Algorithmica (2007) 49: 412–446
DOI 10.1007/s00453-007-9092-8

Localized Techniques for Broadcasting in Wireless Sensor Networks

Devdatt Dubhashi · Olle Häggström ·
Lorenzo Orecchia · Alessandro Panconesi ·
Chiara Petrioli · Andrea Vitaletti

Received: 13 November 2004 / Accepted: 7 June 2005 / Published online: 31 October 2007
© Springer Science+Business Media, LLC 2007

Abstract In this paper we tackle the problem of designing simple, localized, low energy consuming, reliable protocols for one-to-all communication in large scale wireless sensor networks. Our first proposed technique, called the *Irrigator* protocol, relies on the idea to first build a *sparse overlay network*, and then flood over it. The overlay network is set up by means of a simple, distributed, localized probabilistic protocol and spans all the sensor nodes with high probability. Based on the algorithmic ideas of the Irrigator protocol we then develop a second protocol, dubbed *Fireworks*, with similar performance that does not require any overlay network to be set up in advance. Asymptotic analytical results are provided which assess the reliability of the Irrigator and Fireworks techniques. The theoretical analysis of the proposed protocols is complemented and validated by a (simulation based) comparative performance evaluation that assesses several advantages of our new protocols with respect to gossiping and simple flooding. Differently from previous studies, we analyze and demonstrate the performance of our protocols for two different node distributions: The typical uniform distribution and a newly defined *"hill" distribution*, here introduced to capture some of the important and more realistic aspects of

D. Dubhashi · O. Häggström · L. Orecchia · A. Panconesi · C. Petrioli · A. Vitaletti (✉)
DJS, University of Rome, via Salaria 113, Rome 00198, Italy
e-mail: vitale@dis.uniroma1.it

D. Dubhashi
e-mail: dubahshi@cs.chalmers.se

O. Häggström
e-mail: olleh@cs.chalmers.se

L. Orecchia
e-mail: orecchia@princeton.edu

A. Panconesi
e-mail: ale@di.uniroma1.it

C. Petrioli
e-mail: petrioli@di.uniroma1.it

node deployment in heterogeneous terrain. Simulation results show that the proposed schemes achieve very good trade-offs between low overhead, low energy consumption and high reliability. In particular, the Irrigator and Fireworks protocols are more reliable than gossiping, and significantly reduce the number of links along which a message is sent over both flooding and gossiping.

Keywords Broadcasting · Gossiping · Sensor networks · Ad hoc networks

1 Introduction

Network broadcasting concerns the dissemination of a message from a given *source node* to *all* the other nodes in the network. Because of the very nature of networks, broadcast protocols should be distributed, localized, reliable and, in case of networks with limited node resources, they should also be resource-efficient. In particular, broadcast reliability is a central issue, given that it is typical of distributed environments to be subject to various kinds of faults.

Broadcasting is one of the most fundamental problems in distributed computing, taking new forms as new types of networks make their appearance. In this paper we are concerned with the broadcast problem in *wireless sensor networks* (WSNs). Sensor networks can potentially carry out a variety of useful tasks, such as environmental monitoring (e.g., seismic activity, wildlife monitoring, physical plants control, etc.), and are destined to become pervasive. WSNs are usually made of small devices, the *sensors*, that are distributed over a given area for performing a wide variety of measurements of the environment surrounding the nodes. The measurements are collected at special nodes called *sinks*, which are more powerful devices where sensed data are processed. Sinks also act as gateways to the external world. This kind of data collection implies a many-to-one communication, from the nodes to the sink. The “opposite” communication flow, one-to-many or one-to-all is also very typical of WSNs, since it models the communication from the sinks to (part of) the sensors through which the sinks communicate to the nodes the kind of data they are interested in (called simply *sink's interests*). Thus, an efficient and reliable implementation of the broadcast primitive is a basic building block of WSNs. Given that the sensor nodes have severe limitations in terms of energy, memory and computing capabilities, a broadcast protocol should be energy efficient and simple. The key to achieve this goals is to have the protocol executed at each node (distributed) relying only on local information (localized), to be gathered inexpensively.

In this paper we propose broadcast protocols for WSNs that are simple, distributed, localized and energy efficient, and we show via precise mathematical analysis and thorough simulations that the proposed protocols are expected to perform well in realistic scenarios.

The problem of wireless broadcast has been extensively studied. Description of problem, solutions and further references can be found, for instance, in [30] and [27]. To appreciate the necessity of efficient broadcast, it is worth mentioning how major solutions for ad hoc routing rely on this communication primitive. Broadcast is used for efficient route discovery in protocols such as AODV [24], DSR [15], LAR [17] and DREAM [5]. All these routing solutions use *flooding*: Starting from the source, every node that receives the message *for the first time* forwards it to its neighbors. Some heuristic optimizations are added on top of this basic scheme [27, 30]. If the network is connected this process delivers the message to every node in the network. The communication cost of flooding, however, is typically too high. The so-called "broadcast storm" resulting by the flooding can even result in harmful bandwidth congestion. This problem has been observed to be non-negligible for ad hoc routing, and it is naturally exacerbated by the limited node resources in WSNs. Therefore, one-to-all data dissemination in WSNs has been investigated as a problem per se to produce alternative approaches to ad hoc broadcast. One very popular alternative is the so-called *vertex-based gossiping*: starting from the source, every node that receives the message for the first time forwards it to its neighbors with probability p. A related approach is *edge-based gossiping*: starting from the source, every node that receives the message for the first time forwards it to each neighbor with probability p. That is, when a node receives the message, a coin is flipped for every neighbor. Note that flooding is the limit case of gossiping when $p = 1$. Randomized gossiping is now recognized as a main component of large scale distributed systems combining efficient communication with a reasonably good level of robustness (see, among others, [2, 3, 6–8, 10, 16, 29]). Hass, Halpern and Li [14] argue that randomized gossip can be used to significantly increase efficiency by reducing the number of messages sent by up to 35%. Parchuri et al. [22] give a deterministic broadcast scheme based on a geometric covering problem that they claim to be fairly superior to gossip. However, their scheme requires fixed, specially chosen locations for nodes and hence is rather inflexible and inappropriate for the dynamic scenarios of WSNs.

The contribution of this paper is proposing and analyzing alternative strategies to flooding and gossiping which are specifically designed for WSNs. Our first proposed technique, called *Irrigating*, consists on simple flooding via a *sparse overlay network* that covers all nodes and that can be set up inexpensively, efficiently and reliably. The overlay network is set up by means of a simple, distributed, localized probabilistic protocol and spans all the sensor nodes *with high probability*. Here, "with high probability" means that the probability tends to 1 as the number of nodes in the network grows, and, as we prove formally, the convergence is fast. Based on the algorithmic ideas of the Irrigator protocol we then develop a protocol for broadcasting with similar performance that does not require any overlay network to be set up in advance. We name this broadcast protocol the *fireworks protocol* because of the way the broadcasting traverses the network.

The theoretical analysis of the proposed protocols is complemented and validated by a (simulation based) comparative performance evaluation that assesses several advantages of our new protocols with respect to gossiping and simple flooding. Differently from previous studies, we analyze and demonstrate the performance of our protocols for two different node distributions: The typical uniform distribution of the nodes in the deployment area and also a newly defined *"hill" distribution*, here introduced to capture some of the important and more realistic aspects of node deployment in heterogeneous terrain.

A more precise description of our contributions follows.

Wireless sensor networks are here modelled by *geometric random graphs*. We assume that n identical sensors are distributed within the area of interest and that they have the same *transmission radius* r. For simplicity of description, we take the geographical region to be the unit square $[0, 1]^2$, and also make the standard probabilistic model assumption that the positions of the n nodes are random: independent and uniformly distributed on $[0, 1]^2$. We call *visibility graph*, denoted as G_r^n, the network topology graph obtained by drawing an edge between any two nodes whose Euclidean distance is $\leq r$. G_r^n contains all communication links that can potentially be set up or used in the network.

Definition 1 Fix $r > 0$ and a positive integer c. We take $G_{r,c}^n = (V_{r,c}^n, E_{r,c}^n)$ to denote the geometric random graph defined as follows.

- The vertex set $V_{r,c}^n$ consists of n points, picked independently according to the uniform distribution on $[0, 1]^2$.
- Each node $v \in V_{r,c}^n$ connects to c nodes chosen uniformly at random among those within distance r. If the number of such nodes is less than c then v connects to all of them. This is done independently for all nodes v.

(Each link is bidirectional, and the resulting subgraph is undirected.)

This definition embodies a very simple, distributed and localized algorithm to compute a sparse overlay (sub)network of the visibility graph. In a synchronous environment the running-time is constant, and yet, as we prove in this paper, G_{rc}^n is connected with high probability. We remark that the asymptotic result holds for $c \geq 2$ and that our experimental results confirm that in realistic scenarios connectivity does obtain for such small values of c ($c \geq 3, 4$). Also note that $G_{r,c}^n$ can be computed in a completely asynchronous fashion. This feature is quite relevant for changing environments like WSNs, where nodes come and go as they "wake up and fall asleep."

Our new Irrigator protocol is simply this:

$$\text{Flood through } G_{rc}^n.$$

This approach to data dissemination is expected to have a variety of applications. A first example has been given in the context of building up networks of Bluetooth devices where generating sparse overlays was shown to be effective in the context of *scatternet formation* [9, 18]. As we argue in this paper by means of both theoretical and empirical results, irrigating is very beneficial for WSNs. In particular, we show that it compares favorably to the popular flooding and gossiping approaches. We observe that, due to its simplicity, the protocol is not only easy to implement, but also quite efficient in terms of energy and communication.

Let us now describe precisely the analytical results that we prove in this paper.

Definition 2 Fix a positive integer c. We take $T_{r,c}^n$ to denote the random subgraph of G_r^n defined by the following process. At the beginning a root u is *captured* (selected) at random. From then on, every captured node v captures c other nodes by selecting them uniformly at random among those within distance r. If the number of such nodes is less than c then v connects to all of them. Each captured node selects the c neighbors only once, the first time that it is captured.

Although $T^n_{r,c}$ might contain cycles we shall refer to it as a tree because the process resembles a tree of out-degree c growing from the root. Note that one can view $G^n_{r,c}$ as the union of upto n such trees. Our first result says, roughly, that $T^n_{r,c}$ for large n is a giant component. We write **P** for probability (and will sometimes for clarity write $\mathbf{P}_n$ rather than **P** to emphasize that the probability model depends on the number of nodes n).

Definition 3 Let $s \in (0, 1]$ be some fixed constant. An *s-giant component* of an undirected graph G with n vertices is a connected subgraph of G containing at least ns vertices.

Proposition 1 *Fix $r > 0$ and $c \geq 2$. Then there exists a constant $s > 0$ such that,*

$$\lim_{n\to\infty} \mathbf{P}(T^n_{r,c} \textit{ is an } s\textit{-giant component}) = 1. \tag{1}$$

The emergence of a giant component is interesting and potentially useful, especially since $T^n_{r,c}$ is an on-line process, i.e., it can be generated at any moment by any node acting as a root.

Proposition 1 is also a natural step toward proving the main theoretical result of this paper.

Theorem 1 *Fix $r > 0$ and $c \geq 2$. Then*

$$\lim_{n\to\infty} \mathbf{P}(G^n_{r,c} \textit{ is connected}) = 1. \tag{2}$$

Note that this result is *not* trivial: in a related model, the *nearest neighbours* model where each node selects the *c nearest* neighbours, this assertion is *false* for constant c. For connectivity in this model, one needs $c = \Omega(\log n)$, [12, 33].

Although we stated our results in terms of limits, both statements hold with high probability. The probability of the complementary events goes to 0 as $\Theta(n^{-\epsilon})$ for some $\epsilon \in (0, 1)$. Finally, we will show that *no* vertex has degree exceeding $\frac{\alpha \log n}{\log\log n}$ (for some constant $\alpha > 0$) with high probability. For networks of realistic size, this value does not exceed a small constant.

