Managing Random Sensor Networks by means of Grid Emulation*

Zvi Lotker[1] and Alfredo Navarra[2]

[1] Centrum voor Wiskunde en Informatica Kruislaan 413,
NL-1098 SJ Amsterdam, Netherlands. Email: lotker@cwi.nl
[2] Computer Science Department, University of L'Aquila.
Via Vetoio I-67100 L'Aquila, Italy. Email: navarra@di.univaq.it

Abstract. A common assumption in sensor networks is that the sensors are located according to a uniform random distribution. In this paper we show that uniform random points on the two dimensional unit square are almost a "grid". In particular, for a synchronous geographic sensor network we show how to emulate any grid protocol on random sensor networks, with high probability.
This suggests the following framework. In order to solve a problem on a random sensor network we solve the same problem on the grid. Then we use our emulation to make the obtained solution suitable for random sensor network. We analyze the cost of the emulation in terms of consumed energy and time. Finally we provide three examples that illustrate our method.

Keywords: Routing, Scheduling, MAC-layer, Collisions, Grid.

1 Introduction

A sensor network is usually modeled as a radio network where the sensors are spread out at random over a given area according to a uniform distribution. The structure of sensor networks is complex and presents many challenges. This is due to its random characteristic and its induced physical limitations (i.e., energy consumption, transmission range and open medium access constraints).

In a random sensor network usually each sensor does not have any knowledge about the network in which it is working, unless some local information is obtained by exchanging control messages with its neighbors. Moreover, since the sensors are placed at random, a first glance might suggest a total lack of structure. This is not necessarily the case. Dealing with randomness is always a problem. One way of dealing with it is by simulations. This solution is time and effort consuming and its accuracy is usually hard to evaluate. Another way of approaching this problem is by applying sophisticated stochastic geometry tools.

* The research was partially funded by the European projects COST Action 293, "Graphs and Algorithms in Communication Networks" (GRAAL) and COST Action 295, "Dynamic Communication Networks" (DYNAMO).

This approach is again time costly and it is not always simple. Understanding the structure of random sensor networks is a quintessential problem in the field of sensor networks. Clearly an understanding of this structure can lead to a major improvement in energy consumption and in the overall performance of the random network.

A standard and elegant technique when dealing with complex structures is to find a simpler structure that is close enough to the complex one, and yet simple enough to understand (see for instance [8]). This is our main goal in this work.

Our contribution is a grid protocol emulation for random sensor networks. In order to achieve this we develop optimal scheduling schemes that avoid collisions. More precisely we propose a general framework that is capable of emulating any protocol based on a grid structure for random sensors. In this way, we break the problem into 2 steps. The first step is to solve the problem on the grid. Since the grid is a well known and well researched structure, a textbook solution there probably already exists. The second step is to emulate the solution on the random sensor network using our grid emulation protocol. The advantages of this approach are evident. First, there are many problems that are already optimally solved on grids. Second, usually it is much easier to solve a problem on grids than on a random set of points. Moreover we are going to show that the cost of the grid emulation in terms of consumed energy and time is not too high. In particular, we use our method to solve the *Broadcast*, the *Gossiping* and the *Leafy Tree* problems on random sensor networks, obtaining satisfactory solutions. Last, the grid emulation can also be used as a rule of thumb to evaluate the correctness of simulations.

In order to achieve the grid emulation we develop a collision-free scheduling scheme. Using this scheduling scheme we developed a collision-free routing algorithm that can be easily applied in order to perform any desired communication on any sensed area of interest. Our scheme is completely independent of the routing protocol among the location-aware ones [1,11]. It is worth noting that the combination of the routing protocol with the scheduling scheme is the main key for the conservation of energy in any communication. While the routing scheme, in fact, minimizes the energy needed to perform a desired communication, the scheduling prevents cases where communications must be repeated several times before succeeding. This concept was initiated by [7] where the authors dealt with random and deterministic scheduling functions. The main differences reside in their main assumptions for which each sensor is aware about the position of any other one and moreover each virtual grid square is assumed to be not empty. They also assume three basic states for the sensors. *Active*, when a sensor can transmit, *Passive*, when a sensor can receive and *Sleep* when a sensor is switched off in order to save energy. Concerning collisions, those are caused by superpositions of the transmission ranges of the sensors as in [3] but in [7] also by an extra range, called *interference range* (R_p). For the sake of clarity we do not cope with such an extra range but everything is easily scalable.

