III we consider another site percolation problem which is not, however, independent. Let $D \geq 3$ be a constant to be fixed later.

Definition 2. A cell i is *open* if i and all the cells at a distance at most D from i are good. The cell is *closed* otherwise.

We now choose K and D in order to enforce the following two properties. First, if we have a giant cluster of open cells in scenario III then the points inside these cells belong to a giant component $\mathcal{G}$ of scenario I for each Q_i. Second, components other than $\mathcal{G}$ will be trapped inside lakes (delimited by closed cells), i.e. points inside distinct lakes cannot establish edges among them directly, bypassing $\mathcal{G}$.

For simple geometric reasons, the first property is guaranteed by choosing

$$K := 7^2(2\alpha) = 98\alpha \tag{3}$$

(the maximum number of points in the cells at a distance at most 3 from a given open cell). This choice of K ensures that the transmission radius of every point in an open cell will reach all points in neighboring cells. Thus, if two neighboring cells are open, the points in them will form a clique. Thus, any point inside a cell at a distance at most $D - 3$ from an open cell i belongs to the same connected component to which the points in i belong. By choosing D such

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

that $2(D-3)\alpha/2 > K$, say $D = 102$, the second property is also ensured (see the proof of Lemma 5, below). We have not tried to optimize the values of D and K.

In scenario III we have the desired translation of connectivity—if we have a cluster of open cells then all points belonging to these cells lie in a connected component of points in scenario I (for all the Q_is).

Moreover, the probability q_α that a cell is open satisfies $q_\alpha \geq p_\alpha^{(2D+1)^2}$ and, thus, using Lemma 2, we obtain

$$\lim_{\alpha\to\infty} q_\alpha = 1.$$

Unfortunately, scenario III is not a Bernoulli field. Definition 2 however ensures that it is h-dependent with $h = 2D$ (the probability that a cell is open is independent of what happens in cells at a distance $2D+1$ or larger). Therefore, we introduce a fourth scenario, scenario IV, which is a Bernoulli field. The connection between scenarios III and IV is given by a very general theorem of [8]. The theorem states that there is a coupling between scenario III and a Bernoulli field, referred to as scenario IV, with site probability r_α such that

- if q_α goes to 1, so does r_α, and, hence, $\lim_{\alpha\to\infty} r_\alpha = 1$;
- if a cell is open in scenario IV, the same cell is open in scenario III.

Therefore, if we have a giant cluster in scenario IV, the same cells also form a cluster in scenario III. In turn, all points inside these cells will be connected in scenario I.

Scenario IV allows us to estimate the probability of relevant events. A result of Deuschel and Pisztora [3] on site percolation ensures that, for every constant $\delta \in (0, 1)$, there is a value $r_\alpha < 1$ such that, if every site is open with probability r_α then, with probability at least $1 - \mathrm{e}^{-\gamma\sqrt{m}}$, there is a unique giant cluster of at least δm open cells in scenario IV, for some constant $\gamma > 0$.

The following theorem and corollary summarize the discussion above.

Theorem 1. *Let $\mathcal{G}$ denote a connected component of points of maximum size at the end of phase 1 of Algorithms A and B. For every $c \in (0, \frac{1}{2})$, there is a choice of $\alpha > 0$, and so a corresponding choice of K by (3), such that*

$$\Pr(|\mathcal{G}| \leq cn) \leq 2\,\mathrm{e}^{-\xi\sqrt{n}}, \tag{4}$$

where $\xi > 0$ is a constant independent of n.

Proof. Let $m \simeq n/\alpha$ be the number of cells in scenario IV, and let $C^{(m)}$ be a maximum size cluster in scenario IV. By [3], for any given $\delta \in (0, 1)$, there is a value of α such that

$$\Pr(|C^{(m)}| \leq (1-\delta)m) \leq \mathrm{e}^{-\gamma\sqrt{m}}. \tag{5}$$

The same set of cells is open in scenario III. By definition, an open cell contains at least $\alpha/2$ points. Thus, $C^{(m)}$ corresponds to a giant component of at least $(1-\delta)m\alpha/2$ points in scenario I (for all the Q_is). Recalling Lemma 1 and by choosing $\delta < 1 - 2c$ in (5), we have

$$\Pr(|\mathcal{G}^{(n)}| \leq cn) \leq \Pr\biggl(|C^{(m)}| \leq \frac{2cn}{\alpha}\biggr) + \Pr(\overline{\{N_0 \leq n \leq N_t\}}) \leq 2\,\mathrm{e}^{-\xi\sqrt{n}}$$

for some constant $\xi > 0$ and large enough n. This follows from the fact that if the condition $\{N_0 \leq n \leq N_t\}$ does not hold then we give up, while we pursue the construction of the four scenarios only if the condition does hold.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