One possible drawback of the Irrigator strategy is the fact that it needs the overlay infrastructure, namely, G^n_{rc}, to be set up in advance. Proposition 1 is also the conceptual basis to understand the following random process, the *fireworks protocol*, denoted as $F^n_{cpr}(u)$. The process starts from the root (i.e. the source of the broadcast) u that sends the message to all neighbors. Then the following is repeated: when a node sees the message for the first time it either forwards the message to c random nodes within distance r with probability p or, with probability $1 - p$, it forwards the message to all nodes within distance r. We will show that for values of p as small as $\Theta(\log^* n/n)$,[1] fireworks reaches all nodes with probability 1, as n goes to infinity.

[1] The function $\log^* n$ is the number of times one needs to apply the logarithm successively to reduce n to at most 1. It is an extremely slow growing function, for instance, $\log^* 2^{32} = 4$.

Note that the firework protocol is an on-line process that can be generated by the source, like gossiping, and it does not require an overlay network. In this paper we show that fireworks performs better than gossiping under several relevant metrics.

In this paper we also show that gossiping reaches all nodes with probability 1, as n goes to infinity. Thus, gossiping, irrigating and the firework protocol are reliable in the limit, i.e. for large n they reach all nodes with probability ~ 1. It becomes then interesting to see if an experimental evaluation can sharpen the analysis. The main conclusion of our experimental evaluation is that indeed these processes differ significantly.

- It is more likely that G^n_{rc} is connected than gossiping reaches all nodes in the network. Thus the Irrigator is a more reliable protocol than gossiping, whether vertex or edge based. The same conclusion holds for the firework protocol.
- The number of links along which a message is sent by gossiping is much greater than the number of edges of G^n_{rc} or F^n_{cpr}. Thus, in all situations where the cost of a message can be charged to the edge connecting the two nodes, the proposed protocols are much more efficient.

In other words, since in WSNs the energy cost of sending a message is roughly equal to that of receiving it, if we implement a broadcast service by means of the reliable unicast primitive (such as the one provided by IEEE 802.11 DCF), irrigator and fireworks are not only more reliable, but also more energy efficient than gossiping. An apparently cheaper option would be that of implementing a broadcasting service by means of the local-broadcast primitive (in the IEEE 802.11 standard a local-broadcast message is received by all nodes within transmission range from the sender). Besides the drawback of using this unreliable primitive, one have to consider that WSNs are dynamic networks. In order to save energy, nodes periodically go asleep [4, 32]. When they wake up they have to be informed of relevant events, such as new messages broadcasted from the sinks. In such dynamic scenario vertex-based gossiping protocols must retransmit several times to allow nodes to reach all their neighbors. For this reason vertex-based gossiping loses the competitive advantage that local-broadcasting might give in other scenarios. In WSNs nodes will typically have to know their immediate neighbors and their wake-up schedules so that they transmit when the recipients are awake. Relying only on the needed one-hop neighborhood knowledge, our protocols give raise to low-overhead solutions. Indeed, our comparative simulation results show that the on-line fireworks scheme and the irrigator protocol offer a remarkably good compromise between energy saving and reliability of the broadcast when compared to flooding and gossiping.

The paper is organized as follows. In Sect. 2 we discuss the previous work. In Sect. 3 we prove the analytical results. In Sect. 4 we discuss implementation issues, showing that our protocols can be implemented easily and inexpensively. Also a variant of the irrigator protocol which leads to further improvements in practical scenarios is described. In Sect. 5 we discuss the outcome of our extensive simulations. Finally we conclude the paper in Sect. 6.

2 Related Works

In this section we review the major solutions that have been proposed in the literature for broadcasting in multi-hop wireless ad hoc networks. In [30] a taxonomy of the different solutions has been reported. The authors divide the different schemes in four groups:

- Simple Flooding;
- Probabilistic-based schemes [19, 21], which make use of some basic understanding of the network topology to assign to a node a probability p to rebroadcast;
- Area-based methods, which exploit location awareness to estimate the additional coverage associated with a node re-broadcasting the message. Only if the coverage is significantly enlarged the node retransmits the message (see for example [21]);
- Neighbor knowledge methods. This category comprises schemes in which two hop neighborhood knowledge is exploited to identify whether re-broadcasting allows to reach new nodes or not [19, 23, 25]. Only in the former case a node retransmits.

A comparative performance evaluation of the major different solutions for network wide broadcasting has been performed in [34].

A taxonomy similar to the one in [30] is reported in a recent work by Stojmenovic and Wu [27]. In this case, the solutions proposed in the literature are grouped based on whether the protocol is probabilistic or deterministic, on the amount of information on the network topology needed by the protocol to operate, the amount of extra information exchanged between nodes during the protocol operation, and the schemes' reliability (defined as the capability of a broadcasting protocol to successfully reach all the nodes in the network). In other words, solutions are classified based on performance related criteria such as the protocols overhead, complexity and reliability. Apart from the solutions listed in [30] the authors introduce cluster-based schemes such as [28, 31] for sake of broadcasting. In this case a subset of sensor nodes is first selected to build a connected backbone made of so called Backbone Nodes (BN) and gateways chosen for sake of BN interconnection. At the end of this phase each node is either in the backbone, or is an ordinary node within one hop from a backbone node. Broadcasting can thus be performed by the source node sending the message to a one-hop neighbor in the backbone, which floods it over the (sparser) backbone to the final destination. Rules for BN selection and for gateways identification guarantee that all nodes are reached by a broadcast whenever the original network topology was connected. The cost to pay is in the overhead needed for sake of backbone formation and for backbone maintenance. Backbone reorganization might be triggered by nodes mobility, by nodes dying because of energy depletion, or simply be motivated by the need to load-balance the resource consuming role of backbone node among all the nodes in the network. Different schemes have been proposed in the literature for clustering and backbone formation resulting in denser or sparser backbone topology and in more or less overhead for sake of backbone formation and reorganization.

In this paper we are concerned with designing *localized* techniques for network wide broadcasting without assuming any location awareness. Our solutions are localized in the sense that we keep to a minimum the neighborhood knowledge at each node (no more than the one hop neighbors), as well as the information that have to be exchanged for performing broadcasting, resulting in very lightweight solutions. We do not consider protocols that require backbone formation and maintenance due to their associated overhead.

3 Connectivity Results

We begin by showing that T^n_{rc} covers a giant component, and then that the union of these trees is connected, all with high probability. In the last sub-section we show that firework and gossip also span the whole network with high probability.

3.1 Giant Component

For the purpose of proving Proposition 1, and also later, Lemma 3.1 below will be useful. Fix an integer k such that

$$k > \frac{\sqrt{5}}{r} \tag{3}$$

and partition $[0, 1]^2$ into k^2 subsquares of size $\frac{1}{k} \times \frac{1}{k}$ in the obvious way. One point of this choice of k is that it ensures that any two points sitting in adjacent subsquares are within distance r from each other; this will be needed in the proof of Theorem 1.

Lemma 3.1 *Let $k > 0$ be fixed and let $\mathcal{E}_{kn}$ be the event "each of the k^2 subsquares contains at least $\frac{n}{2k^2}$ points". Then*

$$\lim_{n\to\infty} \mathbf{P}_n(\mathcal{E}_{kn}) = 1. \tag{4}$$

Proof Fix a square S and let X denote the number of points in S. Then, $\mu := EX = \frac{n}{k^2}$ and, by the Chernoff-Hoeffding bound,

$$\mathbf{P}_n\left(X < \frac{n}{2k^2}\right) \le e^{-n/8k^2}.$$

Thus, the probability that some square has less than the required number of points is at most $k^2 e^{-n/8k^2}$. □

To investigate connected components of the irrigator graph, we shall employ the following method, which we will call the *sequential discovery procedure* (this is simply a breadth-first exploration). First, select a node v_0 at random (among all n nodes). Then consider the c edges chosen by v_0 (in the device discovery procedure of Definition 1), and denote the endpoints (other than v_0) of these edges by $v_1, \dots, v_c$. Then continue with the edges chosen by v_1, and so on, in a breadth first search manner. Each time a new node is encountered, the node reached by it is included in our list of nodes, and the choice is deemed a *success*. Sometimes, the edge leads to a node already seen in this procedure, in which case the choice is said to be a *failure*. At any point of this search procedure, we may stop, and those vertices encountered whose outgoing edges have not been investigated (yet), are called *fresh* nodes.

At various stages of our arguments, we will invoke a comparison between the sequential discovery procedure and a (Galton–Watson) *branching process*. Such a branching process (see, e.g., Harris [13] or Asmussen and Hering [1]) begins with m_0 individuals. Each of these begets, independently of the others, a number of offspring, which has some given distribution f on the non-negative integers. Each of these children then has a number of children for itself, again independently with distribution f. And so on, again in a BFS manner. One of two things will happen: either the branching process dies out after a finite number of generations, or it survives (forever). Excluding the trivial case where f puts unit mass on 1, it is well known that the branching process has positive probability of surviving if and only if f's first moment is strictly greater than 1.

We shall be particularly concerned with a branching process whose offspring distribution is the binomial distribution $\mathrm{Bin}(2, p)$. This can be compared to the sequential discovery procedure for $c = 2$ in the following way. Suppose that we can show that up until some given stage S of the sequential discovery procedure, each choice of a new node to connect to has probability at least p (conditionally on everything seen so far) of being a success. Then we can make a joint construction (a so-called coupling; see [20]) of the sequential discovery procedure and the $\mathrm{Bin}(2, p)$ branching process in such a way that each individual in the branching process corresponds to a particular node (not shared by any of the other individuals of the branching process) of the sequential procedure, up until the given stage S. We say in this case that the sequential procedure up until stage S *stochastically dominates* the branching process (for stochastic domination see for example [20]). We will show that the sequential discovery procedure first generates almost surely a set of $\log n$ points and that from then on each point u begets offsprings with distribution $\mathrm{Bin}(2, p_u)$, with $p_u \geq \frac{3}{4}$. It follows from a standard application of stochastic domination that the survival probability of the sequential discovery procedure is at least that of $\log n$ independent branching processes with distribution $\mathrm{Bin}(2, \frac{3}{4})$.

Proof of Proposition 1 We prove the result for $c = 2$ only, which is obviously enough since adding edges is not going to destroy a giant component.

Run the sequential discovery procedure until the outgoing edges of $\log(n)$ nodes are investigated (or until there are no more fresh nodes, in which case we are stuck).

By Lemma 3.1, we may assume that the event in (4) happens, and condition on that event. By the choice (3) of k, this means that each time a node selects another node to connect to, there are at least $\frac{n}{2k^2}$ nodes to choose from. And each time, there are at most $2\log(n)$ nodes that have already been seen, so each edge has probability at most

$$\frac{4k^2 \log(n)}{n} \tag{5}$$

of hitting a node that has already been seen. Hence, the probability that *at least one* of the $2\log(n)$ choices is a failure, is at most

$$2\log(n)\frac{4k^2 \log(n)}{n} = \frac{8k^2(\log(n))^2}{n}, \tag{6}$$

which tends to 0 as $n \to \infty$. Hence, the probability that *all* choices, up until the outgoing edges of $\log(n)$ nodes are investigated, are successful, tends to 1 as $n \to \infty$.

Hence, we have shown that with probability approaching 1 as $n \to \infty$, we get a connected component with at least $2\log(n)$ nodes. But this is not enough to prove Proposition 1, which asserts a component whose size is *linear* in n.

We can, however, continue the sequential discovery procedure from the $\log(n)$ fresh nodes that we have (assuming that all choices so far have been successful). Let us continue the sequential procedure until the stage S when either a total of $\frac{n}{8k^2}$ nodes have been found (or no fresh nodes remain). Before stage S, each new discovery has, by an analogous argument as that used to establish (5), probability at most

$$\frac{n/8k^2}{n/2k^2} = \frac{1}{4}$$

of not being successful. It follows that the sequential discovery procedure starting from the $\log(n)$ nodes until stage S stochastically dominates a $\mathrm{Bin}(2, \frac{3}{4})$ branching process with the same initial number of individuals. Consider the following events:

A: *"The sequential procedure fails to survive until $n/8k^2$ nodes are found"*;
B: *"A* $\mathrm{Bin}(2, \frac{3}{4})$ *branching process starting with* $\log(n)$ *individuals dies out"*;
C: *"A* $\mathrm{Bin}(2, \frac{3}{4})$ *branching process starting with* 1 *individual dies out"*.