The paper is organized as follows. In the next section we describe the model and motivations that led to the assumptions made in the paper. In Section 3 we

take care of the MAC-layer in order to avoid collisions in the communications. We also provide analysis in order to estimate the needed time for a source-destination communication. In Section 4 we show how the combination of a routing protocol with our scheduling scheme can be applied in order to emulate grid structures hence implying a virtual infrastructure on the network (see for instance [14]). Finally, in Section 5 we discuss some conclusive remarks.

2 Model

As assumed in the large majority of the papers we consider random instances of sensor networks in the two dimensional space (see [1,11] for a survey on sensor networks routing protocols). The randomness of the spread sensors is usually motivated by the applications. The area of interest, in fact, where the sensing must be computed, can be an impervious, even dangerous area so that the sensors cannot be suitably set up. Without loss of generality we consider a square area using a uniform distribution. Each sensor knows its own location inside the considered area. Positioning information can be obtained through GPS systems, but also by cheaper means such as services like Ad-Hoc Positioning System (APS) [15] or the GPS-less low-cost outdoor localization for very small devices proposed in [4]. Sensors are assumed to be synchronized. As for the location awareness, the synchronization can be accomplished either by some strong assumption like a central clock to which each sensor refers (a GPS device can be also used for this purpose) or by means of cheaper strategies like the one presented in [16]. About the energy consumption concerning the sensor communications we refer to the most common power attenuation model [17] by which the signal power P_s of a sensor s decreases as a function of the distance in such a way that any station s' at distance $||s, s'||$ from s can receive a message from s if $P_s \geq O(||s, s'||^2)$. If a sensor is reached simultaneously by more than one transmission, a collision occurs and the received messages are assumed to be unreadable. Note that, in what follows, with "high probability" we mean a probability of $1 - \frac{1}{N}$ with $N = n \times n$ being the number of considered sensors.

3 MAC-layer

In this section we describe a deterministic MAC-layer schedule based on the locations of the transmitters. For simplicity we assume that the sensors lie on a regular 2-dimensional grid G of $N = n \times n$ vertices V. We will remove this assumption in Section 4. For the sake of generality, we assume that some of the grid points are free from sensors and that some of them have more than one. The second case can be simplified just by considering one sensor in such grid points, since sensors in the same location can check the presence of overlapping ones without loosing too much energy and time. Moreover, we assume that each sensor knows its position but they do not know anything about the topology of the network except that all the sensors are on some grid points. In order to

save energy, collisions should be avoided. We now describe an algorithm to perform communications without collision. Since a sensor does not have information about the other sensors, we have to assign slots of communication to each pair of the network to ensure communication. A time slot is just a window of time during which some sensors are allowed to terminate one transmission operation. Its duration is dependent by the technology of the used sensors and without loss of generality we can consider one time slot as one unit of time (see for instance Figure 1).

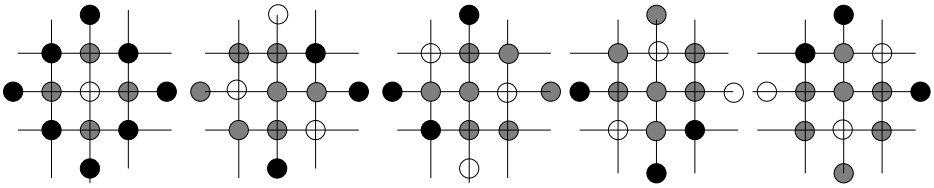

Fig. 1. Schedule scheme for 1-unit square grid transmissions. It needs 5 time slots to perform all the communications at distance 1. The white nodes are the transmitting one, the grey are the receivers and the black are inactive in order to avoid collisions.

