Optimal Memory-aware Sensor Network Gossiping
(or How to Break the Broadcast Lower Bound)

Martín Farach-Colton[a], Antonio Fernández Anta[b], Miguel A. Mosteiro[c]

[a]*Department of Computer Science, Rutgers University, Piscataway, NJ, USA & Tokutek Inc.*
[b]*Institute IMDEA Networks, Madrid, Spain*
[c]*Department of Computer Science, Kean University, Piscataway, NJ, USA & GSyC, Universidad Rey Juan Carlos, Madrid, Spain*

Abstract

Gossiping is a well-studied problem in Radio Networks. However, due to the strong resource limitations of sensor nodes, previous solutions are frequently not feasible in Sensor Networks. In this paper, we study the Gossiping problem in the restrictive context of Sensor Networks. We present a distributed algorithm that completes Gossiping with high probability in a Sensor Network of unknown topology and adversarial start-up. This algorithm exploits the geometry of sensor node distributions to achieve an optimal running time of $\Theta(D + \Delta)$, where D is the diameter and Δ the maximum degree of the network. Given that any algorithm for Gossiping also solves the Broadcast problem, this result shows that the classical Broadcast lower bound of Kushilevitz and Mansour does not hold if nodes are allowed to do preprocessing. The proposed algorithm requires that a linear number of messages be stored and transmitted per unit time. We also show an optimal distributed algorithm that solves the problem in linear time for the case where only a constant number of messages can be stored.

Keywords: Radio Networks, Sensor Networks, Gossiping, Distributed Algorithms, Broadcast, Convergecast

---

Email addresses: `farach@cs.rutgers.edu` (Martín Farach-Colton), `antonio.fernandez@imdea.org` (Antonio Fernández Anta), `mmosteir@kean.edu` (Miguel A. Mosteiro)

A preliminary version of this work is included in [1].

1. Introduction

A ***Sensor Network*** is a type of ***Radio Network***, which in turn is a simplified abstraction of a radio-communication network, described informally as follows. Processors, or ***nodes***, communicate through broadcast on a radio channel. The range of transmission defines an (undirected) reachability graph on the nodes. If two nodes transmit simultaneously on the same radio channel, any node adjacent to both receives ***interference noise***, as opposed to the ***background noise***. Such an event is called a ***collision***. ***Collision detection*** is the ability for nodes to distinguish between interference and background noise. Further details can be found in [2].

The question of how to disseminate information in a Radio Network has led to different well-studied problems. The piece of information that a node holds, which must be distributed to other nodes, is called a ***message***. Dissemination problems can be categorized by the number of nodes holding messages to transmit, the number of different messages to be transmitted, and the set of nodes that must receive the messages. Consider the case where all nodes in the network must receive all messages. When each of k nodes initially has a message, the problem is known as k***-selection*** [3]. If $k = 1$, the problem is called ***Broadcast*** [4, 5], and if $k = n$, where there are n nodes in total, it is called ***Gossiping*** [6, 7]. Information dissemination problems are frequently classified as ***conditional wake-up*** or ***spontaneous wake-up***. In conditional wake-up dissemination, each node is idle before receiving any message. In spontaneous wake-up, nodes may compute and communicate at any point, for instance, to agree on a collision-avoidance protocol. Throughout the paper, any such spontaneous activity is called ***preprocessing***.

Gossiping in Radio Networks is a well-studied problem for which important results have been obtained under a myriad of models. Unfortunately, certain assumptions made sometimes for Radio Networks may be too strong for Sensor Networks. For instance, a global clock makes collisions easier to avoid, but the nature of Sensor Networks makes it unreasonable to assume that a global clock is available.

We study the Gossiping problem in Sensor Networks. (A formal definition of the particular Sensor Network under consideration in this work is given in Section 3.) In brief, a Sensor Network is a Radio Network with some additional restrictions. The nodes of a Sensor Network are called ***sensor nodes*** and are assumed to be low cost, since they are distributed in large numbers over potentially remote or hostile environments. Therefore, they are assumed to be subject to strict resource limitations, for example, in the size of their memory or number of messages that can be transmitted. Sensor nodes collect environmental information which they must share. Typically, there are assumed to be distinguished nodes called ***sinks*** [8], which can either store more environmental information or can communicate to the outside world [1]. Since the identity or the location of sinks is frequently assumed to be unknown to the sensor nodes, information is shared with all nodes in order to reach the sinks. Therefore Gossiping algorithms are used as a critical communication primitive in this setting.

The number of messages that can be transmitted in one time slot has an impact on the time efficiency of deterministic Gossiping algorithms [9], even in general Radio Networks. For Sensor Networks, where memory and communication are restricted, we consider two scenarios. First, we consider the less restrictive ***combined-messages model*** [9], in which messages can be combined, and retransmitted. Thus memory and communication length are effectively bounded to be linear in the number of messages. In the more restrictive ***unit-messages model*** [10, 11] or ***separate-messages model*** [9], memory and communication length are bounded by a constant. Throughout the paper we assume that the restrictions on memory and communication length are identical, so a restriction on one will be the same restriction on the other. Notice that the goal is that sink nodes receive all messages over the course of the algorithm, and so having memory limitations on other nodes still allows Gossiping to take place.

1.1. Our Results

In the combined-messages model, we present a randomized distributed algorithm that, given a network of n nodes, with high probability[2], completes Gossiping in time $O(\Delta + D)$ after the last node starts running the algorithm, where D is the diameter and Δ the maximum degree of the network[3]. Given that $\Omega(D)$ and $\Omega(\Delta)$ are lower bounds for this problem, our algorithm is

[1] Although sinks are not a part of the mathematical formalism of Radio Networks, they can be modeled in Radio Networks through a leader election round. Therefore, Sensor Networks are a type of Radio Network, w.l.o.g.

[2] We say that a parameterized event E_p occurs ***with high probability***, or ***w.h.p.*** for short, if for any constant $\kappa > 0$ there exists a valid choice of parameter p such that $Pr\{E_p\} \geq 1 - n^{-\kappa}$.

[3] A preliminary version of this result was presented in [1].

optimal. This result improves over previous bounds in time efficiency and makes no assumptions about global synchronism; only the clock-tick rate is assumed to be the same for all nodes. Rather, it exploits the geometry of the node placement. For uniformly randomly distributed nodes, it is known that $\Delta \in O(\log n)$[4] and $D \in O(\sqrt{n/\log n})$ w.h.p. Hence, our result of $O(D+\Delta)$ implies an upper bound of $O(\sqrt{n/\log n})$, improving over [12]. This result also shows that the classical expectation lower bound for Broadcast of Kushilevitz and Mansour of $\Omega(D\log(n/D))$ [5] does not hold if nodes are allowed to do some preprocessing before receiving a message to transmit, since the gadget they construct in order to prove their lower bound is geometric and thus applies to Sensor Networks. An efficiency gap between the conditional and spontaneous models was shown [13, 14] but only for deterministic algorithms.

A crucial problem to achieve communication in Radio Networks where nodes do not have topology information is to estimate the number of nodes in a given area, or ***density***. Once such a density is known by nodes competing for the channel, a simple randomized algorithm, where each node transmits with probability inverse of the density, maximizes the probability of achieving non-colliding transmissions. A communication primitive that guarantees that, if Δ out of n nodes compete, all of them achieve a non-colliding transmission within $O(\Delta+polylog(n))$ time steps with high probability is included in our algorithms. This primitive estimates the density by probing the channel in a fashion similar to previous work [15], but our technique achieves a better bound on the probability of success, which is needed in the analysis of our overall algorithm.

Finally, for the unit-message model, we show a distributed algorithm that solves Gossiping w.h.p. in $O(n)$ time. Given the memory restriction, this time complexity is asymptotically optimal.

1.2. Roadmap

In the rest of this paper we overview related work in Section 2, we define the models used throughout in Section 3, we establish lower bounds in Section 4, a preprocessing procedure and a communication primitive used in both algorithms are presented in Sections 5.1 and 5.2, and the details of both algorithms are given in Sections 5.3 and 5.4.

[4]Throughout this paper, log means $\log_2$ unless otherwise stated.

2. Related Work

2.1. Sensor Networks

The literature on Sensor Networks is vast, but most of the work in this area is either non-analytical, includes strong assumptions about node resources, or it is focused on other problems. To the best of our knowledge, the only previous work analyzing Gossiping specifically for Sensor Networks is [12]. The network topology is modeled by a random geometric graph, which is a model widely used in the Sensor Network area. The algorithm presented completes Gossiping in $O(\sqrt{n}\log n)$ w.h.p. In a first stage, nodes obtain an ID and define a coloring in order to avoid collisions later in the gossiping phase. This algorithm is claimed to be optimal with respect to the result of Kushilevitz and Mansour [5] for the Broadcast problem. However, the algorithm in [12] includes preprocessing, which is not allowed in the lower bound proof in [5].

