Localized techniques for broadcasting in wireless sensor networks

Lorenzo Orecchia,[*] Alessandro Panconesi,[†] Chiara Petrioli,[†] Andrea Vitaletti[‡]

Abstract

We present three localized techniques for broadcasting in large scale ad hoc networks, i.e., for the problem of disseminating a message from a source node to all the nodes in the networks. Aim of the proposed techniques is to define broadcasting mechanisms that are simple, thus generating low overhead, energy efficient, for deployment in resource-constrained networks, and reliable, in that all the nodes receive the intended message with high probability. The three techniques follow two different approaches for data dissemination. The first approach relies on the idea of identifying local rules for the sparsification of the network topologies. The resulting virtual topology is the actual structure through which broadcast is performed. While sparsification techniques have been proposed before, our solution makes no use of location information. The second approach follows the line of on-line algorithms for the implementation of probabilistic flooding. In this case, the proposed algorithm has been studied analytically, which lead to asymptotic proofs of guaranteeing successful broadcast with very high probability. Performance evaluation and comparison have been performed via simulations among the three proposed techniques and a previous solution for ad hoc broadcast. We have evaluated various metrics of interests versus different nodes distributions, which include the uniform and a more realistic "hill distribution" that takes into consideration certain characteristics of sensor nodes deployment in uneven areas. Our results show that the on-line approach and one of the proposed virtual topology-based solutions offer a remarkable compromise between energy saving, network load and reliability.

1 Introduction

Advancements in wireless and sensor technologies have paved the way for the development of tiny and cheap devices equipped with sensors and wireless transceivers. Such devices, named *s*ensor nodes, are able to monitor events (e.g. seismic activity, animals moving in a forest, intruders entering a monitored area), to process the sensed information, and to communicate the sensed data to one or more *s*inks. Sinks are more powerful devices gathering data for sake of later processing or acting as gateways to different networks. Data communication to/from the sinks is performed by sensor nodes self-organizing themselves in an ad hoc network, referred to as a Wireless Sensor Network (WSN) in the following.

Despite much research work has been devoted in the last years to the design of protocols for ad hoc networks, sensor networks require the protocol stack to be fully redesigned. Protocols developed for WSN have to account for the severe energy and memory constraints of the sensor nodes, for the low data rate of the WSN's enabling technologies, need to be scalable to high volumes of devices, to have little complexity, and to reflect the unique features of data communication in WSNs. In particular, differently from the traditional ad hoc communication paradigm, communication in wireless sensor networks tends to be one to all (from the sink to the sensor nodes) or many to one (from some sensor nodes to the sink). The former for example well captures the case in which the sink distributes queries, named 'interest messages', to the sensor nodes. Via interest message dissemination the sink specifies the events to be monitored and how frequently packets have to be relayed back to the sink whenever an event matching the propagated interests is detected. On the other hand, whenever one of such events occurs, a subset of sensor nodes S close to the event location will start generating data packets, resulting in many to one communication (from the nodes in S to the sink). The way one to all communication is usually implemented in such schemes is via basic flooding: each node, upon receiving the message for the first time, relays it to all its neighbors except the one from which the message was received. Each node transmits the message only once. Flooding is an inefficient implementation of one to all communication which may result in significant resource consumption (both in terms of energy consumption and in terms of the (limited) wireless channel capacity), resulting in a real bottleneck for the operation of resource-constrained networks, such as wireless sensor networks. Designing effective, low resource consuming, simple solutions for network-wide one to all and many to one communication is thus one of the basic prob-

---

[*] Department of Computer Science, Princeton University, Princeton, NJ E-mail:orecchia@princeton.edu

[†] Dipartimento di Informatica, Università di Roma "La Sapienza", Italia. E-mail: {ale,petrioli}@di.uniroma1.it

[‡] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Italia. E-mail: vitale@dis.uniroma1.it

lems to be addressed in these networks.

In this paper we tackle the problem of broadcasting in multi-hop wireless sensor networks (WSN), defined as the process through which one node sends a message to all other nodes in the network via multi-hop routes.