Independently of the grid structure we need $\binom{N}{2} = \binom{n^2}{2}$ time slots (one for each possible pair). Indeed we can parallelize some of the transmissions in order to reduce the time needed to perform eventual communications.

Let $D = \{D(x, r) : x \in V, r \in \mathbb{R}\}$ be the set of disks of radius r centered at node x. A *schedule* $S : \mathbb{N} \to 2^D$ is a function from time step to a subset of disks. Next we define two properties of deterministic schedule.

Definition 1. *Let S be a deterministic schedule,*

1 *S has* no collisions *if any two nodes transmitting at the same time cannot reach a common node, i.e., $\forall c \in \mathbb{N},\ S(c)$ is a subset of disjoint disks.*
2 *S is* universal *if any source $x \in V$ destination $y \in V$ pair x, y can communicate infinitely many times, i.e., $\forall x, y \in V$ and $t \in \mathbb{N}\ \exists t' > t : D(x, r) \in S(t')$ with $r \geq ||x - y||$.*

Let $S(x, y, k)$ be the number of slots in the schedule S that the node x needs to wait in order to communicate with node y for the k-th times.

Definition 2. *Let S be a schedule, the* fairness *of S is*

$$\phi(S) = \max_{x,y \in V, k \in \mathbb{N}} \{S(x, y, k) - S(x, y, k - 1)\}.$$

Note that, without any information about the topology of the network, ϕ represents the time needed in the worst case to perform any communication.

Lemma 1. *For any universal schedule S without collisions $\phi(S) = \Theta(n^4)$.*

Proof. Assume, by contradiction, that $\phi(S) < \frac{n^4}{64}$. This means that considering any interval of time equal to $\frac{n^4}{64}$ we must find in S all the source-destination pairs. Since the number of pairs at distance more than $\frac{n}{2}$ is bigger than $\frac{n^4}{16}$ and that without collisions we can parallelize at most 4 of them, at least $\frac{n^4}{64}$ time slots are needed. The claim then holds by remembering that the number of all the source-destination pairs is $\binom{n^2}{2}$. □

Since we are interested in random points with uniform distribution, a natural question is whether we can improve the expectation of the communication time. Depending on the desired communications, in many cases a good idea for a routing algorithm may be to prefer short hop instead of long ones. This is due to the fact that short transmissions are less expensive in terms of consumed energy and moreover they can be parallelized much more than the long ones.

Let us divide all the source-destination pairs according to their Euclidean distance. Let $P = \{\pi_1, \pi_2, \ldots, \pi_d\}$ be such a partition where $\pi_i = \{(x, y) : x, y \in V(G)$ and $i - 1 \leq ||x - y|| < i\}$ is the set of all the pairs at distance i on the grid and $d = \sqrt{2}n$ is the diameter of G. Following the previous ideas we want to perform all the communications of each π_i in the best way.

Since the disks close to the boundary of the grid are not full, we define $b(r) = \max_{x \in V(G)} |D(x, r)|$ to be the maximal number of grid points contained in a ball of radius r. Let opt_i be the optimal number of time slots needed to perform the communications defined by π_i.

Note that there is a big difference in the number of possible disjoint disks used by *opt* between the case of radius $r < \frac{n}{4}$ and the case of $r > \frac{n}{4}$ hence we describe two different procedures. In the first case, we consider a dense maximal disjoint packing $P_1(r)$ of the grid points by disks of the radius r. Since such a packing leaves holes between the circles, we need another shifted one, $P_2(r)$ to cover them (see for instance Figure 2).

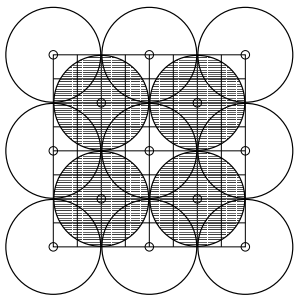

Fig. 2. The coverage of the whole grid by the two described complementary packings $P_1(r)$ (empty circles) and $P_2(r)$ (shaded circles).