2.2. Radio Networks

The prominent literature on Gossiping and related problems in Radio Networks is abundant and includes a myriad of models and different classes of algorithms. Regarding upper bounds, while the Sensor Networks topology can be embedded in most of the topology models assumed in Radio Networks, frequently some of the node capabilities assumed are not feasible in a restricted Sensor Network model. As for lower bounds, while not using restrictions present in Sensor Networks only makes the bound stronger, many times the topology used to prove the lower bound can not be embedded even in a general geometric graph. In what follows, we overview results within the broader scope of Radio Networks that are relevant for this work.

Upper Bounds. Bar-Yehuda, Israeli, and Itai [16] presented a randomized algorithm for Radio Networks with a topology modeled by an undirected graph[5], which completes Gossiping in $O(n\log^2 n)$ on average. Briefly, their technique, used previously in [17] and later re-utilized in [10], is to build an underlying spanning tree to first collect all messages in the root node, and later disseminate all of them to all nodes. In that paper, nodes know the identity of their neighbors, the size of the network n, and an upper bound on the maximum degree. Nodes may transmit and receive only $O(\log n)$

[5]Throughout this paper, the terms undirected graphs, undirected networks and symmetric networks are used indistinctively.

bit messages in synchronous time slots. However, they can store as many messages as needed.

The same bound but with high probability was proved in [7]. Their algorithm relies on global synchronism and works under the combined-messages model. For asymmetric Radio Networks, Chrobak, Gąsieniec, and Rytter [18] showed an upper bound of $O(n \log^3 n \log(n/\epsilon))$ that holds with probability $1 - \epsilon$ and of $O(n \log^4 n)$ in expectation. The main idea of their algorithm is to repeatedly run a limited broadcast that doubles the number of copies of each message in the network in each phase. Thus, combination of messages as well as global synchronism are necessary. Using the same algorithm, but improving the limited broadcast by adding randomization to it, Liu and Prabhakaran [6] reduced that upper bound by a logarithmic factor. More recently, Czumaj and Rytter [19] obtained a bound of $O(n \log^2 n)$ w.h.p. for this algorithm by replacing the limited broadcast by a linear randomized broadcast where the probabilities are chosen with a special distribution. The model in all these results is a directed strongly connected graph where nodes have unique ID's in $\{1, \ldots, n\}$, work synchronously, and messages can be combined. Global synchronism precludes the application of any of these results to a restricted Sensor Network.

In [11], Chlebus, Kowalski and Radzik studied the problem called ***Many-to-Many Communication*** where p out of the n nodes, named ***participants***, start with a message and all participants must receive all messages. In their weakest model, nodes know n, p and the diameter d of the graph induced by the participants. If $p = n$ the problem becomes Gossiping, $d = D$, and the upper bound presented for undirected Radio Networks, becomes $O((D \log n + n) \log n)$ with high probability. The result presented in the present paper improves over that bound.

Deterministic algorithms may also be of interest for comparison. Results for networks with known topology include the following. For the unit-message model, deterministic upper bounds of $O(n^2)$ for asymmetric Radio Networks and $O(n \log^2 n)$ for symmetric Radio Networks were shown in [10] by Gąsieniec and Potapov. Whereas for the same networks but under the combined-messages model, a $O(D + \Delta \log n)$ algorithm was presented by Gąsieniec, Peleg and Xin in [20]. Moving to settings where the topology is unknown and the network is asymmetric, Chrobak, Gąsieniec, and Rytter [21] presented a $O(n^{3/2} \log^2 n)$ protocol and Chlebus, Kowalski and Rokicki showed in [9] an upper bound of $O(n/ \log n)$ in expectation. Both protocols exploit that messages can be combined. Additionally, in the latter, if messages cannot be combined an upper bound of $O(n \log n)$ in expectation is proved. In both results in [9] the expectation is taken over a uniform distribution on all strongly-connected networks. In all these models, nodes start-up simultaneously. Hence, the upper bounds do not apply to harshly restricted Sensor Networks.

Lower Bounds. Chlebus, Gąsieniec, Lingas, and Pangourtzis [7] proved that any deterministic oblivious algorithm requires at least $n^2/2 - O(n)$ time to complete Gossiping in asymmetric Radio Networks. In the same paper, for the class of fair randomized algorithms, i.e., algorithms where all nodes use the same probability of transmission in the same time slot, it was proved that the expected time is $\Omega(n^2)$. Our algorithm is adaptive and is not in the class of fair algorithms so, neither of these lower bounds apply. Later, Gąsieniec and Potapov [10] showed lower bounds of $\Omega(n^2)$ for asymmetric Radio Networks and $\Omega(n \log n)$ for symmetric Radio Networks, both under the unit-message model, even if the topology is known. However, the topology of the construction used for the latter can not be embedded in geometric graphs, and therefore the lower bound does not apply to our model of Sensor Networks.

More recently, Chlebus, Kowalski and Radzik [11] showed that there exist undirected Radio Networks such that it takes $\Omega(p+d\log(n/d))$ expected time to solve the many-to-many communication problem where p is the number of participants and d the maximum hop distance among any pair of them, even if messages can be combined. The problem becomes Gossiping if $p = n$. The lower bound showed becomes linear when the number of participants is linear. Our general lower bound implies this one for such networks.

Results for the Broadcast problem can be used as lower bounds for Gossiping as well, because the former can be solved using an algorithm for the latter. For the conditional wake-up model, a general lower bound for randomized algorithms of $\Omega(D \log(n/D))$ in expectation was obtained by Kushilevitz and Mansour in [5]. Under the same assumption, but for deterministic algorithms, Bruschi and Del Pinto [22] proved the existence of networks where $\Omega(D \log n)$ steps are required to solve Broadcast. Also for determinisc algorithms, Dessmark and Pelc [23] proved an existential lower bound for Broadcast of $\Omega(D + \log n)$ for symmetric networks even in the spontaneous wake-up model. Clementi, Monti and Silvestri [24] improved the lower bound to $\Omega(n \log D)$ for asymmetric networks even in the spontaneous wake-up model, as well as for symmetric networks in the conditional wake-up model. Kowalski and Pelc [25] showed an $\Omega(n)$ lower bound for general deterministic algorithms and in expectation for randomized algorithms that make decisions based only on the ID and the step number provided by a global clock. It was shown in [13] that in geometric networks Broadcast time depends on the minimum distance between any pair of nodes. None of these bounds apply to our model because either they do not hold for adaptive algorithms, or they correspond to deterministic algorithms, or the topology of the construction used for the lower bound can not be embedded in geometric graphs.

Convergecast [26] is a subproblem of Gossiping where all nodes hold a message but there is only one destination node. Lower bounds for Convergecast also apply to Gossiping. In [26], Kesselman and Kowalski showed that it takes at least $\Omega(\log n)$ time steps to solve convergecast in an arbitrary network even if nodes can detect collisions and measure the Euclidean distance to the closest neighbor. Given that the Gossiping upper bound we show is $O(D+\Delta)$, and that $\max\{D,\Delta\} \in \omega(\log n)$ in our model of Sensor Network, this bound does not contradict our result.

3. The Model

Although Sensor Networks have been formalized in many different ways, in this paper we consider the following very restrictive model.

Definition 1. *A* ***Sensor Network*** *is composed of a set of n* ***sensor nodes*** *with radio-communication, processing, and sensing capabilities as detailed in Section 3.1. Sensor nodes are located in a two-dimensional convex area. A communication link between a pair of sensor nodes exists if and only if they are located within communication range, yielding a network topology modeled as detailed in Section 3.2. There exists one distinguished node in the network called* ***sink*** *that may be set to run a different protocol*[6]*.*

Given that the messages are all different in general, the size of a message is assumed to be $\Omega(\log n)$ bits. We assume that a central controller is not available (as customary in Ad-hoc Networks and Sensor Networks). Hence, the Gossiping protocol must be distributed. I.e., it must include n

[6]The existence of a sink node is only used for the unit-message model. Given that we only use the sink node capability of running a different algorithm, should a sink node not be available, nodes could initially elect a leader to serve in place of the sink.

algorithms, one for each node, although some (even all) nodes may run the same algorithm.

In our model nodes may reduce their range of transmission by a constant factor, which has an impact on network connectivity. We assume that the density of nodes is high enough so that connectivity is achieved even using the reduced range of transmissions, which introduces only a constant overhead on the running times.

3.1. Node Capabilities Model

We use a relaxed version of the Weak Sensor Model described in [27] including the following assumptions.