Which is the minimum level of information required by a broadcasting protocol to operate? Despite the fact some solutions claim zero knowledge, as each node simply tosses a coin every time a new broadcast message is received to decide whether to retransmit it or not, we claim this might not be realistic in many practical scenarios in WSN. The reason is that whenever the message transmissions are implemented via unicast packets one-hop neighborhood knowledge (typically implemented via exchange of periodic hello messages) is required. It could be objected that the broadcast nature of the wireless channel can be better exploited by implementing transmissions via local broadcasts. However, this faces the following problem in energy-constrained networks such as WSN: the most powerful knob to reduce energy consumption in these networks is to have the nodes alternating between an awake state, in which the wireless transceiver is operational, and an asleep state in which the nodes cannot transmit or receive but the energy consumption is much lower. Whenever a node transmits a message via local broadcast it will be able to reach only those neighbors which are currently awake. Multiple local broadcasts have thus to be sent for a message to be successfully propagated to the one-hop neighbors. (The adoption of the more reliable unicast packets maybe even more convenient in case a small number of neighbors is simultaneously awake.) Each node is thus required to be aware of its one hop neighbors and of their wake up schedules to be able to synchronize message transmission to times when some uncovered nodes can receive it.

Our goal was thus to design solutions which simply exploit hello messages and the resulting one-hop neighbor knowledge either to locally compute a sparsified virtual topology over which flooding can be performed or to identify to which neighbors to relay the message/whether a message should be relayed at all. In doing so, we traded off the following relevant performance metrics: low complexity, low overhead, low energy consumption, low number of transmissions per broadcast message, and high reliability. In particular, the paper introduces three new techniques for broadcasting following two different approaches for data dissemination. The first approach relies on the idea of identifying local rules for the sparsification of the network topologies. The resulting virtual topology is the actual structure through which broadcast is performed. While sparsification techniques have been proposed before (for example in the context of Bluetooth technology in [1] and [3]), our solution makes no use of location information, required in [3] , and has much lower overhead and complexity over the scheme presented in [1]. Two different schemes of such kind are introduced: the *Irrigator* protocol, and a second version of the Irrigator protocol proceeding in two phases to decrease the number of links in the virtual topology. The second class of broadcasting schemes follows the line of on-line algorithms for the implementation of probabilistic flooding. The idea behind the scheme proposed, dubbed in the following the *Fireworks* protocol, is to transmit with a probability p via local broadcast and with probability (1-p) to a randomly selected small number of neighbors. In this case, the proposed algorithm has been studied analytically, which leads to asymptotic proofs of guaranteeing successful broadcast with very high probability. Performance evaluation and comparison have been performed via simulations among the three proposed techniques and GOSSIP, the solution for ad hoc broadcast presented in [2], to assess the presented schemes effectiveness. In particular, solutions have been compared in terms of (1) number of transmitted messages per broadcast when messages can be transmitted via local broadcast and when they are transmitted in unicast (as it will be explained in the performance evaluation section in a WSN the latter metric also approximates the energy consumption), (2) the percentage of nodes successfully reached by the broadcast message, (3) the capability of the new schemes to maintain the global connectivity properties of the original topology while significantly reducing the number of links in the virtual topology.

The protocols evaluation performed has also addressed the effect on performance of different nodes deployments, ranging from a uniform distribution of nodes in an area to a more realistic ”hill distribution” that takes into consideration certain characteristics of sensor nodes deployment in uneven areas. Our results show that the on-line Fireworks scheme and the sparsified variant of the Irrigator protocol offer a remarkable compromise between energy saving and successful broadcast.

The paper is organized as follows. In the next section we describe the Irrigator and Fireworks protocols, as well as the GOSSIP scheme introduced in [2]. In Section 3 analytical results are derived that assess the reliability of the Fireworks scheme. In Section 4, by means of a thorough comparative performance evaluation, we quantify the trade-offs between the different schemes proposed and the GOSSIP probabilistic broadcasting. Conclusions and future works complete the paper in Section 5.

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 [11] a taxonomy of the different solutions has been reported. The authors divide the different schemes in four groups:

- Simple Flooding;

- Probabilistic-based schemes [5],[4], 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 [5]);
- Neighbor knowledge methods. This category comprises schemes in which two hop neighborhood knowledge is exploited to identify whether rebroadcasting allows to reach new nodes or not [4] [6]. Only in the former case a node retransmits. In the Multipoint Relaying protocol for example [7] each transmitting node computes the set of neighbors that will rebroadcast the message (named Multipoint Relays or MR) and explicitly notifies such nodes via hello message exchange. The MR nodes selection is performed by identifying a minimal set of neighbors allowing to cover the two hop neighborhood. Those one-hop neighbors that are the only possible relay to reach at least one two hop neighbor are in the MR set. The set of two hop neighbors which are not covered by the current MRs is then computed. To cover also such nodes a greedy approach is adopted: at each step the node covering more nodes is inserted in MR and the nodes it covers deleted from the set.