Since for each disk in $P_1(r)$, (resp. $P_2(r)$) there are only 9 discs at distance less than $2r$ (see figure 3), we partition $P_1(r)$, (resp. $P_2(r)$) into 9 subparts in a way that all the distances between discs in each subpart is bigger than $2r$. Denote P_i^j to be the $j = 1, ..., 9$ subparts of the packing $i = 1, 2$. We schedule the points covered by $P_1^j(r)$ to transmit before the ones covered only by $P_2^j(r)$.

Let $g_{0,0}$ be the point at the center of the Grid and let us consider the circle centered on it. We label the contained grid points from 1 to $b(r)$ in such a way that each node get a unique label. We use the same numbering process for each circle of both $P_1(r)$ and $P_2(r)$. This numbering represents the transmitting sequence in which every node of $P_1(r)$ (resp. $P_2(r)$) with the same label can simultaneously transmit.

procedure $\mathcal{S}(T, P_1(r), P_2(r))$
1: **for** $j = 1$ **to** 9 **do**
2: Let v_i be a node covered by P_1^j and let n_i be its label.
3: **for** $i = 1$ **to** $b(r)$ **do**
4: every node labelled as n_i is allowed to transmit at radius $2r$ in the $(T + n_i + b(r)(j-1))$-th time slot
5: **end for**
6: Let v_i be a node covered by $P_2(r)$ and let n_i be its label.
7: **for** $i = 1$ **to** $b(r)$ **do**
8: every node labelled as n_i is allowed to transmit at radius $2r$ in the $(T + n_i + b(r)j)$-th time slot
9: **end for**
10: **end for**

In the second case, that is, when $r > \frac{n}{4}$, if our schedule uses one disk in a time slot it is still ok since the optimal solution cannot parallelize too many of such communications, i.e., no more than 9. In this way we just loose a constant factor.

Lemma 2. *The schedule $\mathcal{S}(T, P_1(r), P_2(r))$ performs all the communications of π_i in $O(opt_i)$ time slots without any collision.*

Proof. Let us first provide a lower bound for opt_i in the case of $i \leq \frac{n}{4}$. Consider the disk $D(i)$ of radius i placed at $g_{0,0}$. Such a disk contains exactly $b(i)$ nodes. The disks centered in those points have an overlapping in $g_{0,0}$. This means that, in order to avoid the collision in the central node, all those nodes have to transmit in different time slots. Note that the Schedule algorithm $\mathcal{S}(T, P_1(r), P_2(r))$ needs $18b(i)$ time steps for all communications of π_i. This means that in this case we obtain a 18-approximation on the number of time slots needed to perform all the communications. If $i > \frac{n}{4}$ using packing arguments, opt_i cannot transmit with more than 9 disks at the same time, hence a 18-approximation holds. To see that $\mathcal{S}(T, P_1(r), P_2(r))$ is collision free we use the fact that the distance of the discs in each P_i^j is bigger than $2r$. Since each sensor transmits at radius $2r$, the sensors that transmit simultaneously do not interfere with each other, and the lemma follows. □

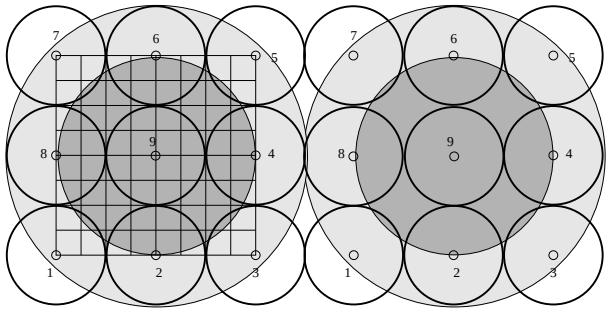

Fig. 3. The subpartition of P_1^j, $j = 1, ..., 9$. The numbers in the figure determine to which subpartition the node belongs. All the grey areas show the total area that can be covered by nodes from P_1^9. The dark grey areas show the nodes that receive the transmission from the central node in P_1^9. Note that in this case $b(3) = 13$, the total time it takes to get all the communications of π_3 is $13 \cdot 18 = 234$. For the sake of clarity the grid is shown just in the left part of the figure.