We therefore get, conditionally on no failures associated with the first $2\log(n)$ nodes,

$$\begin{aligned}\mathbf{P}(\mathrm{A}) &\le \mathbf{P}(\mathrm{B}) \\ &= \mathbf{P}(\mathrm{C})^{\log(n)} \\ &= (1-\alpha)^{\log n}, \end{aligned} \tag{7}$$

where $\alpha > 0$ is the survival probability of a $\mathrm{Bin}(2, \frac{3}{4})$ branching process starting from a single individual (an easy calculation shows that $\alpha = \frac{8}{9}$, but we only need the fact that $\alpha > 0$, which follows from the fact that the offspring distribution has expectation $\frac{3}{2} > 1$). The sum of (6) and (7) tends to 0 as $n \to \infty$, whence (1) follows with $s = \frac{1}{8k^2}$, and we are done. □

Note that by Lemma 3.1, (6) and (7) the probability of not having a giant component is $\Theta(n^{-\epsilon})$ for $\epsilon > 0$.

3.2 Connectivity

In this section we go on to prove the connectedness of $G^n_{r,c}$ asserted in Theorem 1. We begin by proving the following strengthening of Proposition 1.

Proposition 2 *Fix $r > 0$ and $c \ge 2$. Then there exists a constant $s > 0$ such that*

$$\lim_{n\to\infty} \mathbf{P}(\textit{every node of } G^n_{r,c} \textit{ is in some } s\textit{-giant component}) = 1.$$

Proof Again, it suffices to consider $c = 2$. As in the previous section, let α denote the survival probability of a $\mathrm{Bin}(2, \frac{3}{4})$ branching process starting from a single individual.

We proceed using the sequential discovery procedure as in the proof of Proposition 1, with the following modification. Instead of initially running it until the outgoing edges of $\log(n)$ nodes have been checked, run it until the outgoing edges of $a\log(n)$ nodes have been checked, where a is a fixed number chosen so that

$$a > \log\left(\frac{1}{1-\alpha}\right).$$

The estimate in (6) then becomes replaced by $\frac{8k^2a^2(\log(n))^2}{n}$. However, since the result we are trying to prove concerns all n points simultaneously, we need to improve on this estimate (which, when multiplied by n, fails to approach 0). To do this, note we can afford to have *one* failed edge during the discovery of the outgoing edges of the first $a\log(n)$ nodes without very much damage (there will still be $a\log(n)$ fresh edges at the end of this search). To estimate the probability that at least two choices fail, note that there are less than $\frac{(a\log(n))^2}{2}$ pairs of times during the procedure at which the choices can fail, and for each such pair the probability of failure in both is at most $(\frac{2a\log(n)}{n/2k^2})^2$ (assuming as before the event in Lemma 3.1). The probability that at least two of the $2a\log(n)$ choices are failures is therefore at most

$$\frac{(a\log(n))^2}{2}\left(\frac{2a\log(n)}{n/2k^2}\right)^2 = \frac{8k^4a^4(\log(n))^4}{n^2}, \tag{8}$$

which tends to 0 at a rate which (as we shall see) is fast enough for our purposes.

Again imitating the proof of Proposition 1, we go on to run the sequential discovery procedure until a total of $\frac{n}{8k^2}$ nodes have been found. Let A denote the event *"the sequential procedure fails to survive until $\frac{n}{8k^2}$ nodes are found"*. Since we begin with $a\log(n)$ fresh nodes, the analogue of (7) becomes

$$\begin{aligned}\mathbf{P}(\mathrm{A}) &\le (1-\alpha)^{a\log(n)}\\ &= n^{-b},\end{aligned} \tag{9}$$

where $b = -a\log(1-\alpha)$, and $b > 1$ by the choice of a.

On the event in Lemma 3.1 (whose probability tends to 1), we can bound the probability that *some* node fails to sit in an s-giant component (with $s = \frac{1}{8k^2}$) by adding the estimates in (8) and (9) and multiplying by the number of nodes n. This yields

$$\frac{8k^4a^4(\log(n))^4}{n} + n^{1-b}, \tag{10}$$

which still tends to 0 as $n \to \infty$, so the proof is complete. □

Proof of Theorem 1 As usual, we need only consider the $c = 2$ case. Let A denote the event *"$G^n_{r,c}$ contains at least two distinct $\frac{1}{8k^2}$-giant components"*. Note that in view of Proposition 2 with the estimate

$$s = \frac{1}{8k^2} \tag{11}$$

that comes out of its proof, the only thing that can cause (2) to go wrong is if there exists an $\varepsilon > 0$ such that

$$\limsup_{n\to\infty} \mathbf{P}(A) \geq \varepsilon. \tag{12}$$

Now consider the experiment of generating $G^n_{r,c}$ and then picking two of its nodes at random; let A denote the event that these two nodes end up in the same connected component. By conditioning on the first of these nodes, we see that (12) implies that

$$\limsup_{n\to\infty} \mathbf{P}(\neg A) \geq \frac{\varepsilon}{8k^2}.$$

In order to prove the theorem, it therefore suffices to show that

$$\lim_{n\to\infty} \mathbf{P}(\neg A) = 0. \tag{13}$$

Let us denote the two nodes chosen at random by v_0 and v_1. By Proposition 2 and the estimate (11), we may assume that v_0 is in a connected component of at size least $\frac{n}{8k^2}$. Then, by the pigeonhole principle, at least one of the k^2 subsquares of $[0, 1]^2$ introduced in Sect. 3.1 contains at least $\frac{n}{8k^4}$ nodes of that connected component. Let us pick such a subsquare and denote it by B.

Next, fix an integer m, and run the sequential discovery procedure starting from the other node v_1, with the following restriction. As soon as an edge fails to lead to a new node, we give up. Assuming this does not happen, we run the procedure until the outgoing edges of exactly $m - 1$ nodes have been investigated; this leaves us with exactly m fresh nodes. From this stage on, check only *one* of the two edges leading out of each node (this edge is chosen at random among the two), and we continue this for k^2 steps from each the m fresh nodes, and then stop. This means that we check a total of $m - 1 + mk^2$ edges. The probability that any of these is a failure tends to 0 as $n \to \infty$ (this follows from (6)) and can therefore be ignored.

Let $w_1, \ldots, w_m$ denote the fresh nodes after having checked the outgoing edges of $m - 1$ nodes in the sequential procedure. Pick one of these vertices, w_i, and denote the subsquare it sits in by $B_{i,0}$. We can then find a sequence of subsquares $B_{i,1}, B_{i,2}, \ldots, B_{i,\ell}$, $\ell \leq 2k$, such that

(i) for each $j \in \{0, 1, \ldots, \ell - 1\}$, the subsquares $B_{i,j}$ and $B_{i,j+1}$ are adjacent, and
(ii) $B_{i,\ell} = B$.

Fix such a sequence, and consider the "naked-branch" sequential discovery procedure starting from w_i, and denote the nodes found along this branch by $w_{i,1}, w_{i,2}, \ldots, w_{i,\ell}$. Given the event in Lemma 3.1 (which we may assume happens), the probability that $w_{i,1}$ ends up in $B_{i,1}$ is at least $\frac{n/2k^2}{n} = \frac{1}{2k^2}$ (due to our choice (3) of k). Given that, the conditional probability that $w_{i,2}$ ends up in $B_{i,2}$ is at least $\frac{1}{2k^2}$. And so on. Finally, given that $w_{i,\ell-1}$ is in $B_{i,\ell-1}$, the conditional probability that $w_{i,\ell}$ is in the connected component of v_1, is at least $\frac{n/8k^4}{n} = \frac{1}{8k^4}$. Multiplying these conditional probabilities yields that $w_{i,\ell}$ has probability at least

$$\left(\frac{1}{2k^2}\right)^{\ell-1} \frac{1}{8k^4} \geq \left(\frac{1}{2k^2}\right)^{2k-1} \frac{1}{8k^4} \tag{14}$$

of being in the connected component of v_0.

On the event that no checked edges result in failures (which we assume), the m different "naked branches" move independently, so (14) implies that the probability that none of them hit the connected component of v_0 is at most

$$\left(1 - \left(\frac{1}{2k-1}\right)^{k-1} \frac{1}{8k^4}\right)^m.$$

We have thus shown that

$$\limsup_{n\to\infty} \mathbf{P}(\neg A) \leq \left(1 - \left(\frac{1}{2k^2}\right)^{2k-1} \frac{1}{8k^4}\right)^m. \tag{15}$$

Now, m was arbitrary, and the right hand side of (15) can be made as small as we wish by picking m large. Hence (13) is established and the proof is complete. □

A careful examination of the estimates of failure probabilities in the proof above show that the probability of not having connectivity is at most $\Theta(n^{-\epsilon})$, for $\epsilon > 0$.

Finally, we will now show that no vertex in the graph $G^n_{r,c}$ has very high degree.

Proposition 3 *For any integer $t \geq 1$, there is a constant $\alpha > 0$ such that no vertex in $G^n_{r,c}$ has degree exceeding $\alpha \frac{\log n}{\log \log n}$ with probability $1 - O(n^{-t})$.*

Proof Consider any vertex v. The vertices that could possibly connect to it lie in a circle of radius r centered at v and by Lemma 3.1, there are at most $2\pi r^2 n$ such vertices. Each such vertex has probability $1-(1-\frac{1}{n})^c \leq \frac{c}{n}$ of connecting to v. Hence, the expected (in)degree of v is at most $2\pi r^2 c$, and the result now follows by applying the Chernoff-Hoeffding bounds. □

3.3 Reliability of Firework and Gossip

In this section we prove some fundamental results on connectivity of the gossip and fireworks protocol. The main result is the following.

Theorem 2 *If $p = \frac{\log^* n}{n}$:*

$$\lim_{n\to\infty} \mathbf{Pr}(\textit{gossip reaches all nodes}) = 1. \tag{16}$$

Proof We assume that the event of Lemma 3.1 holds and condition on this event for the remaining of the proof.

Denote by B_0 the subsquare containing the source and let B be any subsquare in the partition.

We can find a subsquare sequence $B_0, B_1, \dots, B_t$, with $t \le 2k$, such that

1. for each $i \in \{1, 2, \dots, t\}$ B_i and B_{i-1} are adjacent, and
2. $B_t = B$.

For $i \in \{0, \dots, t\}$, let A_i be the event that the gossip procedure reaches all nodes in B_i and let X_i be the number of nodes in B_i which flood. Notice then that the probability of A_i is at least the probability that the procedure reaches some node in B_{i-1} and such node floods, as B_{i-1} and B_i are adjacent. Moreover such probability is at least the probability that the procedure reaches all nodes in B_{i-1} and at least one of these floods. That is, for $i \neq 0$:

$$\mathbf{Pr}(A_i) \ge \mathbf{Pr}(X_{i-1} \ge 1 | A_{i-1})\mathbf{Pr}(A_{i-1}).$$

It follows immediately that:

$$\mathbf{Pr}(A_t) \ge \left(\prod_{i=0}^{t-1} \mathbf{Pr}(X_i \ge 1 | A_i) \right) \mathbf{Pr}(A_0).$$

Consider the term $\mathbf{Pr}(X_i \ge 1 | A_i)$. As each flooding event takes place with probability p:

$$\mathbf{E}[X_i | A_i] \ge p \frac{n}{2k^2} = \frac{\log^*(n)}{n} \frac{n}{2k^2} = \frac{\log^*(n)}{2k^2}.$$

Since all flooding events are independent, we can apply a Chernoff bound to obtain the following:

$$\mathbf{Pr}\left(X_i \le \frac{\log^*(n)}{4k^2} \middle| A_i \right) \le \mathbf{Pr}\left(X_i \le \frac{\mathbf{E}[X_i | A_i]}{2} \right) \le e^{-\frac{\log^*(n)}{16k^2}}.$$

Hence, for large enough n (such that $\frac{\log^*(n)}{4k^2} \ge 1$):

$$\mathbf{Pr}(X_i \ge 1 | A_i) \ge \mathbf{Pr}\left(X_i \ge \frac{\log^*(n)}{4k^2} \middle| A_i \right) \ge 1 - e^{-\frac{\log^*(n)}{16k^2}}.$$

Moreover

$$\mathbf{Pr}(A_0) = 1$$

because the source always floods.