An ns2-based comparative performance evaluation of the major different solutions for network wide broadcasting has been performed in [13].

A taxonomy similar to the one in [11] is reported in a recent work by Stojmenovic and Wu [10]. 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 [11] the authors introduce cluster-based schemes such as [12] [9] 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. The broadcast message is then flooded over the backbone, reaching all the BN nodes and all the gateways. BN nodes take care of rebroadcasting the message to their one hop ordinary nodes. 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. Localized Broadcasting

In this section we present the Irrigator and Fireworks protocols we propose for network-wide broadcasting. In describing the protocols we will distinguish between infrastructure-based solutions, in which a virtual topology is first identified and broadcasting is then implemented via Flooding over such virtual topology, and on-line broadcasting solutions. A network-wide broadcasting protocol previously proposed, the GOSSIP protocol, selected for sake of benchmarking will also be reviewed.

3.1 Infrastructure based solutions

3.1.1 The Irrigator protocol

Our idea stems from the study of the connectivity features of the following simple random graph model. Given are n nodes distributed uniformly at random in the unit square, a transmission radius r and a fixed integer parameter c. 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. among the nodes within each other transmission radius). Each node in G_r randomly selects c among its neighbors. A link (u, v) is included in the virtual topology G_c iff at least one of the two extremes u, v selected the other.

The question we study is, what is the likelihood that G_c is connected? Remarkably, extensive simulations show that G_c will be connected, for $c \geq 3, 4$, whenever G_r was connected. Indeed, the curves of the percentage of nodes in the giant component, and of the number of connected components basically overlap for G_r and G_c, $c \geq 3, 4$, when varying the nodes density. The sparsified topologies G_c have the advantage that the number of links, thus the number of transmitted unicast packets and the energy consumption per node (associated to the node message transmission and to reception of message addressed to the node [1]) is significantly reduced.

The above observations suggest a simple and effective solution for broadcasting, dubbed the Irrigator protocol in the following.

- **Virtual topology 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_c incident to itself.

- **Broadcast message propagation.** Upon reception of a broadcast message a node will retransmit the message to all its neighbors in G_c but the one from which it has received the message, in a flooding-like fashion. Flooding is however limited to the sparsified virtual topology. Message transmission to the neighbors in G_c 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 but the ones connected to it in the virtual topology discard the message upon receiving it.

We note that the only 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. 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.

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.

3.1.2 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_c by the Irrigator protocol. In such protocol when $c = 4, 5$ the nodal degree in G_c is likely to exceed the c value since all the c neighbors selected by a node u, *plus* all the neighbors that selected u are u's neighbors in G_c. 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_c was 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, the node u computes the total number of links Num_{links} incident to it in G_c, 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 communicate the identity of the nodes selected in this second phase of the protocol in the next exchanged hello message. The idea here was to try to avoid nodes with highly variant 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.

3.2 On-line solutions

3.2.1 The GOSSIP protocol

For sake of protocols benchmarking we implemented the GOSSIP protocol introduced in [2]. 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

---

[1] 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.

message it has not generated, it tosses a coin and, with probability p retransmits the broadcast message to its neighbors (apart for the one from which it has received the message). With probability (1-p) it stays silent.

This solution trades off the number of nodes rebroadcasting the message (the lower the p value the lower the number of nodes involved in rebroadcasting the message) with reliability (the lower p the less reliable the protocol is).

3.2.2 The Fireworks protocols

The Fireworks protocol we introduce 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, was that the Fireworks protocol results in higher reliability given the same number of links over which the broadcast packet is transmitted. In the next section we will show that the Fireworks scheme is asymptotically able to successfully reach all neighbors with high probability.

4. Fundamental Results on Localized Broadcasting

4.1 Fireworks

In this section we prove some fundamental results on connectivity of the Fireworks protocol. Consider n nodes uniformly distributed over the unit square $[0, 1]^2$, with transmission radius $r \in (0, 1]$. Recall that two nodes are connected in the visibility graph iff their Euclidean distance is less or equal to r.