4 Grid Emulation

In this section we apply the previous results in order to achieve a general technique for emulating any grid protocol with random sensors. The idea is to "move" the points to a grid structure. The movements (Long or Short) are performed by increasing the radius of transmissions to ensure that all the neighbors of the grid structure can communicate. The difference between the Long and the Short movements concerns the size of the grid structure and the technique to calculate the relative locations of the points. More precisely, sometimes we use global or local information. Another important issue is the granularity of the considered grid. In what follows we also estimate the needed overhead for the consumed energy induced by our emulation strategy. Note that, since our scheduling scheme is placed at the MAC-layer, our results can be achieved with any location-aware routing protocol.

Let us assume a protocol $\mathcal{A}$ performed on grid networks. Actually for each node (x, y) of the grid a protocol $\mathcal{A}$ defines the instruction $\mathcal{A}_{x,y}(t)$ it has to compute at time t. Let Γ be a mapping from the set of random points P to the grid nodes. Note that Γ changes according to the size of the chosen grid. In order to perform the emulation we accumulate several time steps into one phase. Each phase can be considered as one basic time step in the protocol $\mathcal{A}$. The number of time steps that defines one phase is the output of the schedule we use in order to perform one single communication in the grid. Let μ be the maximal distance between any pair (x, y) and its image $\Gamma(x, y)$. From lemma 2 it follows that $\mathcal{S}(T, P_1(\mu + 1), P_2(\mu + 1))$ has no collision. Moreover the real distance between two sensors that are neighbors on the grid is less than or equal to $2\mu + 1$. It follows that two sensors that are neighbors on the grid can communicate with each other.

Long Movement To achieve a one to one mapping between the grid points $G_{n,n}$ and the n^2 sensors we use the results of [18]. By allowing each sensor to move at most $O(\log^{\frac{3}{4}} n)$ we achieve such a matching Γ with high probability. Therefore we have $\mu \leq O(\log^{\frac{3}{4}} n)$. Without loss of generality let $\mu \in \mathbb{N}$. The schedule will be $\mathcal{S}(T, P_1(\mu+1), P_2(\mu+1))$ according to the procedure described in Section 3. By Lemma 2, the time needed to perform it is then $O(\mu^2)$. Moreover, since it is possible that several (roughly $\log(n)$) sensors will be in the same grid square, we must multiply by a factor of $O(\log n)$ in order to enable all the possible communications given by the emulated protocol $\mathcal{A}$.

In this case we need global information to compute Γ, i.e., each sensor has to know its associated grid node. In order to perform every local communication round on the grid, using the scheduling algorithm of Section 3, we need time $\Theta(\log^{\frac{3}{2}} n)$ and also the energy must be multiplied by the same factor. This means that up to a poly-log factor we achieve an upper bound for the energy and time needed for random points in the plane to emulate the protocol $\mathcal{A}$. More precisely, using the long movement strategy we get the following theorem.

Theorem 1. *Any protocol $\mathcal{A}$ over a grid network $G_{n,n}$ can be emulated with high probability on a set of n^2 random points with stretch factors of $O(\log^{\frac{5}{2}} n)$ in time and $O(\log^{\frac{3}{2}} n)$ in energy.*

Note that, considering one source-destination pair, the previous method is the fastest one in terms of time (scheduling steps), on the other hand each sensor transmits for long distance, i.e., $O(\log(n)^{\frac{3}{4}})$. This is expansive in terms of energy consumption. This suggests to consider a suitable routing scheme in order to manage a good trade-off between the minimum delivery time and the minimum energy consumption. This remains a challenging issue according to the actual desired patterns of communication.

Short Movement In this case we consider a grid $G_{O(\frac{n}{\sqrt{\log n}}),O(\frac{n}{\sqrt{\log n}})}$ but still with n^2 sensors, therefore there is still an average of $\log n$ nodes that belong to the same grid square. We associate to the left bottom grid node of each grid square one sensor lying in that square. This is accomplished by means of standard local leader election strategies [13], which costs $\log \log n$ time steps with high probability. We summarize the Short movement performance in the next theorem.