- No position information or distance estimation capabilities are available.
- Time is devided into communication steps, called ***time steps*** or ***time slots*** indistinctively. Computation time-cost is negligible in comparison with the communication time-cost. Hence, a time slot is long enough so that a node is able to deliver a message to another node (under certain conditions defined below).
- An adversary chooses the time slot at which each node starts running the protocol. We say that nodes ***wake-up*** or ***start-up*** adversarially. No global synchronizing mechanism is available.
- Radio communication: The communication is carried out in a physical medium called a ***radio channel***. The delivery of messages to other nodes is attempted through ***broadcast*** in a radio channel. That is, a node ***transmits*** its message using its radio capabilities, and only nodes located within its range of transmission may receive it, depending on certain conditions as defined below. A sensor node may vary its power of transmission, effectively adjusting its range of transmission, or ***radius*** of transmission. The radius r is the same for all nodes. A node may reduce its range of transmission from a maximum $r \ll 1$ to any value $r_{min} < r$ as long as $r_{min} \in \Theta(r)$[7].

[7]For upper bounds we only use three radii.

- Low-information channel contention: All nodes have access to only one radio channel. Hence, conflicts while attempting to access that channel simultaneously may occur. A node ***receives*** a message in a given time slot if and only if exactly one node located within distance r transmits in that time slot. If more than one message is transmitted, a ***collision*** occurs, the messages are garbled, and a node in this area receives only ***interference noise***. If no message is transmitted in its area, the node still receives some noise from the channel called ***background noise***. Nodes cannot distinguish between interference and background noise and they cannot receive and transmit in the same time slot. We say that the communication is carried out ***without collision detection***.

The Weak Sensor Model also limits memory size, life cycle, and reliability. Regarding memory restrictions, we consider two cases: the ***unit-message model*** where node-memory and number of messages that can be transmitted in one time slot is limited to a constant number of messages (as in the Weak Sensor Model); and the ***combined-messages model*** where such limitation is only linear on the number of messages. Regarding power supply restrictions or node reliability, given that in order to solve the Gossiping problem all nodes have to receive the message, arbitrary failures may render the problem unsolvable. We assume in this paper that no node turns off before completion of the algorithm.

Our model is a relaxation (in the sense that some assumptions are less restrictive) of the Weak Sensor Model [27], which includes the same set of restrictions but where failures are arbitrary and memory is limited as in the unit-message model. Bar-Yehuda, Goldreich, and Itai [4] used a formal model of Radio Network, which additionally includes topology assumptions, specifies many of the node restrictions here, including limits on contention resolution, but they make no mention of computational limits such as small memory. Later on, more restrictions have been added to the model in various papers, such as in the unstructured Radio Network model [28]. Notice that the unstructured Radio Network model does not include all the restrictions of our model. For instance, it does not include lack of position information. But, more importantly, the unstructured Radio Network model does not include limits on memory size, which is a fundamental restriction [29].

3.2. Topology Model

We model the topology as a ***Geometric Graph***, where nodes are distributed arbitrarily in a convex area in the Euclidean plane, the size of which is normalized to one. A pair of nodes is connected by an edge if and only if they are at distance at most r, in which case we call them ***neighbors***. This model is a relaxation of the Random Geometric Graph model frequently used for Sensor Networks since we do not restrict the deployment distribution or area shape. The ***diameter*** of the network D is defined as the length of the longest hop-optimal path between any two vertices. The ***maximum degree*** of the network Δ is defined as the maximum number of nodes within distance d, $0 < d \leq r$, of any node.

As customary in Sensor Networks, nodes are assumed to be deployed densely enough to guarantee connectivity[8]. As in previous work [30, 31], nodes may set the transmission power at different levels. By setting the power of transmission, a node is able to effectively set its radius of connectivity. We assume that connectivity is guaranteed even while using the smallest power of transmission. The only knowledge each node has is the number of nodes in the whole network n and its own unique identifier in $[1, n^\kappa]$, where $\kappa \geq 1$ is some arbitrary constant. In addition, the sink knows that it is the sink.

4. Lower Bounds

The following straightforward theorems establish formally lower bounds for Gossiping. Recall that D and Δ are the diameter of the network and the maximum degree respectively for the maximum radius of transmission r.

Theorem 1. *For any Sensor Network of n nodes, diameter D, maximum degree Δ, and maximum radius of transmission r, under the restrictions detailed in Section 3, $\Omega(D + \Delta)$ time slots are needed in order to solve the Gossiping problem, even if nodes use a radius of transmission reduced by a constant factor $0 < \kappa \leq 1$.*

Proof. If nodes use radius r, $\Omega(D)$ is a trivial lower bound, since messages have to traverse the diameter of the network. If nodes reduce the radius of transmission to κr, D is still a lower bound on the diameter since reducing the radius cannot decrease the diameter.

Given that transmissions are performed in a unique shared channel, to prove that $\Omega(\Delta)$ is also a lower bound it is enough to show that, given a

[8]We assume in this work that the radius of transmission and the sensing radius of each node are the same. Hence, by guaranteeing radio-connectivity sensing coverage is also guaranteed.

network of maximum degree Δ, there exists a clique of size $\Omega(\Delta)$. Let x be a node of degree Δ in a network where the radius of transmission is r. Then, there are $\Delta + 1$ nodes located in a circle of radius r centered on x, call this circle C. In order to prove the claim, it is enough to show that there is a circle of radius $\kappa r/2$ inside C that contains at least $\Omega(\Delta)$ nodes. For the sake of contradiction, assume there is no such circle. However, a constant number of circles of radius $\kappa r/2$ are enough to cover completely C. By our assumption, each of these circles contains $o(\Delta)$ of the nodes in C. But then, the total number of nodes in C is in $o(\Delta)$ which is a contradiction. □

Theorem 2. *For any Sensor Network of n nodes, each holding a different message, under the restrictions detailed in Section 3 and under the unit-message model, $\Omega(n)$ time slots are needed in order to solve the Gossiping problem.*

Proof. A node can not send to other nodes messages that are not stored in its memory. Then, a node can not send more than a constant number of messages in one time slot. Hence, nodes can not receive more than a constant number of messages in each time slot. Given that in order to solve the Gossiping problem all n messages must be received by all nodes, the claim follows. □

5. Upper Bounds

In this section, we present Gossiping distributed algorithms for Sensor Networks under the combined-messages model and the unit-message model. Both algorithms include the same preprocessing part, which is also distributed. Preprocessing, as well as main procedures, are conceptually divided into phases. Given that nodes may be powered up at different times, it may be necessary to execute these phases concurrently. The overall structure of both algorithms, including preprocessing, is described in Algorithms 1 and 2, and the details of each phase are presented later.

For the sake of clarity, each phase is presented and analyzed assuming that all nodes run each phase synchronously. For instance, while nodes run the first phase of preprocessing no node is running any other phase. After each analysis, it is shown that such assumption can be removed so that different nodes can run different phases, but that the computation will complete within the claimed time bounds. Given that in order to solve the Gossiping problem all nodes have to be up and running, we analyze the overall running time after the last node starts running the algorithm, and we assume that no node turns off before completion.

Algorithm 1: Gossiping algorithm for linear node memory. Pseudocode for node i.

```
delegate ← hierarchy-definition(i)
if delegate then
    slot-reservation(i)
    cobegin
        collection-delegate(i)
        dissemination-delegate(i)
    coend
else
    cobegin
        collection-dependent(i)
        dissemination-dependent(i)
    coend
```

5.1. Preprocessing

Both Gossiping algorithms include a preprocessing part composed of the following two phases.

- **Hierarchy Definition.** Label some nodes as ***delegate nodes*** in such a way that every node is within distance at most αr of some delegate, and all delegate nodes are separated by a distance larger than αr, where r is the maximum range of transmission and α is a constant such that $0 < \alpha < 1/4$. All non-delegate nodes are called ***dependent nodes***. Each dependent chooses one of the delegates in its range to carry out the Gossiping computation on its behalf. We say that such delegate is the ***dependent's delegate***.
- **Slots Reservation.** Every delegate node reserves blocks of b (we use $b = 7$ in this paper) consecutive time slots for local use, so that any two different delegates located at distance at most r reserve different blocks.

Algorithm 2: Gossiping algorithm for constant node memory. Pseudocode for node i.

```
delegate ← hierarchy-definition(i)
if delegate then
    slot-reservation(i)
    cobegin
        chain-definition-delegate(i)
        tree-definition(i)
        shift-delegate(i)
    coend
else
    cobegin
        chain-definition-dependent(i)
        shift-dependent(i)
    coend
```

Communication Remarks. Communication between dependents and delegates uses a transmission radius of αr. The choice of the upper bound on α and the slot reservation guarantee that communication between delegate and dependent nodes is achieved without interference from any delegate-dependent pair in a radius r of the delegate. Similarly, using a radius of transmission $\beta r, 2\alpha \leq \beta \leq 1/2$, delegate nodes at βr distance can communicate without interference from other nodes in a radius of r.