Remark 2. By choosing ε appropriately in the definition of the two Poisson processes P_0 and P_t of scenario I, and by defining a cell to be *good* if its number of points in both P_0 and P_t is in the interval $[(1-\varepsilon)\alpha, (1+\varepsilon)\alpha]$, the relative size of $\mathcal{G}$ can be any constant smaller than 1, for a proper choice of the parameter K in the algorithms.

The following corollary states the useful and interesting fact that, for the emergence of a giant component, the value of $k(v)$ can be set for every v to a constant K independent of the number of points n.

Corollary 1. *For every $c \in (0, 1)$, there exist constants $K > 0$ and $\gamma > 0$ such that, if every point v sets $k(v) = K$ then the probability that there is no connected component of size at least cn is at most $\mathrm{e}^{-\gamma\sqrt{n}}$.*

Proof. The proof follows from the proof of Theorem 1 and Remark 2.

5. Minuscule components

In this section we suppose that K is so large that the system of scenario IV is in the supercritical regime. Therefore, at the end of phase 1, there exists a giant component $\mathcal{G}$ with high probability. In this section we show that, at the end of phase 1, every connected component in scenario I distinct from the giant component $\mathcal{G}$ contains $O(\log^2 n)$ points, with high probability. As a consequence, $\mathcal{G}$ is the unique giant component (while the uniqueness of the giant cluster is already guaranteed by [3]).

The next lemma bounds the number of cells of a lake in scenario III.

Lemma 3. *For any lake L of scenario III,*

$$\Pr(|L| > k) \le \mathrm{e}^{-\gamma\sqrt{k}},$$

where $\gamma > 0$ is a constant.

Proof. It is well known that, for a Bernoulli field in the supercritical phase for percolation, if we take any $*$-cluster S in the complement of the giant cluster then

$$\Pr(|S| > k) \le \mathrm{e}^{-\gamma\sqrt{k}}$$

for some constant $\gamma > 0$ [4]. Therefore, the same holds for any lake of scenario IV. Now, by the monotonicity implied by the coupling construction of [8] that relates scenario III and scenario IV, the same bound holds for L in scenario III (lakes can only be smaller).

The next lemma is the key to analyzing the performance of the algorithms at the end of phase 2. The difficulty lies in the careful analysis of the dependencies—knowing that a cell is closed/open affects not only the distribution of points inside this cell, but also that of neighboring cells.

Lemma 4. *Let Z_i be the number of points in cell i, and let L be a lake in scenario III. Then, if the number of points n is large enough, there is a constant $\gamma > 0$ such that*

$$\Pr\biggl(\sum_{i\in L} Z_i > h\biggr) \le \mathrm{e}^{-\gamma\sqrt{h}}.$$

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

Proof. Let $B := (B_1, \dots, B_m)$ be the random vector denoting which cells are good or bad, and let $b = (b_1, \dots, b_m)$ for any particular such configuration. Then

$$\begin{aligned}
&\Pr\biggl(\sum_{i\in L} Z_i > h\biggr)\\
&\quad = \sum_k \Pr\biggl(\sum_{i\in L} Z_i > h \biggm| |L| = k\biggr)\Pr(|L| = k)\\
&\quad = \sum_k \sum_b \Pr\biggl(\sum_{i\in L} Z_i > h \biggm| |L| = k,\ B = b\biggr)\Pr(B = b \mid |L| = k)\Pr(|L| = k)\\
&\quad = \sum_k \sum_b \Pr\biggl(\sum_{i\in L} Z_i > h \biggm| B = b\biggr)\Pr(B = b \mid |L| = k)\Pr(|L| = k).
\end{aligned}$$

The last equality follows since, if we know B, we also know the size of L. We now focus on the term $\Pr(\sum_{i\in L} Z_i > h \mid B = b)$. We will show that we can replace the variables $(Z_i \mid B = b)$ with a set of independent and identically distributed variables that stochastically dominate them and that obey the large deviation principle.