Theorem 2. *Any protocol $\mathcal{A}$ over a grid network $G_{\frac{n}{8\sqrt{\log n}},\frac{n}{8\sqrt{\log n}}}$ can be emulated with probability $1-\frac{1}{n^6}$ on a set of n^2 random points with constant stretch factors in time and in energy.*

*Proof (*sketch). In order to emulate $\mathcal{A}$ we need to have a sensor in each grid square. Note that the size of each square is $64 \log n$, and so the expected number of sensor in each grid square is $64 \log n$. Formally let $0 < i, j < \frac{n}{8\sqrt{\log n}}$. Denote the number of sensors in the grid square i, j by $X_{i,j}$. Using Chernoff we get

$$\Pr[X_{i,j} \leq 64(1-\lambda)\log(n)] \leq e^{-64\lambda^2 \frac{\log(n)}{2}}$$

Taking $\lambda = 1/2$ it follows that $\Pr[X_{i,j} \leq 32\log(n)] \leq \frac{1}{n^8}$. To bound the probability that none of the $\frac{n^2}{64\log n}$ grid squares in $G_{\frac{n}{8\sqrt{\log n}},\frac{n}{8\sqrt{\log n}}}$ is empty we use union bound. Since the number of grid squares is less than n^2 it follows that:

$$\Pr[Min\{X_{i,j} : 0 < i,j < \frac{n}{8\sqrt{\log n}}\} \leq 32\log(n)] < \frac{n^2}{n^8} = \frac{1}{n^6}$$

In order to perform every local communication round on the grid, using the scheduling algorithm of Section 3 it follows that we need a constant stretch factor in time and in energy. The constant time factor follows from the fact that we use π_1 scheduling. The energy constant follows from locality, i.e., two neighbors on the grid are at distance $8\sqrt{\log n}$ (on the grid), and their real distance on the plane is less than $\sqrt{38}\sqrt{\log n}$. Moreover in the short movement, the grid neighborhood of each node coincides with the real neighborhood on the plane. Therefore using the π_1 scheduling we can perform all the desired communications. This is accomplished by the "trick" of throwing a number of sensors (n^2) that is bigger than the grid dimension ($\frac{n^2}{\log n}$). □

The Short Movement performs much better than the Long one both in time and in energy consumption. This is due to the fact that spreading more sensors than the number of grid nodes substantially increases the probability that some sensor is close to a grid node. In fact, each grid square may contain several points (usually $\log n$ points). Therefore, the short movement has an overhead generated in the initial stage due to the leader election which can be done in $O(\log\log(n))$. This leader will be the node that emulates the grid nodes. Moreover we can permute the leader role among all the sensors that belong to the same grid square. By doing this we balance the energy consumption among all the sensors, not only among the leaders. By doing this we prolong the lifetime of the network. The time needed to achieve this permutation is $O\log(n)\cdot\log\log(n))$. Since such a procedure is very local it is also not expensive in energy. By means of such movements we are finally able to remove the assumption of Section 3 for which the sensors were placed on the grid nodes.

4.1 Applications

In order to demonstrate the strength of our results, we now describe some important application problems on random sensor networks easily solvable by means of our technique with high probability related to the location of the sensors. We focus on the upper bounds obtained by such a method. Roughly speaking the idea is to consider a generic protocol $\mathcal{A}$ and perform it by the Long or the Short movement.

Broadcast One of the most important protocols in any kind of communication network is given by the Broadcast protocol (see for instance [2,5,9,10]). In [5] the authors describe the optimal algorithm for grid structure roughly showing that it needs $D + 2$ hops where D is the diameter of the grid. By applying the Short Movement we can achieve the Broadcast in $O(D)$ time and $O(n^2)$ energy hence solving the problem almost optimally. Note that, for the specific broadcast application it is useless to apply the Long Movement wasting much more time and energy.