Given that the hexagonal lattice is the densest of all possible plane packings [32], every dependent node is within distance αr of less than 6 delegate nodes (see Figure 1(a)). The following lemma uses the same argument to give a bound on the number of delegate nodes within a parametrized radius of any delegate node.

Lemma 3. *Since two delegates are at a distance larger than αr, there are at most $\mathcal{D}(\rho) = 3\lceil 2\rho/\alpha\sqrt{3}\rceil(\lceil 2\rho/\alpha\sqrt{3}\rceil + 1) \in O((\rho/\alpha)^2)$ delegate nodes within distance ρr of any delegate node.*

Proof. In order to prove the desired bound, we upper bound the number of delegate nodes in any circle of radius ρr centered on a delegate node. (See Figure 1b.) In order to do that, we take advantage of the fact that any pair of delegate nodes must be separated by a distance at least αr. Notice that, under this restriction, it is not possible to accommodate an $\omega(r)$ number of delegate nodes in a circle of radius $O(r)$. To obtain a precise bound (up to constants) we use a previous result on the optimal way of packing separated points in the plane, the details follow.

All delegate nodes are separated by a distance of at least αr by definition of a delegate node. Consider the smallest regular hexagon whose side is a multiple of αr and covers completely a circle of radius ρr (see Figure 1(b)). Consider a tiling of such hexagon with equilateral triangles of side αr. As proved by Fejes-Tóth in 1940 [32], the hexagonal lattice is indeed the densest of all possible plane packings. In other words, in order to accommodate the maximum number of plane points such that the distance between each pair of points is lower bounded, the optimal point layout is the hexagonal one illustrated in Figure 1(b). Therefore, the number of vertices in such a tiling minus one is an upper bound on the number of delegate nodes at a distance ρr of a delegate node located in the center of such a hexagon. That number is $3\lceil 2\rho/\alpha\sqrt{3}\rceil(\lceil 2\rho/\alpha\sqrt{3}\rceil + 1)$. □

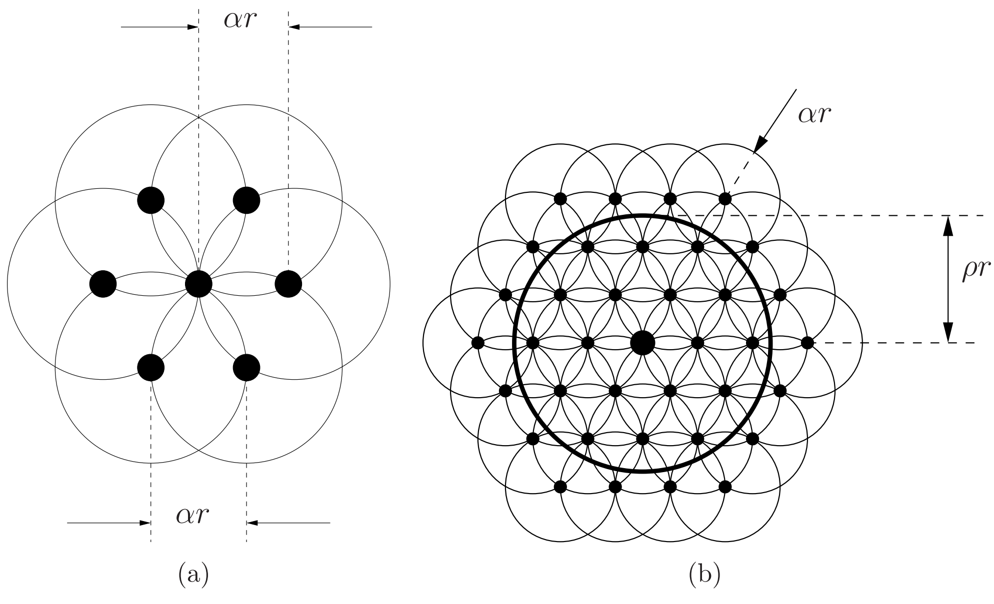


Figure 1: Illustration of maximum degree.

5.1.1. Hierarchy Definition

This phase of preprocessing can be implemented distributedly running a Maximal Independent Set (MIS) algorithm on the nodes adjusting their range of transmission to αr. Each member of the MIS becomes a delegate and all the other nodes become dependents. For that purpose the algorithm presented in [33] is used. In an initial bounding stage, the number of neighboring nodes that will participate in the second stage is upper bounded by $O(\log n)$. In a second stage, nodes keep a counter of the time passed since their first transmission or the last reception of a sufficiently close neighbor-counter. A long enough time without receiving a neighbor's counter enables a node to declare itself a member of the MIS with low probability of error. The second stage, tailored for the Sensor Network setting was presented in [27]. We refer the reader to those papers for further details.

Lemma 4. *For any node i running the Hierarchy Definition algorithm, with high probability, there is at least one delegate within distance αr of i and no two delegates are within range of each other within $O(\log^2 n)$ time slots.*

Proof. As in [33], modified as described in Section 5.1.3 to handle concurrency, and folding in the constant the reserved time slots. □

5.1.2. Slots Reservation

This phase is implemented using a counter to break symmetry as in the previous algorithm. The main idea is for each delegate node to reserve certain consecutive time slots for deterministic transmissions in a way that there are no collisions.

The algorithm works as follows. α_1, α_2, α_3, α_4, b, and γ are constants. Each delegate node x maintains a slot counter, initially set to 0, and a list of incoming reserved slots, initially set to empty. In each slot still not reserved by any of the delegate nodes within distance r from x, x transmits its counter and its identity with probability $1/\alpha_1 \log n$ within radius of r. In each slot that x does not transmit, it is in receiving mode. If x receives the value of a neighbor's counter which is ahead or behind x's counter by at most $\alpha_2 \log n$, x resets its counter to 0. Upon reaching a final count of $\alpha_3 \log^2 n$, x chooses a block of b contiguous available time slots to be used periodically with period γ.

Next, x informs to the neighboring delegate nodes which are the slots it has chosen. In order to do that, x transmits a message containing the number of slots after the current slot in which its reserved block takes place. This

Algorithm 3: Slot reservation algorithm. Pseudocode for delegate d. α_1, α_2, α_3, α_4, b and γ are constants.

```
slot-reservation(d)

begin
    i ← 0
    j ← 0
    incoming-reserved-slots ← emptylist
    status ← undecided
    start tasks 1 and 2 concurrently and return control
end

task 1
    foreach time slot while status is undecided do
        if current time slot is not reserved then
            transmit ⟨d, i⟩ with probability 1/(α1 log n) and radius r
            increase i
            if i ≥ α3 log^2 n then
                reserved ← b contiguous non-reserved slots in an interval
                of γ
                update incoming-reserved-slots with reserved
                status ← decided
    foreach time slot while j < α2 log n do
        if current slot is not reserved then
            transmit ⟨d, reserved⟩ with probability 1/α4 and radius r
            increase j
    foreach time slot do
        if current slot is reserved for the beacon of this delegate then
            transmit a beacon message with radius αr in slot A
end task 1

task 2
    foreach message received in a non-reserved slot do
        case reservation message ⟨d′, r⟩
            update incoming-reserved-slots with r
        case slot counter message ⟨d′, sc⟩
            if status is undecided and |i − sc| ≤ α2 log n then
                i ← 0
end task 2
```

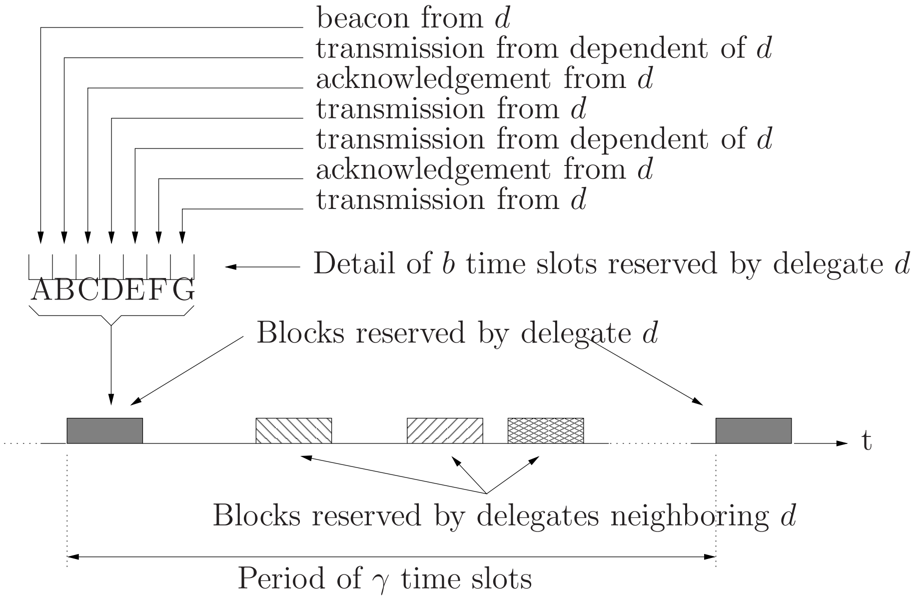


Figure 2: Illustration of time slots usage by delegate node d and some of its neighboring nodes. The algorithm for unrestricted memory uses only slots A to D, whereas the algorithm for restricted memory uses all slots.

message is repeatedly transmitted with probability $1/\alpha_4$, radius r, and using only non-reserved slots. Delegate nodes within distance of r from this node are guaranteed to receive this message within $O(\log n)$ slots and no neighboring delegate node can reach its final count before that w.h.p., as proved below. After $\alpha_2 \log n$ non-reserved slots, the delegate node synchronizes its dependents by repeatedly transmitting a ***beacon message*** with radius αr. After the first beacon message, dependent nodes stop running preprocessing and move to the main procedure. Further details can be seen in Algorithm 3.