Finally, we have:

$$\begin{aligned} \mathbf{Pr}(A_t) &\ge \left(\prod_{i=0}^{t-1} \mathbf{Pr}(X_i \ge 1 | A_i) \right) \mathbf{Pr}(A_0) \\ &\ge \left(1 - e^{-\frac{\log^*(n)}{16k^2}} \right)^t \\ &\ge \left(1 - e^{-\frac{\log^*(n)}{16k^2}} \right)^{2k} \end{aligned}$$

which tends to 1 as n goes to infinity, showing gossip reaches all nodes in B. Then, by union bound on all subsquares, the probability that the procedure does not reach all nodes in the graph is at most

$$k^2(1 - \mathbf{Pr}(A_t))$$

which tends to 0 as n goes to infinity. □

Corollary 1 *If* $p = \frac{\log^* n}{n}$:

$$\lim_{n\to\infty} \mathbf{Pr}(\text{Fireworks } \textit{reaches all nodes}) = 1. \tag{17}$$

Proof Given the same flooding probability p for both procedure it holds that

$$\mathbf{Pr}(\textit{Fireworks} \text{ reaches all nodes}) \geq \mathbf{Pr}(\text{gossip reaches all nodes})$$

as any execution of Fireworks can be modelled as an execution of gossip, followed by the addition of other edges according to the Fireworks protocol. Such addition obviously preserves or improves connectivity, yielding the inequality. By the theorem, it follows:

$$\lim_{n\to\infty} \mathbf{Pr}(\textit{Fireworks} \text{ reaches all nodes}) \geq \mathbf{Pr}(\text{gossip reaches all nodes}) = 1. \quad \square$$

3.3.1 Estimating Overall Number of Links

Fireworks can be modelled as a procedure constructing a directed graph G. If a node u propagates a message to a node v, according to *Fireworks*, an arc is inserted from u to v. Assuming *Fireworks* reaches all nodes and denoting by $\deg_{\text{out}}$ the out degree of a node in G:

$$\mathbf{E}[\deg_{\text{out}}] \leq c(1 - p) + np \in \Theta(\log^*(n)).$$

Hence, the average overall number of arcs is:

$$\mathbf{E}[e(G)] = n\mathbf{E}[\deg_{\text{out}}] \in \Theta(\log^*(n)n)$$

which is almost sparse.

3.3.2 Note on the Choice of p

The proof above works for all choices of p such that

$$p = \Theta\left(\frac{f(n)}{n}\right) \quad \text{and} \quad \lim_{n\to+\infty} f(n) = +\infty.$$

To make this result tight, we prove the following:

Theorem 3 *If* $\limsup_{n\to\infty} f(n) \neq +\infty$, *as* n *goes to infinity some node is not reached by the procedure with constant positive probability.*

Proof We may assume the result of Lemma 4.1. Consider then a node v and its neighborhood. This set contains at least $\frac{n}{2k^2}$ nodes. Let u be one of these nodes. Suppose u is reached by the procedure. The probability that u does not propagate to v is the product of the probability that u does not flood and that u chooses c neighbors distinct from v. The latter is at least $1 - \frac{c}{n/2k^2}$ by the bound in the lemma. Hence, the probability that u does not propagate to v is at least

$$(1-p)\left(1-\frac{2ck^2}{n}\right)$$

as the events are independent. Moreover, as the procedure executes independently on all nodes in the neighborhood of v and there are at most n such nodes, the probability that no node propagates to v, i.e. v is not reached, is at least

$$\left[(1-p)\left(1-\frac{2ck^2}{n}\right)\right]^n.$$

Now, by assumption, there exists a constant $b \geq 0$ such that $f(n) \leq b$ for all n. Then the probability that v is not reached is at least

$$\left[\left(1-\Theta\frac{b}{n}\right)\left(1-\frac{2ck^2}{n}\right)\right]^n$$

which tends to at least $e^{-\Theta(b)-2ck^2}$ as n tends to infinity. This is a constant, proving the theorem. □

This implies that, in this case, on average, at least a constant fraction of nodes is not reached by *Fireworks*. A similar proof applies to gossip.

In the next section we will show that despite similar asymptotic behaviors, the firework, irrigator and gossip schemes show significantly different performance in practice.

4 Implementation Issues

In this section we discuss possible implementations of the Irrigator and Fireworks protocols proposed for network-wide broadcasting. Also a variant of the basic Irrigator protocol which leads to further energy saving will be presented.

In describing the protocols we will distinguish between virtual topology-based solutions, in which a sparse overlay is first identified and broadcasting is then implemented via Flooding over such overlay, and on-line broadcasting solutions. The implementation of the gossip protocol, selected for sake of benchmarking, will also be reviewed.

4.1 Virtual Topology-based Solutions

4.1.1 The Irrigator Protocol

In the Irrigator protocol a sparse overlay G^n_{rc} is first built by each node randomly selecting c among its neighbors in the visibility graph,[2] and then broadcasting is performed via flooding over G^n_{rc}.

Applying the flooding procedure over a much sparser overlay has the advantage that the number of traversed links is significantly reduced. This in turns decreases the number of transmitted unicast packets and the energy consumption per node (as the latter metric is directly associated to the number of messages transmitted by and addressed to each node[3]).

The Irrigator scheme can be easily implemented as follows.

- *Overlay computation.* At the protocol start up each node becomes aware of its neighbors via basic hello messages exchange. Based on such one hop neighborhood knowledge, each node selects c among its neighbors, and communicates its choice to its neighbors in the next periodic hello message. When receiving the second hello messages each node is thus able to compute the links in G^n_{rc} incident to itself (a link (u, v) is included in the overlay iff at least one of the two extremes u, v selected the other).
- *Broadcast message propagation.* Upon reception of a broadcast message a node will retransmit the message to all its neighbors in G^n_{rc} but the one from which it has received the message, in a flooding-like fashion. Flooding is however limited to the sparse overlay. Message transmission to the neighbors in G^n_{rc} can be implemented either via multiple unicast transmissions or via a local broadcast. In the latter case a node transmits the message and all its neighbors which are NOT connected to it in the overlay discard the message upon receiving it.

We note that the overhead associated with the Irrigator protocol operation can be quantified in a few extra bytes (needed to identity up to c neighbors) added in the second hello messages. Extensive simulations reported in this paper show that G^n_{rc} will be connected, for $c \geq 4$, whenever the visibility graph G_r was connected. Given the small value of c this results in almost negligible overhead.

The name of the above described protocol, 'Irrigator', captures the fact that rather than flooding the network with messages, the Irrigator scheme disseminates such messages along a much more reduced set of routes while being able to successfully reach all the nodes with high probability.

---

[2]We denote as visibility graph G_r the graph in which there is a vertex for each sensor node, and an edge between any two neighboring nodes (i.e., between the nodes within each other transmission radius).

[3]In the following we will make the approximation that energy is consumed only when receiving a packet addressed to the node. This reflects the usual practice to switch off the radio transceiver as soon as a node realizes not to be an intended destination for a given packet. The node will then go to sleep over the rest of the message transmission, thus consuming negligible power. As the node can identify whether it is an intended destination by reading only the first few bytes of the packet header, we have considered negligible the overall energy consumption associated to this operation.

Not only is the proposed solution simple, with minimal overhead, and energy saving but it is also robust in practical scenarios. In the performance evaluation section we will show that the assumption of having the nodes uniformly deployed in the area, which may appear a limit of the scheme, can be relaxed to account for more realistic nodes' deployment distributions without affecting the connectivity properties of such scheme.

4.1.2 The Irrigator Protocol, v2.0

The experimental results on the Irrigator protocol provided us with the intuition that, by inserting links in the virtual topology randomly and uniformly so that each node has c^* links incident to it (provided its degree is $\geq c^*$), the global connectivity properties are likely to be maintained. This motivated some further reasoning on ways to reduce the number of links included in G^n_{rc} by the Irrigator protocol. In such protocol when $c = 4$ the nodal degree in G^n_{rc} is likely to exceed such value since all the c neighbors selected by a node u, *plus* all the neighbors that selected u are u's neighbors in G^n_{rc}. The variance of the nodal degree in the virtual topology may force the adoption of a c value higher than what would be needed in case some form of control that all nodes achieve a minimum nodal degree in G^n_{rc} is enforced.

The following simple variant of the Irrigator protocol, denoted as Irrigator v2.0 in the following, has thus been designed.

Each node, based on its one hop neighborhood knowledge first randomly and uniformly selects c neighbors ($c < c^*$, say c could be 2 and c^* could be set to $3, 4$) and communicates this to its neighbors in the following hello message. So far the protocol operates exactly as the Irrigator protocol but with a c value much lower than what would be needed by the Irrigator protocol to result in high reliability. After gathering the second hello messages from all its neighbors, node u computes the total number of links Num_{links} incident to it in G^n_{rc}, either selected by itself or by one of its neighbors. If $Num_{links} \geq c^*$ no further link is selected. Otherwise, node u randomly and uniformly selects $c^* - Num_{links}$ among the unselected links to its one hop neighbors, and communicates the identity of the nodes selected in this second phase of the protocol in the next exchanged hello message. The idea here is to try to prevent nodes from having a highly variable nodal degree, with the rationale that a nodal degree around c^*, when links are randomly selected, is enough to enforce the maintenance of global connectivity properties.

4.2 On-Line Solutions

4.2.1 The Gossip Protocol

For sake of protocols benchmarking we implemented the gossip protocol introduced in [11]. In the following we will discuss the implementation of the two versions of gossiping presented in [11]: vertex gossip and edge gossip.

Vertex gossip is a simple probabilistic flooding-based scheme which works as follows. Whenever a source wants to broadcast a message it sends it to all its neighbors. Whenever a node receives a message it has not generated, it tosses a coin and, with probability p it retransmits the broadcast message to its neighbors (except the one from which it received the message). With probability $(1 - p)$ it stays silent. The implementation of this protocol is straightforward, either via local broadcast, or, in case the adopted awake/asleep schedule makes impossible to reach all neighbors via a few local broadcasts, via unicast packets transmitted to each of the node's neighbors. Whenever, as in typical WSN scenarios, the knowledge of a node's neighbors and of their wake up schedules is needed for each node to know when to transmit and how many times to reach all the intended recipients, such knowledge can be achieved via hello message exchanges.

The implementation of edge gossip is similar. The only difference with vertex gossip is that whenever a node receives a broadcast message it tosses a coin for each of the edges incident to it (but the one from which it has received the message), transmitting to *each* of the neighbors with probability p. This de facto implies that either unicast packets are used for sake of edge gossip implementation, or a possibly long list of intended recipients has to be included in the message in case local broadcast is adopted.

These solutions trade off the number of nodes re-broadcasting the message and the energy consumption (the lower the p value the lower the number of nodes involved in re-broadcasting the message in case of vertex gossip, the lower the number of traversed links in both the two gossip protocols) with reliability (the lower the p value, the less reliable the protocol is).

4.2.2 The Fireworks Protocols

The Fireworks protocol is an on-line scheme which combines features of the gossip protocol and of an Irrigator-like approach. The protocol works as follows. The broadcast source transmits to all its neighbors. Whenever a node receives a new broadcast message it tosses a coin. With probability p it re-broadcasts the message to all its neighbors. With probability $(1 - p)$ it sends it only to c randomly selected neighbors. The way the latter is implemented is by either transmitting the message via local broadcast, including the list of intended destinations in the message, or by sending the message to the selected neighbors via unicast packets.

With respect to the gossip protocol, our intuition, confirmed by the results summarized in the performance evaluation section, is that the Fireworks protocol results in higher reliability given the same number of links over which the broadcast packet is transmitted.