4.1.1 Connectivity

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

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

We prove the result for $c = 2$. An analogous proof works for $c > 2$. Fix an integer k such that

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

and partition $[0, 1]^2$ into k^2 subsquares of equal size in the natural way. This ensures that any pair of nodes lying in adjacent subsquares are adjacent in the visibility graph. We first prove a lemma.

Lemma 4.1.

$$\mathbf{Pr}(\textit{Each subsquares contains at least} \frac{n}{2k^2} \textit{ nodes}) = 1-o(1).$$

Proof. Consider a subsquare B and let X denote the number of nodes in B. Then, $\mathbf{E}[X] = \frac{n}{k^2}$ and, by Chernoff bound,

$$\mathbf{Pr}(X < \frac{n}{2k^2}) \leq e^{-\frac{n}{8k^2}}$$

Thus, the probability that some subsquare has less than the required number of nodes is at most $k^2 e^{-\frac{n}{8k^2}}$. □

Hence we may condition on this event for the remaining of the proof.

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 $B_0, B_1, \cdots, B_t$, with $t \leq 2k$, such that

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

For $i \in \{0, \cdots, t\}$, let A_i be the event that the Fireworks 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 $i \neq 0$:

$$\mathbf{Pr}(A_i) \geq \mathbf{Pr}(X_{i-1} \geq 1 | A_{i-1})\, \mathbf{Pr}(A_{i-1})$$

It follows immediately that:

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

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

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

.

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

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

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

$$\mathbf{Pr}(X_i \geq 1|A_i) \geq \mathbf{Pr}(X_i \geq \frac{\log^*(n)}{4k^2}|A_i) \geq 1-e^{-\frac{\log^*(n)}{16k^2}}$$

Moreover

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

because the source always floods.

Finally, we have:

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

which tends to 1 as n goes to infinity, showing *Fireworks* reaches all nodes in B. Then, by union bound on all sub-squares, 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. ☐

4.1.2 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 reaching all nodes and denoting by deg_{out} the out degree of a node in G.

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

Hence, the average overall number of arcs is:

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

which is almost sparse.

4.1.3 Note on the choice of p

The proof above works for all choices of p such that

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

Moreover, if $\limsup_{n\to\infty} f(n) \neq +\infty$ it is easy to show that, as n goes to infinity, at least a constant fraction of the nodes are not reached by the procedure.

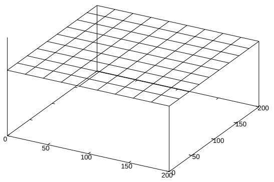

Figure 1. Uniform distribution.

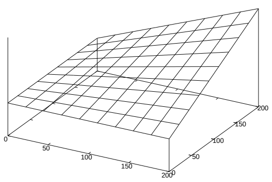

Figure 2. Hill distribution.

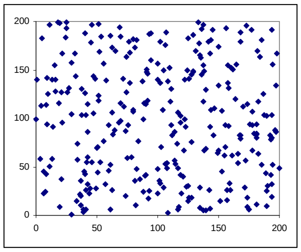

Figure 3. Uniform.

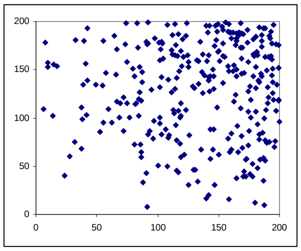

Figure 4. Hill.

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 infrastructure-based Irrigator schemes allow to maintain the global connectivity properties of 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' density, 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 in terms of energy cost, channel capacity demand and reliability of the broadcast process.

In the simulated scenarios, unless otherwise specified, $n \leq 300$ 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 ≤ 30m (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. 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. 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 [8]) due to the short transmission radius which makes the cost of the circuitery prevalent over the emission power. 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 rebroadcasted) plus a constant accounting for the cost of rebroadcasting 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. 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. Infrastructure-based solutions: 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_c when applying one of the proposed infrastructure-base 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 Fig. 6, 9, 12 and Fig. 15 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.

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_c, $c \geq 4$ 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_c 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).