The block of b reserved slots is big enough to include slots for dependent transmissions, delegate acknowledgements to dependent transmissions, the beacon message, and transmissions among delegate nodes. The specific number of slots of each kind can be suited to each algorithm. In this paper, we use blocks of $b = 7$ slots (see Figure 2).

The period γ is a constant big enough to ensure that each delegate node gets to reserve some block. As shown in Lemma 3, the number of delegate nodes in any circle of radius r is bounded by $\mathcal{D}(1) \in O(1)$. Thus, we set γ to a constant value such that $\gamma \geq (2b-1)(\mathcal{D}(1)+1)$, so that all delegate nodes are able to reserve a block.

The proof of the following lemma uses the same techniques used in Theorem 6.6 in [27], changing probabilities and constants appropriately. We include in the Appendix the auxiliary lemmas appropriately modified for completeness. (Further modifications along the same lines are needed to handle concurrency, as described in Section 5.1.3.)

Lemma 5. *For any delegate i running Algorithm 3, with high probability, i reserves a block of $b \in O(1)$ slots every $\gamma \in O(1)$ slots, and this block does not overlap with the block reserved by any other delegate node separated by a distance at most r from i, within $O(\log^2 n)$ time slots.*

Proof. From Lemmas 18 and 17 in the Appendix the claim holds for any given delegate with probability at least $1 - n^{-\kappa}$, for some $\kappa > 1$. Given that there are less than n delegate nodes, the claim follows. □

5.1.3. Concurrency and Lack of Synchrony

As shown, choosing γ big enough, there is always some non-reserved slot available every γ slots. Nodes running the Hierarchy Definition phase use non-reserved slots w.h.p. because, as shown in [33], the MIS algorithm that implements the Hierarchy Definition phase includes a waiting phase where, w.h.p., a node i starting up receives from all neighboring delegates before producing any transmission. If while running the Hierarchy Definition phase a non-delegate node receives the beacon of some delegate node, the non-delegate node becomes dependent and it leaves the Hierarchy Definition phase immediately. Likewise, if a delegate x running the Slot Reservation phase (using also non-reserved slots w.h.p.) receives a reservation message from another delegate y, stops using that slot immediately. As shown, w.h.p., the reservation message of y is received before y starts using its reserved slots to communicate with its dependents.

Regarding concurrency, choosing the constant factors of the probabilities appropriately, nodes running the hierarchy definition phase and the slot reservation phase do not interfere with each other w.h.p. The intuition follows. Given that there are $\mathcal{D}(2) \in O(1)$ delegates located within distance $2r$ of any delegate (Lemma 3) and that each delegate in the Slot Reservation phase transmits with probability $1/(\alpha_1 \log n)$ or $1/\alpha_4$, the probability that all these nodes do not transmit in a time slot is lower bounded by a constant ($(1 - 1/(\alpha_1 \log n))^{\mathcal{D}(2)}$ or $(1 - 1/\alpha_4)^{\mathcal{D}(2)}$). As for the nodes running the Hierarchy Definition phase, they either use constant probability of transmission after becoming delegates, or they are upper bounded by a logarithmic number when they use inverse logarithmic probability of transmission, or they use inverse linear probability otherwise. Hence, their silence probability is also lower bounded by a constant. Hence, in order to still be able to show for instance that a successful transmission still occurs w.h.p. it is enough to tune the constants in the algorithm to compensate for these additional interferences. Unfortunately, node-density bounds of one phase are needed before analyzing the other phase and vice-versa. But a careful analysis considering both phases simultaneously and using induction can be carried out as in Lemma 18 in the Appendix.

5.2. Local Communication Primitive

In order to implement efficient Gossiping algorithms on top of the delegate-dependent hierarchy defined, a fast communication primitive to send messages from dependent nodes to their delegates is needed. More precisely, let a ***neighborhood*** be the set given by a delegate node and its dependent nodes. We say that a ***successful*** transmission occurs in a neighborhood if exactly one of the dependents transmit in a time slot. We consider the subproblem where all dependent nodes hold one message that must be successfully transmitted to their delegates. We denote a dependent i as ***loaded*** if its delegate did not receive the message that i initially held.

As explained before, dependent nodes periodically receive a beacon message from their delegate node indicating the forecoming available slots for local use. A block of reserved slots includes, among others, a slot for dependent transmission and a slot for delegate acknowledgement. This acknowledgement informs a dependent that its transmission was successful, implementing a collision detection mechanism. Thus, we can take advantage of the local synchronism achieved by the beacon message and the collision detection implemented by the acknowledgement.

A simple randomized algorithm can achieve this task in $O(\Delta \log n)$ steps, but such a multiplicative factor over Δ would yield a suboptimal Gossiping algorithm. In this section, it is shown that it can be done with only a polylogarithmic additive factor over $O(\Delta)$, using the local synchronism achieved in preprocessing.

In order to implement this primitive efficiently, an approach similar to the algorithm presented in [34] could be used. This algorithm solves the problem of realizing arbitrary h-relations in an n-node network with high probability in $\Theta(h + \log n \log\log n)$ steps. In an h-relation, each processor is the source as well as the destination of h messages. However, the algorithm requires that nodes know h, which in our problem is Δ, but such knowledge is not available in our model. Other constructive upper bounds potentially useful [15], require availability of collision detection. Instead, we use a scheme where the only topology information is the size of the whole network n, and that achieves time complexity in $O(\Delta + \log^3 n)$.

The algorithm, called the ***local communication primitive*** throughout, works as follows. Let each block of reserved slots (including time slots for the transmission of a beacon, a reception, and the transmission of an acknowledgement) be called a ***communication step***. In the first part of the delegate algorithm (see Algorithm 4), each iteration of the inner cycle is a communication step and each iteration of the outer cycle is a ***round***. Each iteration of the external while loop in Algorithm 4 is an ***epoch*** of the local communication primitive.

In this section, we call the number of loaded dependents in a neighborhood at any given time the ***density***. For any neighborhood $\mathcal{N}$ and any communication step, the delegate broadcasts to the dependents an estimation of the density in $\mathcal{N}$. We call such estimation the ***density estimator***. Upon receiving such an estimator, the dependent nodes transmit with probability the inverse of it (see Algorithm 5). If there is a successful transmission, the delegate acknowledges such an event. Upon receiving such an acknowledgement that dependent stops participating.

It is well known that in fair algorithms, i.e., algorithms where all nodes transmit with the same probability in the same time step, the probability of achieving a unique transmission is maximized when nodes transmit with probability equal to the inverse of the number of nodes. However, nodes do not have any topology information but n. Therefore, in Part 1 of Algorithm 4, the density estimator is halved iteratively starting from n.

Once the estimator is within a constant factor of the density, all nodes can be made to transmit with high probability as long as enough communication steps use the same probability. However, including that many steps for each density trial would yield a suboptimal solution for Gossiping. Thus, each round includes only $\Theta(\log n)$ communication steps if no transmission is successful. Upon receiving a successful transmission, the length of the round is extended by a constant number of steps and the density estimator is decreased by one. In this manner, in one epoch, the total cost of non-successful communication steps within Part 1 of Algorithm 4 is in $O(\log^2 n)$ whereas the total cost of successful transmissions is at most Δ. It will be shown that a constant number of epochs is enough to solve the problem.

Once the estimator is within a constant factor of the density, in expectation all nodes transmit. However, the techniques used to prove high probability bounds require at least a logarithmic density. Therefore, a second part is included in the algorithm to take care of transmissions whenever the density is smaller. It is shown in the analysis that the overhead introduced by this part does not affect the complexity of the Gossiping algorithm.

Algorithm 4: Local communication primitive algorithm. Pseudocode for delegate d. Transmissions and receptions are performed only in communication steps of d. $\kappa = e^{(\tau-1)/(\log^2 n-1)}$ and $\tau = 7(96 \log n + 1)$. All transmissions use radius αr.