5 Performance Evaluation

This section summarizes the results of extensive simulations that have been conducted to prove the effectiveness of the proposed solutions and to quantify the improvements that can be achieved over previous schemes. Our experiments have been conducted by means of a simulator we have developed in Java. Simulations have proceeded in two phases: first we have evaluated whether, and for which parameters values, the Irrigator schemes allow to maintain the global connectivity properties of

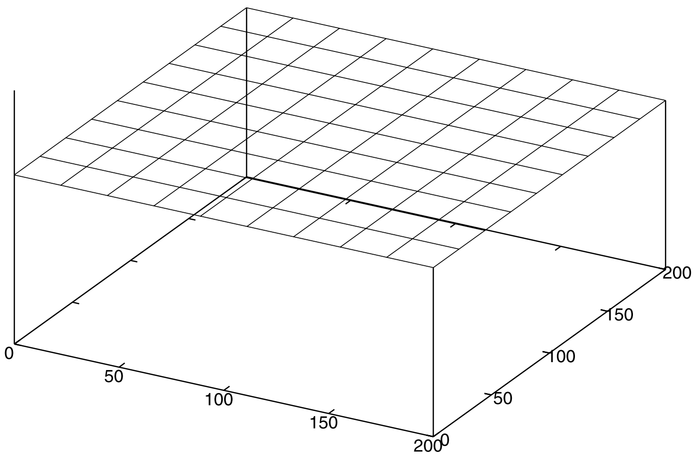


Fig. 1 Uniform distribution

the network. The size of the giant component (normalized to the number of nodes in the network), the number of connected components and the number of links in the generated virtual topologies have been evaluated under different nodes densities, and compared to the same metrics in the visibility graphs. We have then conducted simulations to compare, under different nodes deployments, and for varying nodes density, the performance of the different schemes proposed and of the gossip protocol (both vertex and edge gossip) in terms of energy consumption, channel capacity demand and reliability of the broadcast process.

In the simulated scenarios $n \leq 300$ sensor nodes, with maximum transmission radius of 30 meters, are scattered in a geographic area which is a square of side $L = 200$ m. We make the assumption that two nodes are in each other transmission range if and only if their Euclidean distance is $\leq$30 m (i.e., the visibility graph is a unit disc graph).

Nodes are deployed in the area either randomly and uniformly, or according to the distribution reported in Fig. 2, named in the following the Hill distribution. In the Hill distribution, the random variables x and y defining the position of a sensor node are defined as follows: $x = L\sqrt{u}$ and $y = L\sqrt{v}$ with u, v uniformly distributed in $[0..1]$. The introduction of the Hill distribution allows to capture more realistic uneven deployments. Consider for example the case in which sensor nodes are spread over an area by an airplane flying over it. Even if the intended distribution is uniform, wind conditions and terrain features are likely to perturbate such distribution, concentrating the nodes more in certain areas over others. For example sensor nodes might roll down from a steep hill (this motivated the name of the distribution). The introduction of the Hill distribution thus allows us to evaluate which is the effect of perturbating the uniform deployment on the different schemes performance and reliability.

Once nodes have been distributed in the square area, to simulate the broadcasting process, a source node is randomly selected among the ones belonging to the visibility graph giant component, and the broadcast dissemination process is performed.

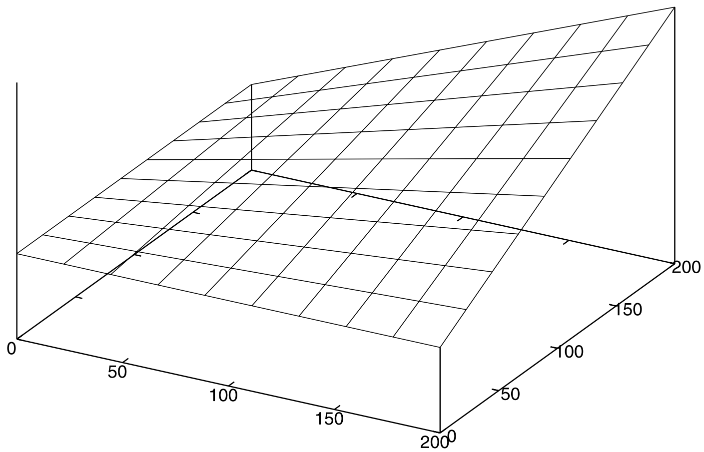
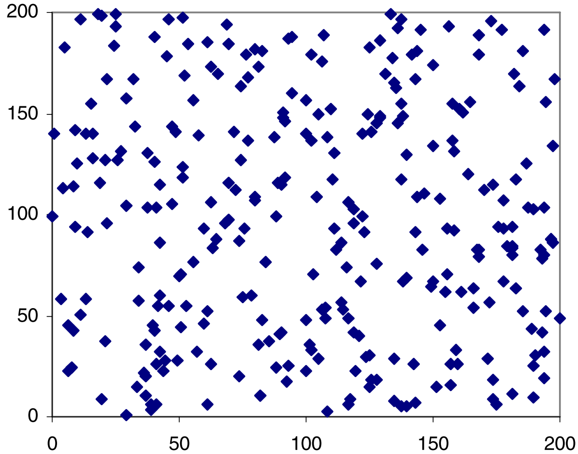


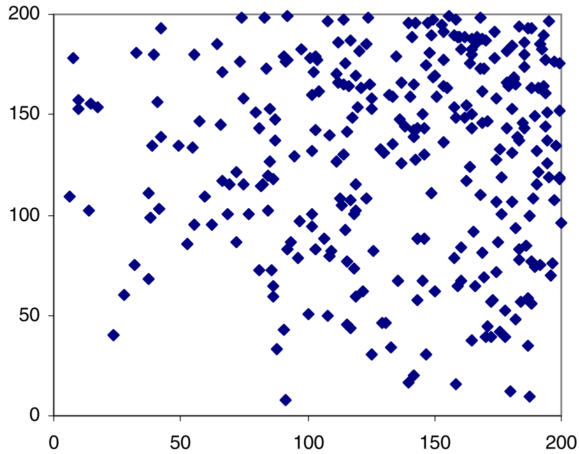
Fig. 2 Hill distribution

Fig. 3 Uniform



The metrics we consider are the following averages: the number of nodes involved in message transmission, the number of links over which the message is transmitted to reach all the intended destinations, and the percentage of nodes successfully reached by the broadcast process. The first metric well evaluates the network load per broadcast dissemination in the ideal case in which messages can always be successfully transmitted via local broadcasts. The second refers to the network load in case unicast packets are used for message transmission. These two metrics are also used for estimating the energy consumption. Indeed, in sensor nodes prototypes, the energy consumption when transmitting or receiving is basically the same (see for example the data reported in [26]) due to the short transmission radius which makes the cost of the circuitry prevalent over the emission power. The cost when in idle mode is also as high as the cost when receiving a packet. In our evaluation we make the assump-

Fig. 4 Hill



tion that nodes adopt an ideal awake-asleep schedule allowing nodes to go to sleep whenever they are not involved in packet transmission or reception. Under this assumption, the energy consumption per broadcast message can thus be approximated by summing up, for each transmitting node, the energy consumption for receiving the packet (a constant times the number of intended destinations to which the message is re-broadcasted) plus a constant accounting for the cost of re-broadcasting the message. The number of (unicast) transmissions accounts for the former, the number of transmissions in case of adoption of local broadcasts for the latter, so that the two curves can also approximate the energy consumption trend.[4] Finally, the number of sensor nodes reached by a broadcast message allows to assess the reliability of the proposed schemes.

All the results have been obtained by averaging over 100 runs on different topologies.

5.1 Irrigator Schemes: Topological Properties

In this section we report the number of connected components, the relative size of the giant component and the number of links in G^n_{rc} when applying one of the two Irrigator methods vs. the same metrics in the visibility graph. Results for the number of connected components, when n varies from 100 to 300, are reported in Figs. 7 and 8 for the uniformly and hill distributed nodes deployment scenarios. Changing n from 100 up to 300 allowed us to test our protocol on increasingly dense networks, from (moderately) sparse networks to highly dense ones.

[4]The decision of adopting an ideal awake-asleep schedule is motivated by the need to derive general results, independent of the specific protocol adopted. The price to pay is that the energy-saving results upper bound what would occur in reality. In case non-ideal awake-asleep schedules are adopted nodes spend a fraction of time in idle mode, consuming energy while neither transmitting nor receiving packets. In such practical scenarios the overall decrease in energy consumption is thus lower than what one would expect by considering the energy saving associated to the reduced number of transmitted/received packets.

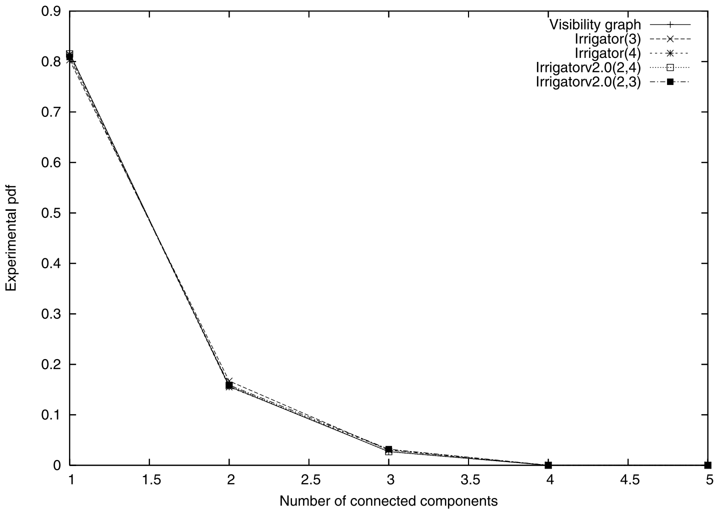


Fig. 5 The figure shows the experimental pdf of the number of connected components. Nodes are $n = 150$, deployed uniformly

As expected, as n increases the graph tends to become globally connected. The striking feature of the figures however is that for the visibility graph G_r and for G^n_{rc}, $c \geq 4$ in the Irrigator protocol, and $c = 2$, $c^* = 4$ in the Irrigator v2.0 protocol, the plots are basically identical, independently of the nodes density. When $c = 3$ in the Irrigator protocol and $c^* = 3$ in the Irrigator v2.0 the plot for G^n_{rc} is slightly worse, meaning that more nodes are needed to have global connectivity, but the trend is the same. The case $c = 2$ in the Irrigator protocol instead shows significantly worst performance. This outcome is confirmed by a wide range of experiments, for varying values of r and n, ranging from very sparse scenarios ($r = 30$), to very dense scenarios ($r = 333$, number of nodes in the hundreds). Figures 5 and 6 also show the experimental probability density function of the number of connected components at $n = 150$ (the critical nodes density), for both uniform and hill nodes deployment. This metric captures, for one of the proposed algorithms, the percentage of visibility graphs (runs) in which the algorithm generates an overlay made of x connected components. As expected, the Hill deployment shows a higher percentage of visibility graphs which result in disconnected overlays, and in overlays with a higher number of connected components. The striking feature however is that again the behaviour of the proposed algorithms matches that of the visibility graph.