Experiments have also been performed to compare the relative size of the largest connected component of G_c as the number of nodes grows (see Figures 5, 8, 11, 14). 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. Once again the striking feature we observed is that the plots for G_c, $c = 4$ 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 important 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 WSN's 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 Fig. 5 through 16 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 $100 \leq n \leq 200$ 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 of nodes. This motivates the better results obtained in case of hill deployments. In Fig. 7,10,13,16 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_c, even for moderately sparse network topologies. As the c and c^* values increase the saving slightly decrease 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_c 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_c 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 scheme previously introduced. Results are plotted in Fig.17 to Fig.19. In Fig. fig:uni 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. [2] As the number of nodes (and thus the links in the visibility graph, and the nodes density) increase, the improvements of the proposed Irrigator and Fireworks solutions over the GOSSIP protocol also increase. This is motivated by the fact that, when n increases, 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 significative 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, the selection of bigger subsets of the one-hop neighborhood to which to rebroadcast, leads to more energy consumption and 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 $c = 4$.

Fig. 17 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 GOSSIP protocol, in which all nodes transmits only with probability p, has better performance. The Fireworks protocol can lead to comparatively good performance. The price to pay for adopting the 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

---

[2] As was previously explained 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.

clearly shown in Fig.19. 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 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 $n = 150$.

6. Conclusions and Future Works

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 protocols. 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 floodings. 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] F. Ferraguto, G. Mambrini, P. Panconesi, and C. Petrioli. A new approach to device discovery and scatternet formation in bluetooth networks. In *IPDPS*, 2004.

[2] Z. Haas, J. Halpern, and L. Li. Gossip based ad hoc routing. In *IEEE INFOCOM 2002*, 2002.

[3] X. Li and I. Stojmenovic. Partial delaunay triangulation and degree limited localized bluetooth scatternet formation. In *Proc. AD-HOC NetwOrks and Wireless (ADHOC-NOW), Fields Institute*, pages 17–32, Fields Institute, Toronto, CA, September 20–21 2002.

[4] H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, 2000.

[5] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *5th annual ACM/IEEE international conference on Mobile computing and networking(MobiCOM '99)*, 1999.

[6] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *ACM international symposium on Mobile ad hoc networking (MobiHoc 2000)*, 2000.

[7] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *Technical report 3898, INRIA - Rapport de recherce.*, 2000.

[8] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. B. Srivastava. Topology management for sensor networks: Exploiting latency and density. In *Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, 2002.

[9] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. In *IEEE Transactions on Parallel and Distributed Systems.*, 2001.

[10] I. Stojmenovic and J. Wu. Broadcasting and activity scheduling in ad hoc networks. In *Ad Hoc Networking (S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, eds.), IEEE Press, to appear.*, 2004.

[11] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *ACM international symposium on Mobile ad hoc networking (MobiHoc 2002)*, 2002.

[12] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM 99)*, 1999.

[13] Y. Yi, M. Gerla, and T. J. Kwon. Efficient flooding in ad hoc networks: a comparative performance study. In *IEEE ICC*, 2003.

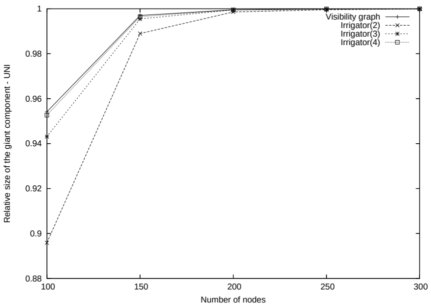

Figure 5.

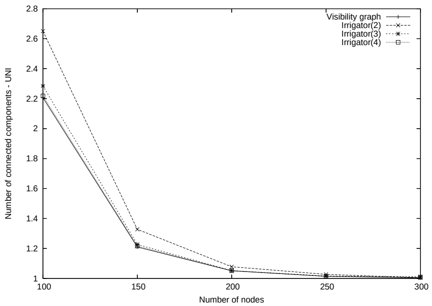

Figure 6.

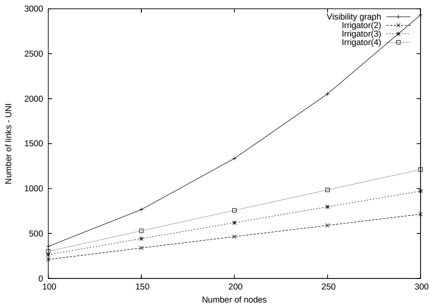

Figure 7.

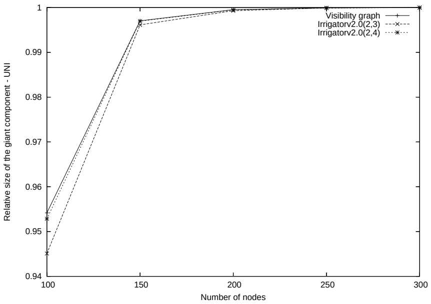

Figure 8.