`collection-delegate(`d`)`

while `true` *//epoch*
do
    Part 1:
    for *round* $= 1$ *to* $\log n - \log\log n$ **do**
        $j \leftarrow 1$
        $\tilde{\delta} \leftarrow n/2^{\text{round}-1}$
        while $j \leq 96 \log n$ *//communication step*
        do
            transmit a density estimator $\langle d, \tilde{\delta} \rangle$ in slot A (beacon)
            if $\langle d, s, message \rangle$ *is received from dependent* s *in slot B*
            then
                transmit acknowledgement $\langle d, s \rangle$ in slot C
                $\tilde{\delta} \leftarrow \tilde{\delta} - 1$
                $j \leftarrow j - 95$
            else
                $j \leftarrow j + 1$
    Part 2:
    for $2\kappa \log^2 n \ln n$ *communication steps* **do**
        transmit a density estimator $\langle d, \log^2 n \rangle$ in slot A
        if $\langle d, s, message \rangle$ *is received from dependent* s *in slot B* **then**
            transmit acknowledgement $\langle d, s \rangle$ in slot C

Algorithm 5: Local communication primitive algorithm for dependent s. Transmissions and receptions are performed only in communication steps of the delegate of s with radius αr.

```
collection-dependent(s)

repeat
    upon receiving ⟨d, δ̃⟩ in slot A (beacon)
    transmit ⟨d, s, message⟩ with probability 1/δ̃ in slot B
until an acknowledgement ⟨d, s⟩ is received in slot C
```

The following lemma, shows the efficiency and correctness of the second part of the delegate algorithm together with the dependent algorithm for small density. First, we state the following useful fact.

Fact 1. *[35, §2.68]* $e^{-x} \geq 1 - x \geq e^{-x/(1-x)}, 0 < x < 1$.

Lemma 6. *For any neighborhood formed by delegate i running Algorithm 4 and set of dependents S running Algorithm 5, if at time t the number of loaded dependents is at most $\tau = 7(96 \log n + 1)$, with probability at least $1 - 1/n^2$ the delegate i receives their messages within $O(\log^3 n)$ time slots after t.*

Proof. Recall that $\kappa = e^{(\tau-1)/(\log^2 n-1)}$. The probability for any of those dependent nodes of failing to transmit in $2\kappa \log^2 n \ln n$ communication steps is

$$\begin{aligned}
Pr_{fail} &\leq \left(1 - \frac{1}{\log^2 n}\left(1 - \frac{1}{\log^2 n}\right)^{\tau-1}\right)^{2\kappa \log^2 n \ln n}, \text{ using Fact 1,} \\
&\leq \left(1 - \frac{1}{\kappa \log^2 n}\right)^{2\kappa \log^2 n \ln n}, \text{ using again Fact 1,} \\
&\leq e^{-2\ln n}, \\
&= n^{-2}.
\end{aligned}$$

Given that each communication step is executed in one block of b reserved slots every $\gamma \in O(1)$ time slots, and that $\kappa \in O(1)$, the claim follows. □

For the case where the number of dependents with a message to transmit is larger than τ, consider the first part of the delegate algorithm. Let the rounds be numbered as $r \in \{1, 2, \ldots, \log n - \log\log n\}$ and the communication steps within a round as $t \in \{1, 2, \ldots\}$. For a given neighborhood $\mathcal{N}$, let $X_{r,t}$ be an indicator random variable such that, $X_{r,t} = 1$ if there is a successful transmission in $\mathcal{N}$ at the communication step t of round r, and $X_{r,t} = 0$ otherwise. Let $\delta_{r,t}$ be the number of dependent nodes in $\mathcal{N}$ that still did not transmit their message to the delegate successfully at the beginning of communication step t of round r. Let $\tilde{\delta}_{r,t}$ be the density estimator broadcasted by $\mathcal{N}$'s delegate at communication step t of round r. Then,

$$Pr(X_{r,t} = 1) = \frac{\delta_{r,t}}{\tilde{\delta}_{r,t}} \left(1 - \frac{1}{\tilde{\delta}_{r,t}}\right)^{\delta_{r,t}-1}$$

Also, for a round r, let the number of successful transmissions in the interval of communication steps $[1, t)$ of r be $\sigma_{r,t}$. The following intermediate results will be useful.

Lemma 7. *For any round r where $\tilde{\delta}_{r,1} \leq 4\delta_{r,1}/5$, for all $t \geq 1$ in r such that $\delta_{r,t} > 2$, the random variables $X_{r,t'}$, are not positively correlated.*

Proof. To prove this claim, we first prove that, under the conditions of the lemma, $Pr(X_{r,t} = 1)$ is monotonically non-increasing with respect to t. Consider any communication step $t \geq 1$ in r where $\delta_{r,t} > 2$. We want to show that $Pr(X_{r,t} = 1) \geq Pr(X_{r,t+1} = 1)$. If there is no successful transmission at step t, then $Pr(X_{r,t} = 1) = Pr(X_{r,t+1} = 1)$, because $\delta_{r,t+1} = \delta_{r,t}$ and $\tilde{\delta}_{r,t+1} = \tilde{\delta}_{r,t}$. If there is a successful transmission at step t, then $\delta_{r,t+1} = \delta_{r,t} - 1$ and $\tilde{\delta}_{r,t+1} = \tilde{\delta}_{r,t} - 1$. Then, we want to show

$$\begin{aligned}
\frac{\delta_{r,t}}{\tilde{\delta}_{r,t}} \left(1 - \frac{1}{\tilde{\delta}_{r,t}}\right)^{\delta_{r,t}-1} &\geq \frac{\delta_{r,t+1}}{\tilde{\delta}_{r,t+1}} \left(1 - \frac{1}{\tilde{\delta}_{r,t+1}}\right)^{\delta_{r,t+1}-1} \\
&= \frac{\delta_{r,t} - 1}{\tilde{\delta}_{r,t} - 1} \left(1 - \frac{1}{\tilde{\delta}_{r,t} - 1}\right)^{\delta_{r,t}-2}.
\end{aligned}$$

Which is equivalent to

$$\frac{\tilde{\delta}_{r,t} - 2}{\tilde{\delta}_{r,t}} \left(\frac{(\tilde{\delta}_{r,t} - 1)^2}{\tilde{\delta}_{r,t}(\tilde{\delta}_{r,t} - 2)}\right)^{\delta_{r,t}-1} \geq \frac{\delta_{r,t} - 1}{\delta_{r,t}}.$$

Using calculus, it can be seen that the left-hand side has a minimum for $\tilde{\delta}_{r,t} = \delta_{r,t}$ and is monotonic for $\tilde{\delta}_{r,t} < \delta_{r,t}$. Then, given that $\tilde{\delta}_{r,t} = \tilde{\delta}_{r,1} - \sigma_{r,t} \leq 4\delta_{r,1}/5 - \sigma_{r,t} \leq 4\delta_{r,t}/5$, it is enough to show

$$\frac{4\delta_{r,t}/5 - 2}{4\delta_{r,t}/5}\left(\frac{(4\delta_{r,t}/5 - 1)^2}{4\delta_{r,t}/5(4\delta_{r,t}/5 - 2)}\right)^{\delta_{r,t}-1} \geq \frac{\delta_{r,t} - 1}{\delta_{r,t}}.$$

Which is equivalent to

$$\frac{4\delta_{r,t} - 10}{4\delta_{r,t} - 4}\left(\frac{(4\delta_{r,t} - 5)^2}{4\delta_{r,t}(4\delta_{r,t} - 10)}\right)^{\delta_{r,t}-1} \geq 1.$$

Again using calculus, it can be seen that the left-hand side is monotonically non-decreasing for any $\delta_{r,t} \geq 3$.

We complete the proof as follows. Consider the beginning of any time step $t > 1$ in round r where more than two messages are left to transmit. Let q be the value of the probability $Pr(X_{r,t} = 1)$. Consider all time steps $t' < t$ when there was not a successful transmission ($X_{r,t'} = 0$), if there was any. If instead for any of those steps there would have been a successful transmission ($X_{r,t'} = 1$), the number of successful transmissions before t would have been bigger and as we showed above it would have been $Pr(X_{r,t} = 1) \leq q$. Consider now all time steps $t'' < t$ when there was a successful transmission ($X_{r,t''} = 1$), if there was any. If instead for any of those steps there would not have been a successful transmission ($X_{r,t''} = 0$), the number of successful transmissions before t would have been smaller and as we showed above it would have been $Pr(X_{r,t} = 1) \geq q$. Hence, the claim holds.

□

The following lemma gives a lower bound on the probability of successful transmission at a given step under certain conditions.