Experiments have also been performed to compare the relative size of the largest connected component of G^n_{rc} as the number of nodes grows (see Figs. 7 and 8). The size is relative to the total number of nodes. The Irrigator protocol $c = 2, 3, 4$, the Irrigator v2.0 protocol, with $c = 2$ and $c^* = 3, 4$, and G_r for $r = 30$, have been compared with respect to this metric. (The protocols parameters values have been tuned by means of extensive simulations.) Once again the striking feature we observed is that the plots for G^n_{rc}, $c = 4$ in case of the Irrigator protocol ($c = 2$, $c^* = 4$ in case of the Irrigator v2.0 protocol) and for G_r coincide. These empirical facts have im-

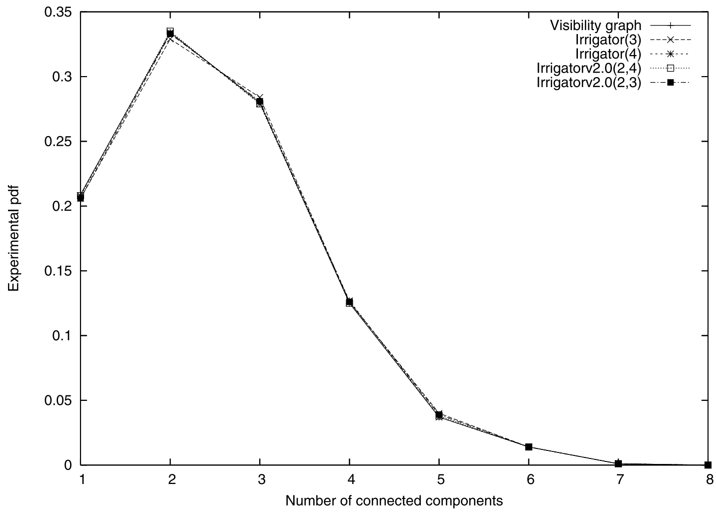


Fig. 6 The figure shows the experimental pdf of the number of connected components. Nodes are $n = 150$, deployed according to the Hill distribution

portant practical implications. In essence they show that, as far as global connectivity is concerned, it does not pay off to set up all possible links (as in a bare Flooding protocol). Rather, it suffices to limit the number of links to a very small constant, as the connectivity properties will be maintained.

We also notice that, for a given n, a WSN made of nodes uniformly deployed tends to have a larger giant component and a more reduced number of connected components over the case in which WSNs nodes are Hill distributed. This accounts for the uneven density of the nodes in the latter scenario. If nodes are Hill distributed, even if n is small there are areas in the WSN in which nodes are concentrated, thus likely belonging to the same connected components. On the other hand, even for high n values there are areas in the WSN in which node deployment is very sparse, resulting in multiple connected components. This also motivates the results plotted in Figs. 7 and 8 which surprisingly show a better capability of the different schemes to generate virtual topologies whose connectivity closely follows that of G_r when nodes are Hill distributed. When WSN nodes are uniformly distributed there is a range of nodal densities, falling in the $100 \le n \le 200$ interval, which are particularly critical for our solutions. Indeed, when the WSN is very sparse, our schemes will tend to select basically all the links; when the WSN is very dense then some links can be removed from the network maintaining the global connectivity. There is however an intermediate interval of nodal densities (of high practical interest, as networks are likely to be deployed with such densities) for which the links to remove have to be carefully selected not to impact the global connectivity of the resulting virtual topology. It is indeed in this range that the effectiveness of our simple and local schemes can be fully appreciated. The Irrigator schemes do not appear to be affected by the Hill uneven deployment and actually benefit from the fact such distribution results in less critical nodal densities (mixing sparsified and dense areas) when varying the number

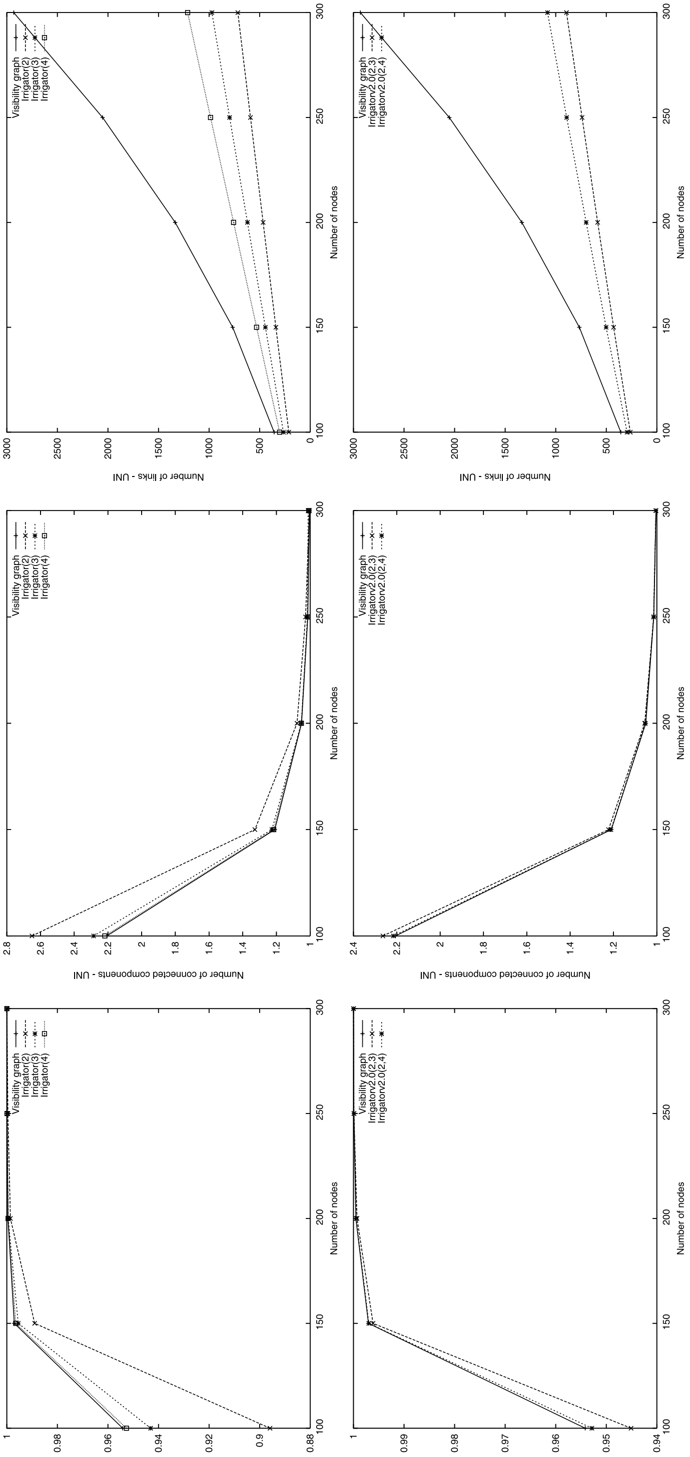


Fig. 7 Uniform distribution. In the picture we report the relative size of the giant component (*left* side), the number of connected components (*center*) and the number of links (*right* side) in G^n_{rc} when applying one of the two Irrigator methods vs. the same metrics in the visibility graph. The number of nodes n varies between 100 and 300 nodes

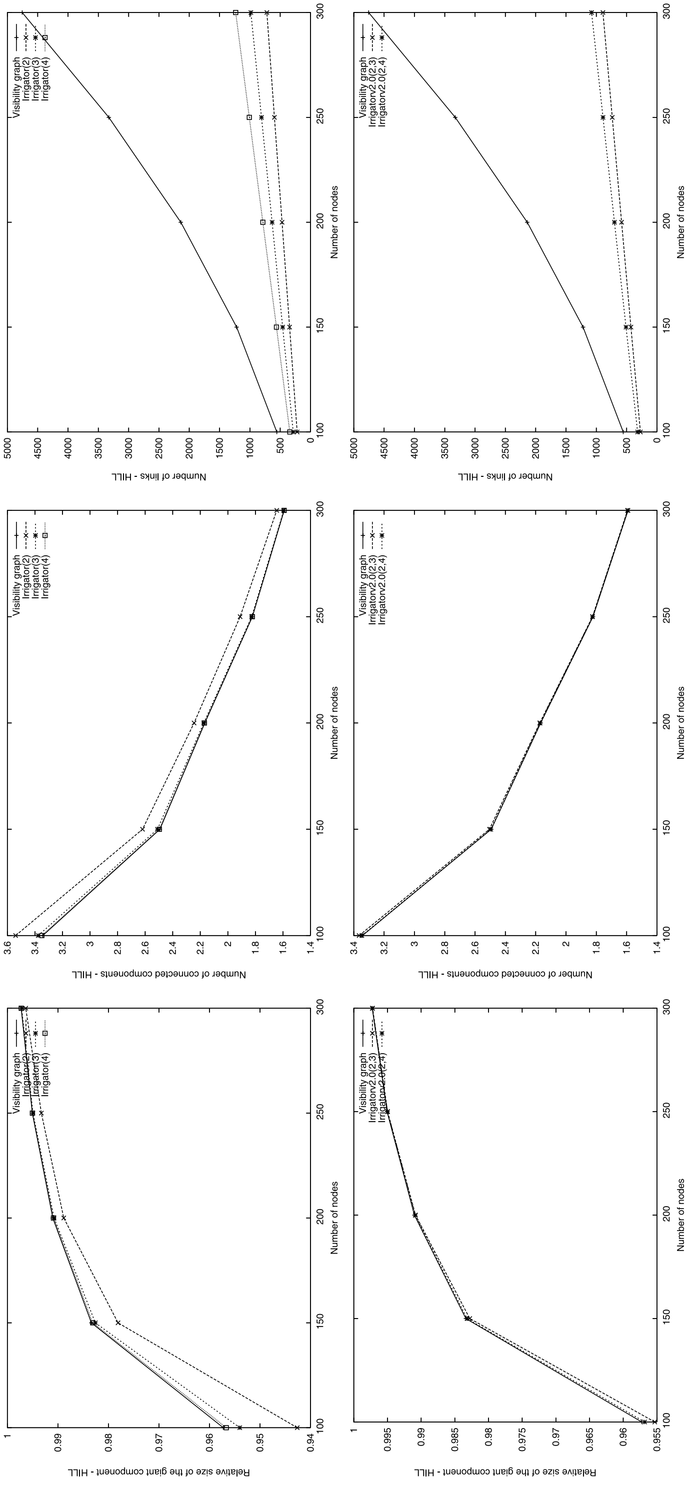


Fig. 8 Hill distribution. In the picture we report the relative size of the giant component (*left* side), the number of connected components (*center*) and the number of links (*right* side) in G^n_{rc} when applying one of the two Irrigator methods vs. the same metrics in the visibility graph. The number of nodes n varies between 100 and 300 nodes

of nodes. This motivates the better results obtained in case of Hill deployments. In Figs. 7 and 8 results on the number of links in the virtual topologies generated by the Irrigator schemes are displayed. The curves reported in the figures show that the proposed solutions are effective in significantly decreasing the number of links in G^n_{rc}, even for moderately sparse network topologies. As the c and c^* values increase the saving slightly decreases but remains always extremely significant for c values of practical interest (i.e., small c values, high enough to be able to guarantee that the global connectivity properties are maintained). In case of the Irrigator protocol, $c = 4$, uniform distribution (Hill distribution), for example, the number of links in G^n_{rc} is reduced of one third (more than halved) at $n = 150$ and is equal to 40% (one fourth) of the links in G_r at $n = 300$. Adopting a Hill deployment results in more remarkable reductions. This is due to the uneven nodes deployments typical of this distribution. As the reduction in the number of links grows fast with n, the reduction obtained in the dense areas of the Hill deployment leads to a considerable reduction in the overall number of links wrt the uniform case. The decrease in the number of links of the virtual topology G^n_{rc} over G_r is even more evident when the virtual topology is obtained with Irrigator v2.0. This scheme leads to a reduction in the number of links up to 12% over the basic Irrigator protocol.

As the traversed link metric can provide an idea of the energy consumption associated to flooding over these topologies, this immediately shows that the adoption of these schemes will result in considerable energy saving over plain Flooding. For the same reason a longer network lifetime can be obtained by adopting the Irrigator v2.0 variant.

5.2 Comparative Performance Evaluation

In this section we summarize the results of a comparative performance evaluation to assess the advantages and limits of the proposed approaches, and to compare them with the gossip schemes previously introduced. In the figures we will denote Gossip or VGossip the vertex gossip protocol and EGossip the edge gossip protocol. We will first compare all the proposed protocols with vertex gossip (Fig. 9 to Fig. 11) and then verify whether the same trends hold also with edge gossip (Fig. 12 to Fig. 14). In Fig. 10 the number of links over which a broadcast message is transmitted when varying n between 150 and 300 is evaluated. This provides insights on the network load in case the message transmission is implemented via unicast, and gives an idea of the energy consumption associated to the different schemes.[5] As the number of nodes (and thus the links in the visibility graph, and the nodes density) increases, the improvements of the proposed Irrigator and Fireworks solutions over the vertex gossip protocol also increase. This is motivated by the fact that, when n increases,

[5] As was previously explained, under an ideal awake-asleep schedule, the energy consumption is given by the sum of the number of transmissions of the same message and the number of times the message is received. Only the latter is accounted for by this metric but it can be seen by combining these figures with the figures on the number of times each message is transmitted that the trends of the different protocols in terms of energy consumption are basically the same as those displayed for the number of links over which a broadcast message is transmitted. Due to space limits we haven't displayed the energy consumption figures.