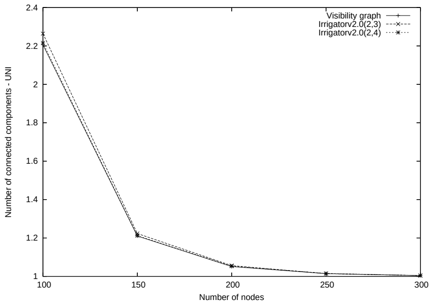

Figure 9.

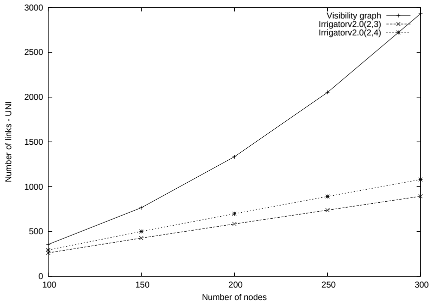

Figure 10.

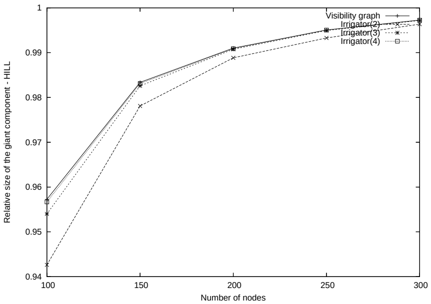

Figure 11.

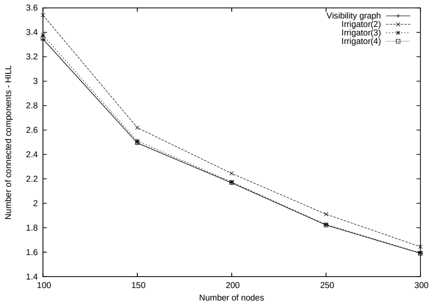

Figure 12.

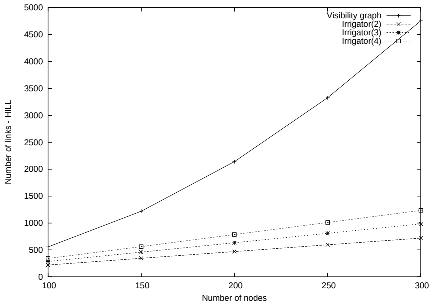

Figure 13.

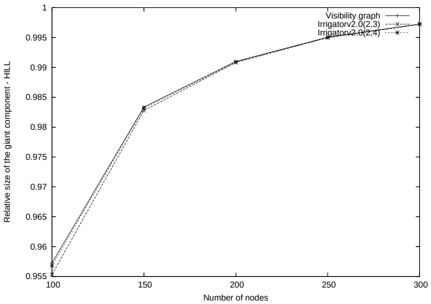

Figure 14.

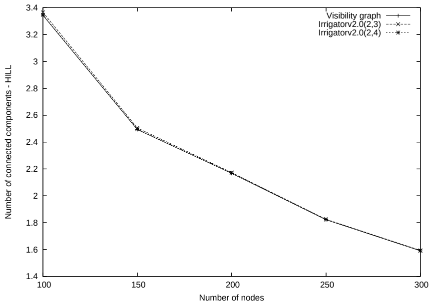

Figure 15.

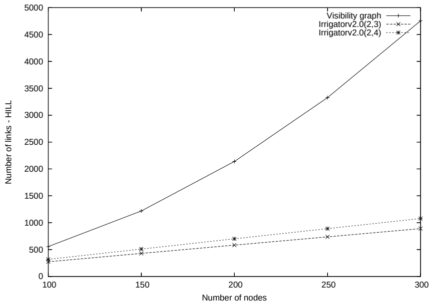

Figure 16.