The Poisson process can be realized as the product of m independent Poisson processes, each operating inside a cell. This implies that if we have a set of events E_i where each event depends only on what happens in cell i, then $\Pr(\bigcap_i E_i) = \prod_i \Pr(E_i)$. Thus, we have

$$\begin{aligned}
\Pr\biggl(\bigcap_i \{Z_i = h_i\} \biggm| B = b\biggr) &= \frac{\Pr(\bigcap_i \{Z_i = h_i,\ B_i = b_i\})}{\Pr(\bigcap_i \{B_i = b_i\})}\\
&= \frac{\prod_i \Pr(Z_i = h_i,\ B_i = b_i)}{\prod_i \Pr(B_i = b_i)}\\
&= \prod_i \Pr(Z_i = h_i \mid B_i = b_i).
\end{aligned}$$

If we define $X_i = (Z_i \mid B_i = \text{good})$ and $Y_i = (Z_i \mid B_i = \text{bad})$, it follows that $\sum_i (Z_i \mid B)$ is distributed like the sum of independent variables each of which is X_i or Y_i depending on whether cell i is good or bad. Let us define a collection of independent and identically distributed positive random variables W_i, each of which has the distribution of $(Z_i \mid Z_i > 2\alpha)$. Each W_i stochastically dominates both X_i and Y_i so that

$$\Pr\biggl(\sum_{i\in L} Z_i > h \biggm| B = b\biggr) \le \Pr\biggl(\sum_{i\in L} W_i > h\biggr)$$

for each configuration b. Moreover, the W_i obey the large deviation principle, i.e. the probability of large deviations from the mean is exponentially small. More precisely, we have

$$\Pr\biggl(\sum_{i\in L} W_i > h\biggr) \le \mathrm{e}^{-\gamma_1 h} \tag{6}$$

for a suitable constant $\gamma_1 > 0$ independent of h.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

We thus have, for $\beta < 1/\mathrm{E}[W_1]$,

$$\begin{aligned}\Pr\biggl(\sum_{i\in L} Z_i > h\biggr) &= \sum_k \sum_b \Pr\biggl(\sum_{i\in L} Z_i > h \biggm| B = b\biggr) \Pr(B = b \mid |L| = k) \Pr(|L| = k)\\ &\le \sum_k \sum_b \Pr\biggl(\sum_{i\le k} W_i > h\biggr) \Pr(B = b \mid |L| = k) \Pr(|L| = k)\\ &= \sum_k \Pr\biggl(\sum_{i\le k} W_i > h\biggr) \Pr(|L| = k)\\ &= \sum_{k\le\beta h} \Pr\biggl(\sum_{i\le k} W_i > h\biggr) \Pr(|L| = k) + \sum_{k>\beta h} \Pr\biggl(\sum_{i\le k} W_i > h\biggr) \Pr(|L| = k)\\ &\le \sum_{k\le\beta h} \Pr\biggl(\sum_{i\le\beta h} W_i > h\biggr) \Pr(|L| = k) + \sum_{k>\beta h} \Pr\biggl(\sum_{i\le k} W_i > h\biggr) \Pr(|L| = k)\\ &\le \sum_{k\le\beta h} \Pr\biggl(\sum_{i\le\beta h} W_i > h\biggr) + \sum_{k>\beta h} \Pr(|L| = k)\\ &= \beta h \Pr\biggl(\sum_{i\le\beta h} W_i > h\biggr) + \sum_{k>\beta h} \Pr(|L| = k).\end{aligned}$$

From Lemma 3, for a proper constant $\gamma_2 > 0$,

$$\Pr(|L| = k) \le \mathrm{e}^{-\gamma_2\sqrt{k}}.$$

Altogether,

$$\beta h \Pr\biggl(\sum_{i\le\beta h} W_i > h\biggr) + \sum_{k>\beta h} \Pr(|L| = k) \le \beta h\, \mathrm{e}^{-\gamma_1 h} + \sum_{k>\beta h} \mathrm{e}^{-\gamma_2\sqrt{k}} \le \mathrm{e}^{-\gamma\sqrt{h}}$$

for a suitable constant $\gamma > 0$. This concludes the proof.