Corollary 1. *$\mathcal{A} \equiv$ Broadcast over a grid network $G_{O(\frac{n}{\sqrt{\log n}}),O(\frac{n}{\sqrt{\log n}})}$ can be emulated with high probability on a set of n^2 random points with $O(\frac{n}{\sqrt{\log n}})$ time and $O(n^2)$ energy.*

Gossiping Another important basic protocol in communication tasks is the gossiping. Each node participating in the protocol is assumed to have a value which should be transmitted to all the other ones. A trivial solution is then given by performing n^2 broadcasting communication, that is, one per node. In [19] a $O(n^3)$ deterministic Gossiping algorithm for radio networks of n^2 nodes is presented without any knowledge about the node locations. Restricting the attention to $G_{n,n}$ as shown in [6] the number of communications has an upper bound of $O(n^2)$ and the needed time has an upper bound of $O(n)$.

Corollary 2. *$\mathcal{A} \equiv$ Gossiping over a grid network $G_{n,n}$ can be emulated with high probability on a set of n^2 random points with $O(n \log^{\frac{5}{3}} n)$ time and $O(n^2 \log^{\frac{3}{2}} n)$ energy.*

In order to apply the Short Movement we have to pay attention to the values that belong to the nodes that are not actively participating in the protocol. We divide the protocol into two phases. In the first one each elected "leader", representative of every grid node, has to collect all the values belonging to the surrounding sensors that are physically associated to the same grid node (at most $O(\log n)$). This phase costs $O(\log n \log \log n)$. In the second phase the real gossiping starts between the grid nodes.

Corollary 3. *$\mathcal{A} \equiv$ Gossiping over a grid network $G_{O(\frac{n}{\sqrt{\log n}}),O(\frac{n}{\sqrt{\log n}})}$ can be emulated with high probability on a set of n^2 random points with $O(\frac{n}{\sqrt{\log n}})$ time and $O(\frac{n^2}{\log n})$ energy.*

Leafy Trees Given a graph $G = (V, E)$ the problem is to find a spanning tree with a maximal number of leaves [12]. Such a problem is very interesting in the field of sensor networks since increasing the number of leaves reduces the number of needed transmissions and hops. Usually the underlying graph that models a sensor network is complete so the leafy tree can be trivially solved by one node connected to all the other ones hence obtaining $n^2 - 1$ leaves. Actually such a

solution is practically not feasible due to the limited resources of the sensors, moreover, we are interested in the emulation of grid structures. On the full grid the maximal number of leaves is approximatively $\frac{2}{3}n^2$ (see Figure 4).

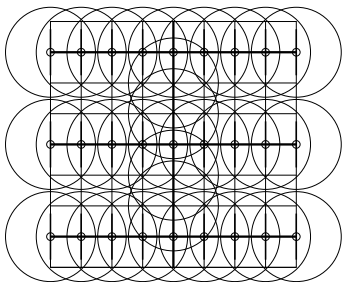

Fig. 4. The Leafy Tree for a grid network of 81 nodes. It contains 45 leaves.

Using the Short Movement we obtain a number of leaves proportional to $\frac{2}{3}$ of the grid nodes plus all the nodes associated to the same grid node but one, hence obtaining,

Corollary 4. *$\mathcal{A} \equiv$ Leafy Tree over a grid network $G_{O(\frac{n}{\sqrt{\log n}}),O(\frac{n}{\sqrt{\log n}})}$ can be emulated with high probability on a set of n^2 random points obtaining roughly $\frac{2}{3}\frac{n^2}{\log n} + n^2(1 - \frac{1}{\log n})$ leaves.*

5 Conclusion

In this paper we have shown that the combination of routing and the MAC-layer can be efficient in a sensor network in terms of energy consumption and delivery time. We have proposed a scheduling scheme that perfectly matches with any location-aware routing protocol, hence obtaining a fully functional protocol for sensor networks. We have shown that a simple algorithm can avoid any collision when the sensors know their own location and when they are synchronized. Actually we have proposed a powerful framework able to emulate any protocol based on grid structures for random instances of sensors. This can be used as a rule of thumb, that is, instead of solving problems on random sensors, we solve the problems on grid networks and adapt the obtained solutions to the random instances. We have also shown the strength of the proposed framework by solving basic problems like *Broadcast*, *Gossiping* and *Leafy Trees*.