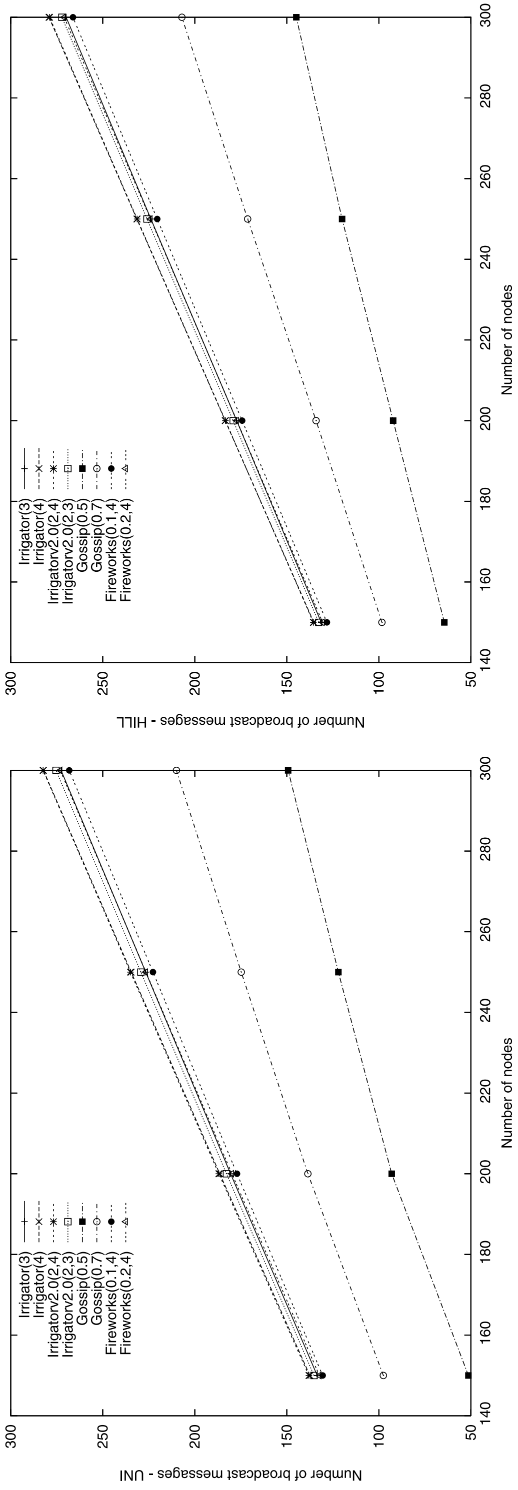


Fig. 9 Number of broadcast messages. The figure shows the average number of times a broadcast message is (re-)transmitted during the broadcast process. The number of nodes n varies between 150 and 300 nodes. Nodes are either uniformly deployed (UNI) or Hill distributed (HILL)

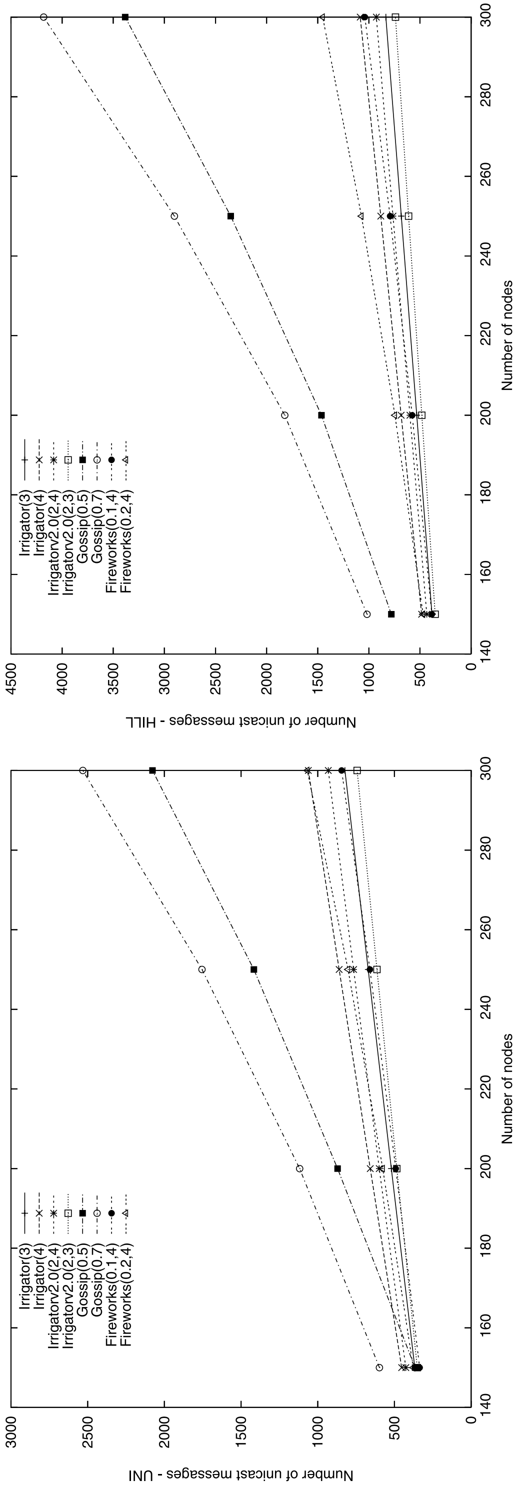


Fig. 10 Number of unicast messages. The figure shows the average number of links over which a broadcast message is transmitted. The number of nodes n varies between 150 and 300 nodes. Nodes are either uniformly deployed (UNI) or Hill distributed (HILL)

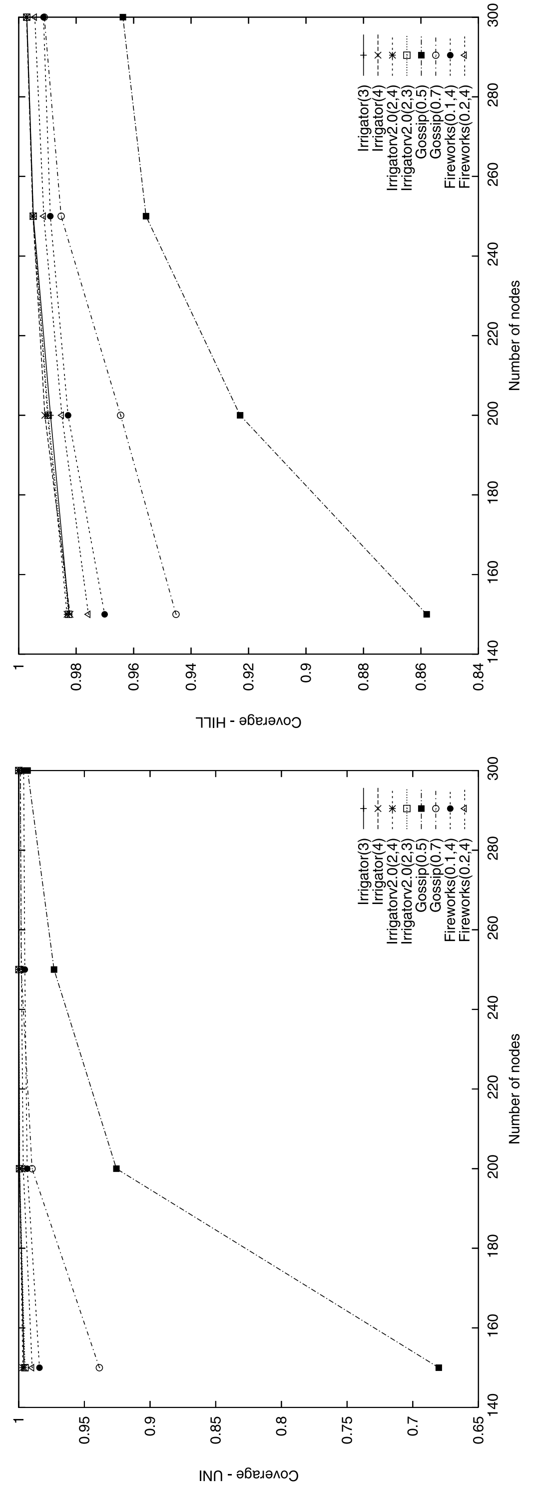


Fig. 11 Coverage. The figure shows the fraction of nodes reached by the broadcasting process. The number of nodes n varies between 150 and 300 nodes. Nodes are either uniformly deployed (UNI) or Hill distributed (HILL)

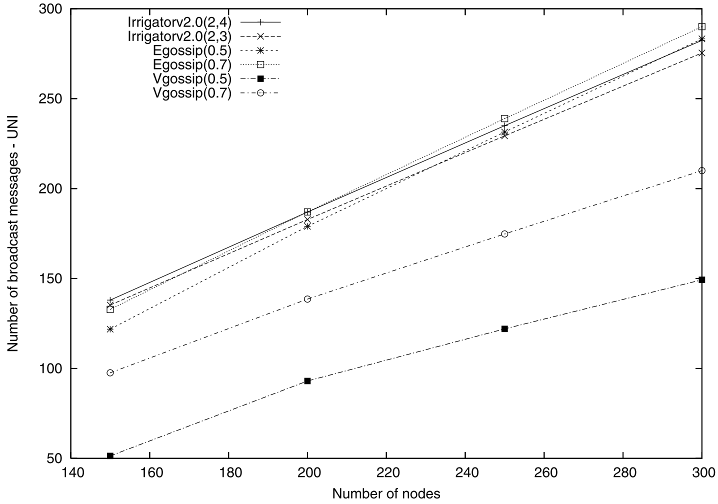


Fig. 12 Number of broadcast messages: Irrigator v2.0 vs. edge and vertex gossip, uniform deployment

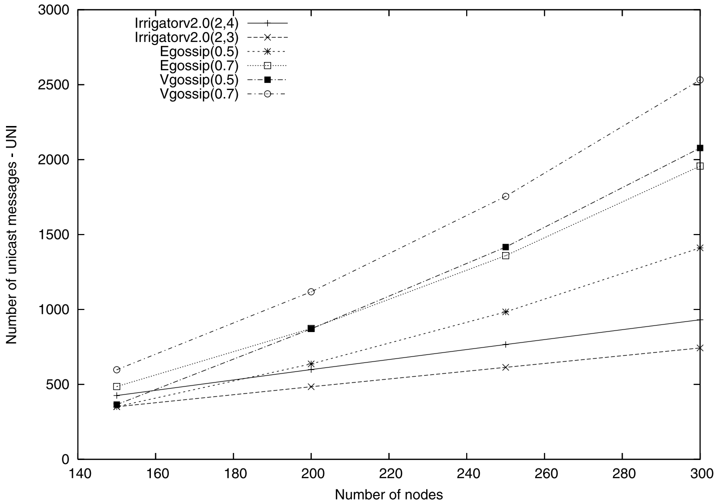


Fig. 13 Number of unicast messages: Irrigator v2.0 vs. edge and vertex gossip, uniform deployment

the number of links over which a message is transmitted by each node increases (being directly related to the node degree), and the saving in the number of links becomes more and more evident in case of solutions which selectively transmit to a restricted subset of the one-hop neighbors. When nodes are Hill distributed this also leads to more evident improvements due to the fact that the uneven deployment leads to remarkable savings in the highly dense areas.