The next lemma shows that, for any given Q_i of scenario I, components of points inside distinct lakes cannot be linked, bypassing $\mathcal{G}$. Recall that we assumed that $K = 98\alpha$.

Remark 3. Let i be an open cell, and let v_i be a point inside it. By the definition of open, if v_j is a point inside a cell j within a distance $D - 3$ from i then v_i and v_j belong to the same connected component, because every point in a cell is a neighbor of every other point inside the cell or a neighboring point. In particular, if v_i belongs to the giant component, so does v_j.

Lemma 5. *Let u and v be points contained in two distinct lakes of scenario III. Unless they both belong to $\mathcal{G}$, they are not neighbors.*

Proof. Let us assume by contradiction that u and v are neighbors in scenario I without belonging to $\mathcal{G}$. Since u and v belong to different lakes, they must be separated by a portion of the giant cluster. Thus, from Remark 3, u and v must be separated by at least $2(D - 3)$ good cells. But every good cell contains at least $\alpha/2$ points, and, therefore, if uv is an edge then u reaches at least $2(D - 3)\alpha/2 = 102\alpha - 3\alpha > 98\alpha$ points. This is a contradiction because a vertex in phase 1 reaches $K = 98\alpha$ points.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

The following theorem immediately follows from Theorem 1, Lemmas 1, 4, and 5. We assume that K is so large that (4) of Theorem 1 is verified, i.e. there exists a giant component with high probability. Moreover, we note that in Theorem 1 the constant ξ depends on the constant c and, for a given constant c, there is a constant $K(c)$, depending on c, such that, for every $K > K(c)$, the result follows.

Definition 3. Let $E_{\varphi,c,n}$ be the event that at the end of phase 1 there is a unique giant component containing at least cn points, while the remaining components are contained inside lakes, with each lake containing at most $\varphi \log^2 n$ points.

The next theorem shows that, by choosing φ large enough, we can upper bound $\Pr(\overline{E_{\varphi,c,n}})$ with any inverse polynomial.

Theorem 2. *For every $c \in (0, 1)$,*

$$\Pr(\overline{E_{\varphi,c,n}}) \leq \frac{2}{n^d}$$

for sufficiently large n, where $d = \gamma\sqrt{\varphi} - 1$ and $\gamma > 0$ is the same constant as in Lemma 4.

Proof. The event $E_{\varphi,c,n}$ fails if the Poisson construction fails. This happens with probability $\mathrm{e}^{-\gamma_0 n}$, where γ_0 is given by Lemma 1.

By Theorem 1, for every $c \in (0, 1)$, there exists $K(c)$ such that, when the algorithm is run with parameter $K > K(c)$, the probability that there is no component with at least cn points at the end of phase 1 is at most $2\mathrm{e}^{-\xi\sqrt{n}}$. By Lemma 4 and the union bound, the probability that there exists a lake with more than $\varphi \log^2 n$ points is at most

$$n\mathrm{e}^{-\gamma\sqrt{\varphi \log^2 n}} = n^{-\gamma\sqrt{\varphi}+1}.$$

Using Lemma 5, we know that the points belonging to distinct lakes are in distinct components; therefore, the total failure probability is

$$\Pr(\overline{E_{\varphi,c,n}}) \leq \mathrm{e}^{-\gamma_0 n} + 2\mathrm{e}^{-\gamma_1\sqrt{n}} + n^{-\gamma\sqrt{\varphi}+1} \leq \frac{2}{n^d}$$

for large enough n.

6. Connectivity

In this section we discuss the connectivity properties of Algorithms A and B. Recall the event $E_{\varphi,c,n}$ of Theorem 2. We crucially observe that $E_{\varphi,c,n}$ implies that (a) there is a unique giant component $\mathcal{G}$ in the network and (b) a device is continental if and only if it belongs to $\mathcal{G}$.

Theorem 3. *For every $d > 0$, there exists a choice of the parameters K and φ such that the probability that Algorithm A sets up a connected network is at least $1 - n^{-d}$.*

Proof. Assume that the event $E_{\varphi,c,n}$ of Definition 3 holds. Then, after the test '$|C(v)| > \varphi \log^2 n$' in phase 2, every node in $\mathcal{G}$ will declare itself to be continental, and every other node will remain lacustrine. Then, in step (ii) of phase 2, every lacustrine node will find a neighbor in the unique giant component. The claim follows from Theorem 2.