References

1. Al-Karaki, J. N., and Kamal, A. E. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications 11*, 6 (2004), 6–28.

2. Alon, N., Bar-Noy, A., Linial, N., and Peleg, D. A lower bound for radio broadcast. *Journal on Computer and System Sciences 43*, 2 (1991), 290–298.
3. Barriere, L., Fraigniaud, P., and Narayanan, L. Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *Proc. of the 5th Int. Workshop on Discrete algorithms and methods for mobile computing and communications (DIALM)* (2001), ACM Press, pp. 19–27.
4. Bulusu, N., Heidemann, J., and Estrin, D. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications 5* (2000).
5. Farley, A. M., and Hedetniem, S. T. Broadcasting in grid graphs. In *Proc. of the 9th S-E Conf. combinatorics, graph theory, and computing* (1978), pp. 275–288.
6. Farley, A. M., and Proskurowski, A. Gossiping in grid graphs. *Journal of Combinatorics, Information and System Science 5*, 2 (1980), 161–172.
7. Friedman, R., and Korland, G. Timed grid routing (tigr) bites off energy. In *Proc. of the 16th ACM Int. Symp. on Mobile ad hoc networking and computing (MobiHoc)* (2005), pp. 438–448.
8. Klasing, R., Lotker, Z., Navarra, A., and Perennes, S. From Balls and Bins to Points and Vertices. In *Proc. of the The 16th Annual Int. Symp. on Algorithms and Computation (ISAAC)* (2005), vol. 3827 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 757–766.
9. Kortsarz, G., and Peleg, D. Approximation algorithms for minimum time broadcast. In *Proc. of the Symp. on Theory of computing and systems (ISTCS)* (1992), Springer-Verlag, pp. 67–78.
10. Kowalski, D. R., and Pelc, A. Deterministic broadcasting time in radio networks of unknown topology. In *Proc. of the 43rd Symp. on Foundations of Computer Science (FOCS)* (2002), IEEE Computer Society, pp. 63–72.
11. Kozma, G., Lotker, Z., Sharir, M., and Stupp, G. Geometrically aware communication in random wireless networks. In *Proc. of the 23rd Annual ACM Symp. on Principles of distributed computing (PODC)* (2004), pp. 310–319.
12. Lu, H., and Ravi, R. The power of local optimization: Approximation algorithms for maximum-leaf spanning tree. In *Proc. of the 30th Annual Allerton Conf. on Communication, Control and Computing* (1992), pp. 533–542.
13. Malpani, N., Welch, J. L., and Vaidya, N. H. Leader election algorithms for mobile ad hoc networks. In *Proc. of ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)* (2000).
14. McCann, J. A., Navarra, A., and Papadopoulos, A. A. Connectionless Probabilistic (CoP) routing: an efficient protocol for Mobile Wireless Ad-Hoc Sensor Networks. In *Proc. of the 24th Int. Performance Computing and Communications Conf. (IPCCC)* (2005), pp. 73–77.
15. Niculescu, D., and Nath, B. Ad Hoc Positioning System (APS). In *Proc. of the 44th IEEE Global Telecommunications Conf. (GLOBECOM)* (2001).
16. PalChaudhuri, S., Saha, A. K., and Johnson, D. B. Adaptive clock synchronization in sensor networks. In *Proc. of the 3rd international Symp. on Information processing in sensor networks(IPSN)* (2004), ACM Press, pp. 340–348.
17. Rappaport, T. S. *Wireless communications: principles and practice.* Prentice-Hall, Englewood Cliffs, NY, 1996.
18. Shor, P. W., and Yukich, J. E. Minimax Grid Matching and Empirical Measures. *The Annals of Probability 19*, 3 (1991), 1338–1348.
19. Xu, Y. An $O(n^{1.5})$ deterministic gossiping algorithm for radio networks. *Algorithmica 36*, 1 (2003), 93–96.