Lemma 8. *For any round r where $\delta_{r,1} \geq \tilde{\delta}_{r,1} \geq 2\delta_{r,1}/5$, and for any communication step t in r where $\sigma_{r,t} \leq \delta_{r,1}/7 - 1$, the probability of a successful transmission is at least $Pr(X_{r,t} = 1) \geq 1/12$.*

Proof. We want to show

$$\frac{\delta_{r,t}}{\tilde{\delta}_{r,t}}\left(1 - \frac{1}{\tilde{\delta}_{r,t}}\right)^{\delta_{r,t}-1} \geq \frac{1}{12}.$$

Given that all nodes are awake, it is enough to show

$$\frac{\delta_{r,1}-\sigma_{r,t}}{\tilde{\delta}_{r,1}-\sigma_{r,t}}\left(1-\frac{1}{\tilde{\delta}_{r,1}-\sigma_{r,t}}\right)^{\delta_{r,1}-1-\sigma_{r,t}} \geq \frac{1}{12}$$

Using calculus, it can be seen that the left hand side has a maximum for $\tilde{\delta}_{r,1} = \delta_{r,1}$ and it is monotonically non-increasing for $\tilde{\delta}_{r,1} < \delta_{r,1}$. Then, it is enough to show

$$\frac{\delta_{r,1}-\sigma_{r,t}}{2\delta_{r,1}/5-\sigma_{r,t}}\left(1-\frac{1}{2\delta_{r,1}/5-\sigma_{r,t}}\right)^{\delta_{r,1}-1-\sigma_{r,t}} \geq \frac{1}{12}.$$

Bounding $\frac{\delta_{r,1}-\sigma_{r,t}}{2\delta_{r,1}/5-\sigma_{r,t}} \geq 5/2$, it is enough to show

$$\frac{5}{2}\left(1-\frac{1}{2\delta_{r,1}/5-\sigma_{r,t}}\right)^{\delta_{r,1}-1-\sigma_{r,t}} \geq \frac{1}{12}.$$

Using Fact 1, it is enough to show

$$\frac{5}{2}\left(\exp\left(\frac{\delta_{r,1}-\sigma_{r,t}-1}{2\delta_{r,1}/5-\sigma_{r,t}-1}\right)\right)^{-1} \geq \frac{1}{12}.$$

Which is equivalent to

$$\delta_{r,1} \geq \frac{5\ln 30-5}{2\ln 30-5}\sigma_{r,t} + \frac{5\ln 30-5}{2\ln 30-5}.$$

Bounding the right-hand side, it is enough to show

$$\delta_{r,1} \geq \frac{5\ln 30-5}{2\ln 30-5}(\sigma_{r,t}+1).$$

Which is true for $\sigma_{r,t} \leq \delta_{r,1}/7-1$. □

The following lemma, shows the efficiency and correctness of the first part of the delegate algorithm, together with the dependent algorithm for the case when the density is more than $7(96\log n+1)$.

Lemma 9. *For any neighborhood formed by delegate i running Algorithm 4 and set of dependents S running Algorithm 5, if at time t the number of loaded dependents is more than $\tau = 7(96\log n+1)$, with probability at least $1-1/n^{\xi}$, where $\xi > 1$, that number is reduced to at most τ within $O(\Delta+\log^2 n)$ time slots after t.*

Proof. Consider the first round r such that $4\delta_{r,1}/5 \geq \tilde{\delta}_{r,1} > 2\delta_{r,1}/5$. Recall that all nodes are assumed to stay on until the problem is solved and that we are analyzing the algorithm for now assuming that nodes run the primitive synchronously. Therefore, in one epoch, either all nodes have achieved a successful transmission or such a round r exists. (There is at least one round unless $n = 3$, in which case the second part of the algorithm guarantees the communication because for $n = 3$ it is $n \approx \log^2 n$.) Even if no transmission was successful, the round would still has at least $96 \log n$ communication steps. So, consider the first $96 \log n$ communication steps of round r. Let Y_1 be a random variable such that $Y_1 = \sum_{i=1}^{96 \log n} X_{r,i}$. Given that $\delta_{r,1} > \tau$, even if there were successful transmissions in each and every step, the total number of successful transmissions would be less than $\delta_{r,1}/7 - 1$. Thus, by Lemma 8, the expected number of successful transmissions in the first $96 \log n$ communication steps is $E[Y_1] \geq (1/12)96 \log n = 8 \log n$. By Lemma 7, the random variables $X_{r,i}$ are not positively correlated. Therefore, in order to bound from below the number of successful transmissions we use Chernoff-Hoeffding bounds as follows.

$$Pr(Y_1 \leq (1-\varepsilon)E[Y_1]) \leq e^{-\varepsilon^2 E[Y_1]/2}$$

Taking for instance $\varepsilon = 7/8$,

$$\begin{aligned} Pr\left(Y_1 \leq \log n\right) &\leq e^{-\frac{49}{16}\log n} \\ &\leq \frac{1}{n^3}. \end{aligned}$$

So, more than $\log n$ nodes achieve a successful transmission w.h.p. Given that each success delays the end of the round 96 communication steps, we know that, w.h.p., after the first $96 \log n$ steps the round will have another $96 \log n$ steps. Conditioned on this event, the same analysis applies to this second set of communication steps obtaining $Pr(Y_2 \leq \log n) \leq 1/n^3$. Furthermore, the analysis is repeated as long as $\sigma_{r,t} \leq \delta_{r,1}/7 - 1$ or the remaining dependent nodes with messages to transmit is at most τ. If the remaining messages are at most τ we are done. Otherwise, the same analysis can be repeated over the next round r' such that $4\delta_{r',1}/5 \geq \tilde{\delta}_{r',1} > 2\delta_{r',1}/5$. Given that for each new round the density estimator is halved, unless the problem is solved before, the round r' exists before completing one iteration of the whole algorithm. Given that the number of dependents in one neighborhood is bounded by Δ

(even using a reduced radius, Δ is an upper bound on the maximum degree), which in turn is bounded by n, the overall number of sets of communication steps along the various rounds is bounded by $s \leq n/\log n$. Let E_i be the event $Y_i > \log n$. Then, using conditional probability, the overall probability of success is

$$
\begin{aligned}
Pr(E_1 E_2 E_3 \ldots E_s) &= Pr(E_1)Pr(E_2|E_1)Pr(E_3|E_1E_2)\ldots Pr(E_s|E_1E_2E_3\ldots E_{s-1}) \\
&\geq \left(1 - \frac{1}{n^3}\right)^{n/\log n}, \text{ using Fact 1 twice },\\
&\geq e^{-n/(n^3-1)\log n} \\
&\geq 1 - \frac{n}{(n^3-1)\log n} \\
&\geq 1 - \frac{1}{n^\xi}, \text{ for some constant } \xi > 1.
\end{aligned}
$$

Given that the goal is achieved in one execution of the first part of the algorithm, the claimed time complexity holds. □

The following lemma shows the overall efficiency of this primitive.

Lemma 10. *With high probability, a delegate nodes running the local communication primitive receives the message of its dependent nodes within* $O(\Delta + \log^3 n)$ *time slots.*

Proof. Consider any neighborhood $\mathcal{N}$. If the number of loaded dependent nodes in $\mathcal{N}$ is at most $7(96\log n+1)$, the delegate of $\mathcal{N}$ receives all messages within $O(\log^3 n)$ times steps with probability at least $1 - 1/n^2$ as shown in Lemma 6. If instead that number is more than $7(96\log n + 1)$, as shown in Lemma 9, after $O(\Delta + \log^2 n)$ steps it is reduced to at most $7(96\log n + 1)$ with probability at least $1 - 1/n^\xi$ for some $\xi > 1$, after which Lemma 6 applies. Given that there are at most n neighborhoods in the whole network, using the union bound the claim follows. □

5.3. An Optimal Algorithm for Linear Memory

We describe now a Gossiping algorithm for Sensor Networks for the combined-messages model. The algorithm has the following two phases.

- **Collection.** Every delegate node maintains a set of messages received, initially containing only its own message. Each dependent node transmits its message to its delegate nodes. Every delegate node adds messages received from its dependents to its set.

- **Dissemination.** Every delegate node transmits its set of messages to all delegate nodes within radius βr, where $2\alpha \leq \beta \leq 1/2$, in slot D, and repeatedly adds the messages received from other delegates and re-transmits. While doing so, dependent nodes receive also these messages.

Given the asynchronous start-up of nodes, these phases and the preprocessing must be executed concurrently. Nevertheless, for the sake of clarity, the details of each phase are analyzed assuming that all nodes start each phase synchronously. Later, it is shown that such assumption can be removed so that different nodes can run different phases, but that the computation will complete within the claimed time bounds.

For the collection phase, we need to guarantee that all dependent nodes transmit their message to their delegates. A straightforward application of the local communication primitive presented in Section 5.2 solves the problem in $O(\Delta + \log^3 n)$ time steps w.h.p. as shown in Lemma 10.