The proof of the connectivity properties of Algorithm B is more involved.

Theorem 4. *For every $d > 0$, there exists a choice of the parameters K, φ, and t such that the probability that Algorithm B sets up a connected network is at least $1 - n^{-d}$.*

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

Proof. We will refer to one iteration of phase 2 of Algorithm B as an *exploration phase*. Let us define the event $G_{t,n} = \{\text{the network is connected within } t \log n \text{ exploration phases}\}$. We will show the stronger claim that, for every $d > 0$, there exists $t > 0$ such that

$$\Pr(\overline{G_{t,n}}) \leq \frac{1}{n^d}.$$

Recall that $\Pr(\overline{E_{\varphi,c,n}}) \leq n^{-\Theta(\varphi)}$. This probability can be made arbitrarily small by choosing the parameter φ appropriately in the algorithm.

Consider an (alternative) partition of the unit square into square cells such that the expected number of points in every cell is $b \log n$, where the value of b will be fixed later. Focus on one of these cells, and let X be the number of points that end up inside the cell. Here X is a Bernoulli variable $B(n, p)$ with $p = b \ln n/n$. By the large deviation principle for Bernoulli random variables we have

$$\Pr\left(|X - b \ln n| \geq \frac{b \ln n}{2}\right) \leq \frac{1}{n^{c_0 b}} = n^{-\Theta(b)}, \tag{7}$$

where c_0 is a positive constant not depending on n and b. So, by increasing the constant b, we can make the probability in (7) smaller than any inverse polynomial.

Let us now define the event $A_{b,n} = \{\text{the number of points in every cell is in } [\frac{1}{2}b \log n, \frac{3}{2}b \log n]\}$. By (7) and the union bound,

$$\Pr(\overline{A_{b,n}}) \leq \frac{1}{n^{c_0 b - 1}}.$$

We now show that $E_{\varphi,c,n}$ and $A_{b,n}$ imply $G_{t,n}$. Let $J := 49 \times (\frac{3}{2}b \log n)$. The point of this definition is that after J exploration phases a device u has captured all devices inside its cell and in the neighboring cells. This is because, by the definition of Algorithm B, every time a device runs an exploration phase a new point is captured. Let us see what happens by round $J + 1$.

Pick any cell and consider any point u inside it. If the cell of u or one of its neighboring cells contains a continental device v then, by the definition of Algorithm B, v will capture u, and u will join $\mathcal{G}$ since, as remarked, the event $E_{\varphi,c,n}$ ensures that $\mathcal{G}$ contains all devices that mark themselves continental, and only those. Otherwise, let v be any point in the cell of u or in one of its neighboring cells. Like u, v is lacustrine and, therefore, it has executed J exploration phases. At this point, u has captured v, and vice versa. But then u and v are linked by an edge. Since u and v are arbitrary, this shows that the points in the cell of u and in its neighboring cells form a complete graph. Since our choice of the cell and of u were arbitrary, by transitivity, the whole network must be connected after J exploration phases. Therefore, $E_{\varphi,c,n}$ and $A_{b,n}$ imply $G_{t,n}$.

The above argument shows that if we set $t = 49\frac{3}{2}b$ in phase 2 of Algorithm B, then the network is not connected within $t \log n$ exploration phases with probability at most

$$\Pr(\overline{G_{t,n}}) \leq \Pr(\overline{A_{b,n}}) + \Pr(\overline{E_{\varphi,c,n}}) \leq \frac{1}{n^{\gamma\sqrt{\varphi} - 1}} + \frac{1}{n^{c_0 b - 1}}.$$

This can be made smaller than any inverse polynomial by choosing b and φ appropriately. The claim follows.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

7. Expected and maximum degrees

The next theorem bounds the maximum degree of the nodes at the end of the algorithms.

Theorem 5. *With high probability, the maximum degree of a node in the network set up by Algorithms A and B is $O(\log^2 n)$ and $O(\log n)$, respectively.*

Proof. We prove the claim for Algorithm B. The proof for Algorithm A is analogous. Assume that the event $E_{\varphi,c,n}$ of Theorem 2 holds. By construction, the degree of lacustrine nodes is upper bounded by $K + t\log n \leq a\log n$ for a proper constant a. To conclude the proof, it is enough to show that continental nodes that increase their radius in phase 2 also have degree $O(\log n)$ at the end of the protocol.