As the c parameter of the Irrigator protocol, the p and c parameters of the Fireworks scheme, and the c and c^* parameters of the Irrigator v2.0 protocol increase,

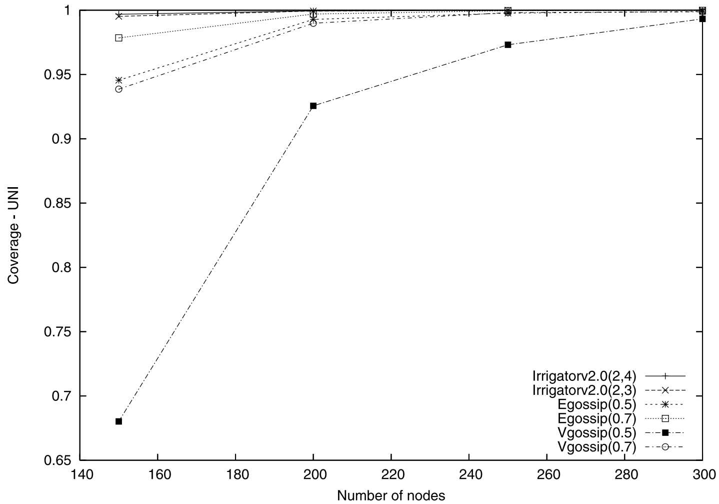


Fig. 14 Coverage: Irrigator v2.0 vs. edge and vertex gossip, uniform deployment

the selection of bigger subsets of the one-hop neighborhood to which to rebroadcast leads to more energy consumption and higher network load. In all the cases, Irrigator v2.0 allows to achieve considerable improvements over the basic Irrigator protocol. Fireworks tends to experience a faster increase in the number of links over which the broadcast message is transmitted over the Irrigator and Irrigator v2.0 protocols. This is easily explained due to the fact that Fireworks transmits to all neighbors with probability p, and the one hop neighborhood average size fast increases with n. This explains the fact that Fireworks experiences similar or slightly better performance than the other protocols at $n = 150$ and then degrades, achieving up to a 30% increase over the Irrigator protocol at $n = 300$ when nodes are Hill distributed. When the densities are lower (e.g., being the nodes uniformly distributed) the Fireworks protocol, for $p = .2$ and $c = 4$ always outperforms the Irrigator protocol with $c = 4$.

Figure 9 shows the number of times a broadcast message is (re-)transmitted in the process of being disseminated to the nodes. With respect to this metric clearly the vertex gossip protocol, in which all nodes transmits only with probability p, has better performance. The Fireworks protocol leads to an increased number of retransmissions, similarly to what obtained by running the Irrigator schemes, as in the Fireworks protocol nodes re-transmit the message, though to a subset of the one hop neighborhood. The price to pay for adopting the vertex gossip protocol is increased energy consumption (from two to three times as much as required by the other protocols) and lower reliability. The latter is clearly shown in Fig. 11. While the Irrigator and Irrigator v2.0 offer a reliability compared to the basic Flooding, and Fireworks achieve a smaller but in any case excellent reliability (with a decrease in terms of percentage of nodes successfully reached never higher than 2%), the vertex gossip protocol experiences worse performance. A very high number of links have to be traversed to be able to reach a high percentage of nodes. Whenever p is decreased from 0.7 down to 0.5 only 65% (85%) can be reached in a uniformly (Hill) distributed WSN at $n = 150$.

Results on the Irrigator and Firework protocols suggested that in a WSN scenario the edge gossip might be a better solution over vertex gossip, as it trades-off a higher number of nodes involved in re-broadcasting the message with lower energy-consumption and higher reliability. We have therefore investigated whether the proposed approaches result in significative advantages also wrt the edge gossip protocol.

Figures 12 to 14 compare the performance of the edge gossip, vertex gossip and Irrigator v 2.0 protocols, under uniform deployment. We have chosen to plot these results in separate figures for sakes of readability. The Hill deployment case shows similar trends.

Edge gossip appears much better performing than vertex gossip. The number of unicast packets transmitted decreases up to 33% with respect to the number of unicast packets sent by vertex gossip. At the same time the reliability of the edge gossip is considerably improved. At $n = 150$, for $p = 0.5$ ($p = 0.7$), "only" 5.5% (2.2%) of the nodes are not successfully reached by the broadcast primitive over the 32% (6%) which do not receive the broadcast message in case of vertex gossip. The price to pay is in the increased number of nodes involved in broadcast message re-transmission. With respect to this metric edge gossip performs similarly to the Irrigator and Fireworks protocols.

Despite these improvements the proposed Irrigator and Fireworks protocols still lead to significantly improved performance with respect to edge gossip. The number of traversed links when edge gossip is adopted is up to 51% (110%) higher (at $n = 300$) when $p = 0.5$ ($p = 0.7$) than the number of links traversed by the broadcast process when the Irrigator v2.0 protocol, $c = 2$, $c^* = 4$, is used. Also basically all the nodes of the networks are reached in case the latter protocol is adopted. The Irrigator and Fireworks protocols thus represent the best trade-off between low overhead, low energy consumption, and high reliability.

6 Conclusions

In this paper we have introduced localized techniques for broadcasting in multi-hop ad hoc sensor networks. Our aim has been to design solutions which only require local (one-hop neighborhood) knowledge, have low complexity, low overhead, and result in low energy consumption, low network load and high reliability.

Three different schemes have been presented: the Irrigator protocol, the Irrigator v2.0 scheme and the Fireworks protocol. The first two schemes are based on the idea to flood over a sparse virtual topology computed by means of inexpensive and fully decentralized protocols. The Fireworks protocol instead belongs to the class of on-line probabilistic flooding. The three approaches have been evaluated by means of thorough simulations, and compared to the gossip protocol previously presented. Simulation results have shown that the presented approaches allow to significantly decrease the energy consumption and network load (the latter in case of unicast transmissions) and to increase the reliability of the broadcasting primitive over the gossip protocol, resulting in promising solutions for the energy-constrained WSNs.

References

1. Asmussen, S., Hering, H.: Branching Processes. Birkhäuser, Boston (1983)
2. Badishi, G., Keidar, I., Sasson, A.: Exposing and eliminating vulnerabilities to denial of service attacks in secure gossip-based multicast. IEEE Trans. Dependable Secur. Comput. **3**(1), 45 (Jan. 2006)
3. Badishi, G., Keidar, I., Melamed, R.: Towards survivability of application-level multicast. In: Proceedings of S.O.S. Survivability: Obstacles and Solutions 2nd Bertinoro Workshop on Future Directions in Distributed Computing, http://www.cs.utexas.edu/users/lorenzo/sos
4. Basagni, S., Carosi, A., Petrioli, C.: Sensor-DMAC: dynamic topology control for wireless sensor network. In: Proc. of IEEE VTC 2004 Fall, Los Angles, California, September 26–29, 2004
5. Basagni, S., Chlamtac, I., Syrotiuk, V.R., Woodward, B.A.: A distance routing effect algorithm for mobility (DREAM). In: ACM/IEEE Intl. Conference on Mobile Computing and Networking (MobiCom), 1998
6. Birman, K.P., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M., Minsky, Y.: Bimodal multicast. ACM Trans. Comput. Syst. **17**(2), 1999
7. Eugster, P.T., Guerraoui, R., Handurukande, S.B., Kermarrec, A.-M.: Lightweight probabilistic broadcast. ACM Trans. Comput. Syst. **21**(4), Nov. 2003
8. Eugster, P.T., Guerraoui, R., Kermarrec, A.-M., Massoulié, L.: Epidemic information dissemination in distributed systems. Computer **37**(5), 60–67 (May 2004)
9. Ferraguto, F., Mambrini, G., Panconesi, A., Petrioli, C.: A new approach to device discovery and scatternet formation in bluetooth networks. In: Proc. of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04), Santa Fe, New Mexico, April 26–30, 2004, pp. 221
10. Gupta, I., Birman, K.P., van Renesse, R.: Fighting fire with fire: using randomized gossip to combat stochastic scalability limits. Qual. Reliab. Eng. Int. **18** (2002)
11. Haas, Z., Halpern, J., Li, L.: Gossip-based ad hoc routing. In: Proc. of IEEE INfocom 2002, New York, June 23–27, 2002
12. Häggström, O., Meester, R.: Nearest neighbor and hard sphere models in continuum percolation. Random Struct. Algorithms **9**(3), 295–315 (1996)
13. Harris, T.E.: The Theory of Branching Processes. Springer, Berlin (1963)
14. Hass, Z.J., Halpern, J.Y., Li, L.: Gossip-based ad hoc routing. In: IEEE Infocom, June 2002
15. Johnson, D., Maltz, D., Broch, J.: DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. In: Perkins, C. (ed.) Ad Hoc Networking. Addison-Wesley, Reading (2001)
16. Kermarrec, A.-M., Massoulié, L., Ganesh, A.J.: Probabilistic reliable dissemination in large-scale systems. IEEE Trans. Parallel Distrib. Syst. **14**(3) (March 2003)
17. Ko, Y.B., Vaidya, N.H.: Location-aided routing (LAR) in mobile ad-hoc networks. In: ACM/IEEE Intl. Conference on Mobile Computing and Networking (MobiCom), 1998
18. Li, X., Stojmenovic, I.: Partial Delaunay triangulation and degree limited localized bluetooth scatternet formation. In: Proc. of AD-HOC Networks and Wireless (ADHOC-NOW), Fields Institute, Toronto, CA, September 20–21, 2002, pp. 17–32
19. Lim, H., Kim, C.: Multicast tree construction and flooding in wireless ad hoc networks. In: Proc. of the ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), Boston, MA, August 11, 2000, pp. 61–68
20. Lindvall, T.: Lectures on the Coupling Method. Wiley, New York (1992)
21. Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. In: Proc. of the 5th annual ACM/IEEE International Conference on Mobile Computing and networking (MobiCOM '99), Seattle, Washington, August 15–20, 1999, pp. 151–162
22. Paruchuri, V., Durresi, A., Jain, R.: Optimal flooding protocol for routing in ad-hoc networks. Comput. Netw. (submitted)
23. Peng, W., Lu, X.: On the reduction of broadcast redundancy in mobile ad hoc networks. In: Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2000), Boston, MA, August 11, 2000, pp. 129–130
24. Perkins, C.E., Roper, E.M.: Ad hoc on demand distance vector routing. In: IEEE Workshop on Mobile Systems and Applications, Feb. 1999
25. Qayyum, A., Viennot, L., Laouiti, A.: Multipoint relaying: an efficient technique for flooding in mobile wireless networks. Technical Report 3898, INRIA, Rapport de recherche (2000)
26. Schurgers, C., Tsiatsis, V., Ganeriwal, S., Srivastava, M.B.: Topology management for sensor networks: exploiting latency and density. In: Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02), Lausanne, Switzerland, June 9–11, 2002, pp. 135–145

27. Stojmenovic, I., Wu, J.: Broadcasting and activity scheduling in ad hoc networks. In: Basagni, S., Conti, M., Giordano, S., Stojmenoviceds, I. (eds.) Ad Hoc Networking. IEEE Press, New York (2004)
28. Stojmenovic, I., Seddigh, M., Zunic, J.: Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. IEEE Trans. Parallel Distrib. Syst. **13**(1), 14–25 (2002)
29. Vogels, W., van Renesse, R., Birman, K.: The power of epidemics: robust communication for large-scale distributed systems. ACM SIGCOMM Comput. Commun. Rev. **33**(1) (January 2003)
30. Williams, B., Camp, T.: Comparison of broadcasting techniques for mobile ad hoc networks. In: Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002), Lausanne, Switzerland, June 9–11, 2002, pp. 194–205
31. Wu, J., Li, H.: On calculating connected dominating set for efficient routing in ad hoc wireless networks. In: Proc. of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM 1999), Seattle, Washington, August 20, 1999, pp. 7–14
32. Xu, Y., Heidemann, J., Estrin, D.: Geography-informed energy conservation for ad hoc routing. In: Proc. of the 7th Annual International Conference on Mobile Computing and Networking (Mobicom'01), Rome, Italy, July 16–21, 2001, pp. 70–84
33. Xue, F., Kumar, P.R.: The number of neighbors needed for connectivity of wireless networks. Wirel. Netw. **10**(2), 169–181 (2004)
34. Yi, Y., Gerla, M., Kwon, T.J.: Efficient flooding in ad hoc networks: a comparative performance study. In: Proc. of the IEEE International Conference on Communications (ICC 2003), Anchorage, AK, May 11–15, 2003