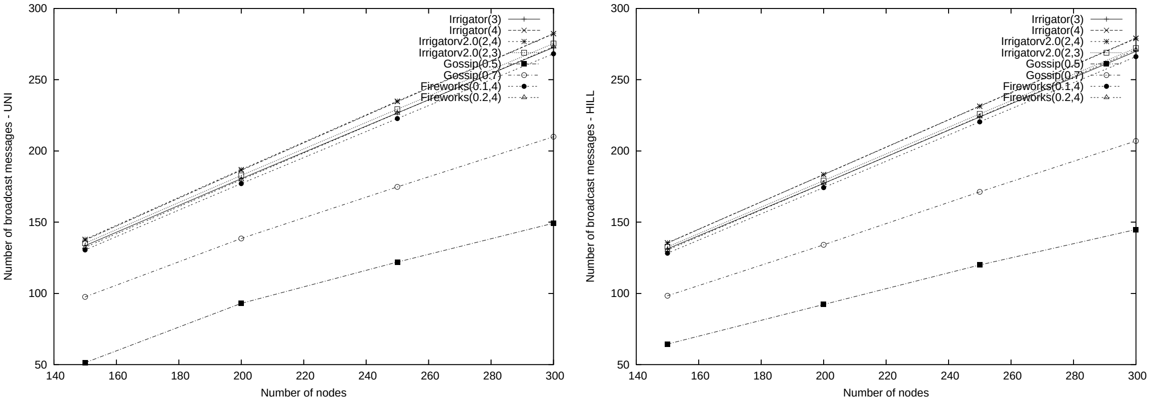

Figure 17. Number of broadcast messages

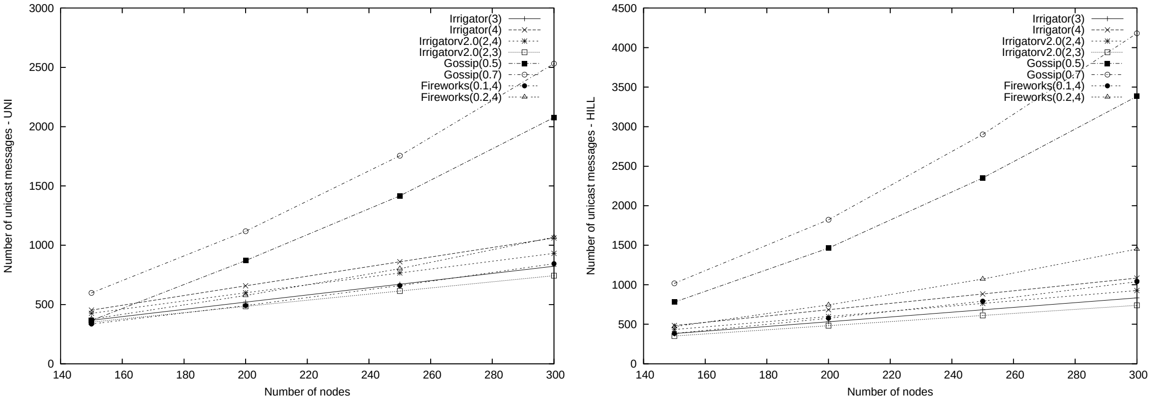

Figure 18. Number of unicast messages

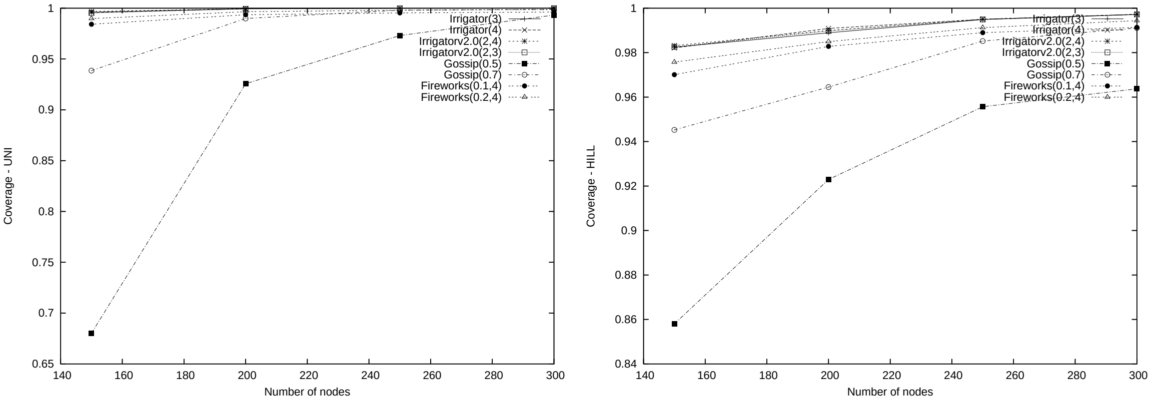

Figure 19. Coverage