Let $R := \sqrt{d\log n/n}$ for a constant d to be fixed later, and let $B(u, R)$ denote the ball of radius R centered at a fixed vertex u. The expected number of points inside $B(u, R)$ is $\pi d\log n$ for any u. Fix a device u, and consider the following event

$$E(u, R) = \{a\log n + 1 \leq |B(u, R)| \leq 2\pi d\log n\}.$$

Consider then $E(R) := \bigcap_u E(u, R)$. We can choose d in such a way that $E(R)$ holds with probability growing to 1 as n goes to ∞. In what follows, assume that $E(R)$ occurs.

Observe now that the maximum transmission radius r at the end of the algorithm is realized by a lacustrine node. This is because after phase 1 continental nodes increase their radius only in response to a request of a lacustrine node. It follows that $r \leq R$. Thus, we have, for every node u, $|B(u, r)| \leq |B(u, R)| = O(\log n)$, which implies the claim.

Theorem 6. *The expected degree of every point at the end of Algorithms A and B is bounded by a constant.*

Proof. Given the same configuration of nodes within the unit box, the degree of each node in the network constructed by Algorithm A is larger than or equal to the degree of the same node in the network produced by Algorithm B. Therefore, it suffices to argue for Algorithm A only.

Consider first the expected degree of any lacustrine point v. Let L be the lake containing v at the end of phase 1, and let $s(v)$ be the point of the initial giant component $\mathcal{G}$ closest to v. By Lemma 5, the value of $k(v)$ is bounded by $1 + \sum_{i\in L} Z_i$, since in the worst case, v will capture the points in L plus $s(v)$. By Lemma 4,

$$\mathrm{E}\biggl[\sum_{i\in L} Z_i\biggr] \leq \sum_h h\,\Pr\biggl(\sum_{i\in L} Z_i \geq h\biggr) \leq \sum_h h\mathrm{e}^{-\gamma\sqrt{h}} < \infty.$$

The degree of continental nodes can be bounded in a similar way.

8. Near-optimal power consumption

In this final section we study the expected power consumption needed to sustain the networks generated by our algorithms. We assume that the power needed to cover a circle of radius r is proportional to the area of the circle. More precisely, denoting by r_v the radius covered by node v, the *total power consumption* is defined to be $\sum_{v\in V} r_v^2$, where V denotes the set of n devices. For a given distribution of points in the square, denoted as D, let $\mathrm{opt}(D)$ denote the optimal power consumption, and let $A(D)$ and $B(D)$ denote the power consumptions needed to sustain the network generated by Algorithms A and B, respectively. We will show that there

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

exist two absolute constants α and β (independent on the number of points) such that, for all D, $\alpha \leq \mathrm{E}[\mathrm{opt}(D)]$, and $\mathrm{E}[A(D)] \leq \beta$ and $\mathrm{E}[B(D)] \leq \beta$.

We begin by showing that $\alpha \leq \mathrm{E}[\mathrm{opt}(D)]$. Let $\tilde{R}_v$ be the random variable denoting the minimal distance from v to all other nodes. The random variables $\{\tilde{R}_v : v \in V\}$ are identically distributed. Obviously, these radii are the minimal requirement to realize a connected network. We will bound $\mathrm{opt}(D)$ by the total area induced by them. Therefore,

$$\sum_{v \in V} \tilde{R}_v^2 = n\tilde{R}_{v_1}^2 \leq \mathrm{opt}(D).$$

Taking expectations we obtain

$$\mathrm{E}\bigg[\sum_{v \in V} \tilde{R}_v^2\bigg] = n\,\mathrm{E}[\tilde{R}_{v_1}^2] \leq \mathrm{E}[\mathrm{opt}(D)]. \tag{8}$$

We then bound $\mathrm{E}[\tilde{R}_{v_1}^2]$. Let us arbitrarily fix the position of the first point v_1 in the unit square. Then the probability that all the other $n-1$ points have distance from v_1 at least $1/\sqrt{n}$ is larger than 0 for every n, and also the limit of this probability, when n goes to ∞, is larger than 0. Therefore, there exists a constant $p_0 > 0$ such that

$$\mathrm{E}[\tilde{R}_{v_1}^2] \geq \frac{p_0}{n}.$$

From (8), it follows that $\mathrm{E}[\mathrm{opt}(D)] \geq p_0$ for all D with at least two points.

We now turn to the upper bound for the expected power consumption of our algorithms.

Theorem 7. *The expected power consumption of the networks that are set up by Algorithms A and B is upper bounded by a constant β, which does not depend on the number of devices n.*

Proof. By the remark at the beginning of the proof of Theorem 6, the average power consumption of Algorithm A is no smaller than that of Algorithm B. So we will consider Algorithm A only.

We divide the proof into two steps. First we prove that after phase 1 of Algorithm A the expected power consumption is finite. Let $W_v^{(1)}$ and $W_v^{(2)}$ denote the powers of device v at the end of phase 1 and phase 2, respectively, of the algorithm, and let V denote the set of n points that are uniformly and randomly distributed within the unit box. Note that if a device u uses power w then the portion of the unit box that is covered by u's ball has area at least $w/2 = \Theta(w)$. The total power at the end of phase 1 is thus $\sum_v W_v^{(1)}$. The random variables $\{W_v^{(1)} : v \in V\}$ are identically distributed, and so $\mathrm{E}[\sum_v W_v^{(1)}] = n\,\mathrm{E}[W_{v_1}^{(1)}]$.

Let $N(v, w)$ denote the number of points inside the ball centered at v when the power of v is w; we also define $N_0(v, w)$ to denote the number of points inside the same ball when the Poisson process P_0 is used to generate the points (see Section 4).

Clearly, the following two events are equal:

$$\{W_{v_1}^{(1)} > w\} = \{N(v, w) < K\},$$

where K is the parameter of phase 1 of the algorithm. We know by Lemma 1 that the probability that (in a given region) the Poisson process P_0 generates more than n points is at most $\mathrm{e}^{-\gamma n}$ for

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

a constant γ. Therefore,

$$\begin{aligned}
\mathrm{E}[W_{v_1}^{(1)}] &= \int_0^2 \mathrm{P}(W_{v_1}^{(1)} > w)\,\mathrm{d}w \\
&= \int_0^2 \mathrm{P}(N(v_1, w) < K)\,\mathrm{d}w \\
&\le \int_0^2 [\mathrm{P}(N_0(v_1, w) < K) + \mathrm{e}^{-\gamma n}]\,\mathrm{d}w \\
&\le \int_0^2 \mathrm{P}(N_0(v_1, w) < K)\,\mathrm{d}w + 2\mathrm{e}^{-\gamma n}. \qquad (9)
\end{aligned}$$

Note that the intensity $\mu = \frac{3}{4}n$ of the Poisson process $P_0(n)$ satisfies $\mu \ge 3wn/8$ for each $w \in [0, 2]$. Therefore, if $3wn \ge 8K$ then $\mu \ge K$.

We now use the fact that $K \ln(K/\mu) - (K - \mu)$ is increasing in μ for $\mu \in [K, \infty)$. If we choose w to satisfy $w > 8K/3n$ then

$$K \ln\biggl(\frac{K}{\mu}\biggr) - (K - \mu) \ge K \ln\biggl(\frac{8K}{3wn}\biggr) - \biggl(K - \frac{3wn}{8}\biggr),$$

and, therefore, the right-hand side of (9) is at most

$$\int_0^{8K/3n} 1\,\mathrm{d}w + \int_{8K/3n}^2 \exp\biggl(-K \ln\biggl(\frac{8K}{3wn}\biggr) + \biggl(K - \frac{3wn}{8}\biggr)\biggr)\,\mathrm{d}w + 2\mathrm{e}^{-\gamma n}.$$

By setting $y = 3nw/8K$ we can see that this is at most

$$\frac{8K}{3n} + \frac{8K}{3n}\int_1^\infty \exp(-K \ln(y) + (K - y))\,\mathrm{d}y + 2\mathrm{e}^{-\gamma n} \le \frac{A_K}{n},$$

where A_K is a positive constant depending only on K. Therefore, $\mathrm{E}[\sum_v W_v^{(1)}] \le A_K$. Thus, the expected power is bounded by a constant at the end of phase 1. Let us turn to phase 2.

Given a lake L, we use $|L|$ to denote the number of cells that belong to it, and N_L to denote the number of points inside it. For a given lake L, we have the bound

$$\sum_{v \in L} W_v^{(2)} \le b\frac{|L|}{n} N_L,$$

where $b > 0$ is a constant not depending on n.

Consider the random variables $\{W_i\}$ defined in Lemma 4 to stochastically dominate the random variable N_L. We have $N_L \le_{\text{stoch}} \sum_{i=1}^{|L|} W_i$. Hence,

$$\begin{aligned}
\mathrm{E}\biggl[\sum_{v \in L} W_v^{(2)}\biggr] &\le \frac{b}{n}\,\mathrm{E}[|L| N_L] \\
&\le \frac{b}{n}\,\mathrm{E}\biggl[|L| \sum_{i=1}^{|L|} W_i\biggr] \\
&\le \frac{b}{n} \sum_{l=1}^{\infty} \mathrm{E}\biggl[|L| \sum_{i=1}^{|L|} W_i \biggm| |L| = l\biggr] \Pr(|L| = l).
\end{aligned}$$

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press

The W_is are independent from $|L|$ and have the same distribution. Thus, recalling Lemma 3,

$$\begin{aligned}\frac{b}{n}\sum_{l=1}^{\infty}\mathrm{E}\biggl[l\sum_{i=1}^{l}W_i \biggm| |L|=l\biggr]\Pr(|L|=l) &= \frac{b}{n}\sum_{l=1}^{\infty}\mathrm{E}\biggl[l\sum_{i=1}^{l}W_i\biggr]\Pr(|L|=l)\\ &= \frac{b}{n}\sum_{l=1}^{\infty}l^2\,\mathrm{E}[W_1]\Pr(|L|=l)\\ &\le \frac{b}{n}\,\mathrm{E}[W_1]\sum_{l=1}^{\infty}l^2\mathrm{e}^{-\gamma\sqrt{l}}\\ &= \frac{c}{n}.\end{aligned}$$

Therefore, the expected power after phase 2 can increase at most by a constant because the cardinality of the lakes are random variables with the same distribution, and the total number of the lakes is at most n.

Acknowledgement

We would like to thank Ashish Goel for interesting discussions.

References

[1] Blough, D. M., Leoncini, M., Resta, G. and Santi, P. (2006). The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. *IEEE Trans. Mobile Computing* **5,** 1267–1282.

[2] Clementi, A. E. F., Penna, P. and Silvestri, R. (2004). On the power assignment problem in radio networks. *Mobile Networks Appl.* **9,** 125–140.

[3] Deuschel, J.-D. and Pisztora, A. (1996). Surface order large deviations for high-density percolation. *Prob. Theory Relat. Fields* **104,** 467–482.

[4] Grimmet, G. (1999). *Percolation*, 2nd edn. Springer, Berlin.

[5] Hou, T. and Li, V. (1986). Transmission range control in multihop packet radio networks. *IEEE Trans. Commun.* **34,** 38–44.

[6] Kleinrock, L. and Silvester, J. A. (1978). Optimum transmission radii for packet radio networks or why six is a magic number. In *Proc. IEEE Nat. Telecommun. Conf.* (Birmingham, AL, December 1978), pp. 431–435.

[7] Kučera, L. (2005). Low degree connectivity in ad-hoc networks. In *Algorithms – ESA 2005* (Proc. 13th Europ. Symp. Algorithms; Lecture Notes Comput. Sci. **3669**), Springer, Berlin, pp. 203–214.

[8] Liggett, T. M., Schonmann, R. H. and Stacey, A. M. (1997). Domination by product measures. *Ann. Prob.* **25,** 71–95.

[9] Ni, J. and Chandler, S. A. G. (1994). Connectivity properties of a random radio network. *IEEE Proc. Commun.* **141,** 289–296.

[10] Takagi, H. and Kleinrock, L. (1984). Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. Commun.* **32,** 246–257.

[11] Xue, F. and Kumar, P. R. (2004). The number of neighbors needed for connectivity of wireless networks. *Wireless Networks* **10,** 169–181.

[12] Zhu, S. Setia, S. and Jajodia, S. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *Proc. 10th ACM Conf. Comput. Commun. Security* (Washington, DC, October 2003), ACM, New York, pp. 62–72.

https://doi.org/10.1239/aap/1275055242 Published online by Cambridge University Press