Lemma 11. *Any delegate node running the dissemination algorithm as described receives all messages held by other delegate nodes within $O(D)$ time steps after the last node wakes up, where D is the diameter of the network.*

Proof. Given that the delegate nodes form a maximal independent set with distance αr, $\alpha \in O(1)$, and the assumption of connectivity, the diameter of the subgraph induced by them while using a transmission radius of βr, as well as the diameter of the network D, are both asymptotically bounded by the maximum girth [9] of the convex area of deployment, divided by r. Since delegate nodes re-transmit all messages ever received deterministically every $\gamma \in O(1)$ steps, the claim follows. □

5.3.1. Concurrency and Lack of Synchrony

Nodes may wake up and become dependents after the density estimator broadcasted in the collection phase is below the actual density. Nevertheless, given that time is analyzed only after the last node wakes up, the claimed running time still holds. To see why, consider the time step at which the last

[9] Given a two-dimensional convex body and a given direction, the corresponding *girth* is defined as the length of the orthogonal projection of the body onto a line orthogonal to the assigned direction. The maximum girth is defined as the maximum of the girth over all directions.

node wakes up. By Lemmas 4 and 5 all nodes that are not in the collection or dissemination phases will be in the collection phase within $O(\log^2 n)$ steps after that. Additionally, by definition of the local communication primitive, and due to the fact that all nodes are already awake, in all neighborhoods an appropriate density estimator will be broadcasted within two epochs of the primitive after that time step.

Regarding concurrency, given that nodes running preprocessing use non-reserved slots (refer to Section 5.1.3 for details) there is no conflict between preprocessing and the main procedure. The main procedure is deterministic and utilizes time multiplexing, synchronized by the beacon message. Thus, there is also no conflict among nodes in the collection and dissemination phases.

5.3.2. Overall Analysis

A straightforward application of the lemmata of previous sections, gives the overall running time.

Theorem 12. *Given a network of n nodes, after the last node wakes up, under the combined-messages model, the Gossiping problem can be solved distributedly with high probability in $\Theta(D + \Delta)$ time steps.*

Proof. Using Lemmas 4, 5, 10, and 11 the overall complexity of the algorithm including preprocessing is in $O(\Delta + \log^3 n + D)$ with high probability. Given the geometric constraints, the number of one-hop neighborhoods is bounded by $O(D^2)$. In addition, the maximum number of nodes in any one-hop neighborhood is at most Δ. Hence, at least one of D and Δ has to be in $\Omega(n^\eta)$, for some $\eta > 0$. Thus, the upper bound follows and, given the lower bound of Theorem 1, it is tight. □

5.4. An Optimal Algorithm for Constant Memory

In this section we study the Gossiping problem in Sensor Networks under the unit-message model. I.e., nodes cannot store in memory (and hence transmit in one slot) more than a constant number of pieces information, namely, node identifiers, messages, etc.

Roughly, the algorithm presented in this section is based on an Eulerian tour of a virtually defined tree. Euler-tour traversals of trees have been used [36] for a long time in parallel computing. However, dealing with the various restrictions present in Sensor Networks, to define the tree and the tour and to disseminate the information through it, is not trivial.

The goal of the algorithm is to define a structure among nodes so that messages can be shifted along the structure until all nodes have received at least one copy of each message. Relying on one single node to define such structure is not possible due to the memory limitation. Additionally, given the asynchrony of the nodes start-up schedule, it is necessary to guarantee that sensor nodes waking up late can join the structure. Relying on one single node to handle these arrivals would introduce undesired communication overhead. Therefore, the definition of the structure is done locally.

The algorithm uses the delegate-dependent hierarchy as defined in preprocessing (Section 5.1). Each communication step, i.e., the γ slots reserved for local use, include a slot for the delegate beacon, two slots for dependent transmissions, two slots for delegate acknowledgement of those transmissions, and two slots for communication among delegates (See Figure 2).

The algorithm can be broadly divided in the following phases.

- **Chain Definition.** Define an order among dependent nodes of each delegate node.
- **Tree Definition.** Define a rooted tree among the delegate nodes.
- **Shift.** Repeatedly shift messages throughout that structure.

Given the asynchronous start-up schedule, these phases and the preprocessing must be executed concurrently. Nevertheless, for the sake of clarity, the details of each phase are presented separately and the efficiency is analyzed assuming that all nodes start each phase synchronously. Later, it is shown that such assumption can be removed so that different nodes can run different phases, but that the computation will complete within the claimed time bounds.

5.4.1. Chain Definition

The dependents of a delegate are ordered in a list, or ***chain***, such that each of them knows its predecessor in the chain. To define the chain, each dependent node requests to its delegate to be appended to its current chain. In order to do that, the local communication primitive is used, suited conveniently for this purpose as follows. Recall that in the local communication primitive dependents send messages to their delegate. Here, the message sent is a distinguished request message. On the other hand, delegates acknowledge such request by sending the ID of the requester and the ID of the previous node in the chain or the same ID if it is the first dependent in the chain. Delegate nodes only need to keep track of the last dependent to grant new requests of appending. On the other hand, each dependent node keeps track of its predecessor in the chain. If a dependent is the first node in a chain, it stores the ID of the last node in the chain. I.e., the chain is circular with an entry point at the dependent that first requested to be appended. The following lemma is an immediate consequence of Lemma 10.

Lemma 13. *Any dependent node running the Chain Definition algorithm, with high probability, joins a chain within* $O(\Delta + \log^3 n)$ *time steps.*

5.4.2. Tree Definition

In order to define a rooted-tree among the delegate nodes, we rely upon the existence of at least one node that can be set up conveniently to initiate the following algorithm. Such an assumption is valid given the existence of a sink node. Without loss of generality, we assume that the sink node is a delegate (otherwise, the sink's delegate is used). Using the slots D of their communication steps and starting from the sink, a distinguished signal is broadcasted throughout the network of delegates (See Algorithm 6 and Figure 3). By keeping track of the node from which the signal is received for the first time and by re-transmitting such information, delegates define a tree. Given that a delegate node has a constant number of neighboring delegates (Lemma 3) such bookkeeping is feasible.

Lemma 14. *Any delegate node running Algorithm 6 joins the tree of delegates within* $O(n)$ *time steps after the last node wakes up.*

Proof. Correctness is straightforward by definition of the algorithm. Notice that since the decision of connecting between a parent node and a child is taken uniquely by each node in its role of a child, the link is properly established, i.e., a node is in the children list of its parent and vice-versa. Delegate nodes use reserved time slots, they re-transmit deterministically the signal after $\gamma \in O(1)$ steps, and the depth of the tree is linear in the worst case. Hence, the claim follows. □

5.4.3. Shift Phase

The intuition of this phase follows. Omit first the messages held by the dependent nodes. In order to shift messages among delegate nodes, using the slot G of their communication steps, each delegate passes its message to

Algorithm 6: Algorithm for tree definition for delegate node d. Transmissions are performed in slot D of the communication steps of d, whereas receptions occur in slot D of the communication steps of delegate neighbors of d. All transmissions use a radius βr.

```
tree-definition(d)

children ← emptylist
if d is the distinguished sink node then parent ← null
else parent ← undefined
foreach communication step do
    if parent ≠ undefined then transmit ⟨d, parent⟩
    if a message ⟨d′, d″⟩ is received then
        case parent = undefined
            parent ← d′
        case d″ = d and d′ is not in children
            append d′ to children
```

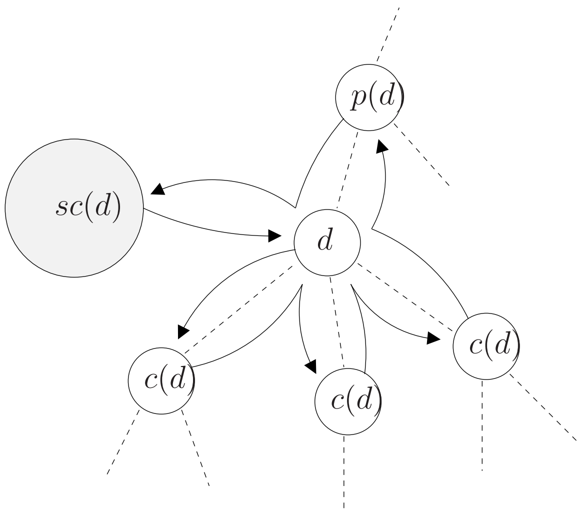


Figure 3: Illustration of Algorithm 6. d: internal node, $p(d)$: parent of d, $c(d)$: child of d, $sc(d)$: dependent chain of d. The arrows indicate the flow of each message.

a neighboring delegate according to the order given by a depth-first-search traversal of the tree. Now consider the messages held by dependent nodes. Each delegate node d introduces the messages of the dependents in its chain in the dissemination by exchanging the message received from its parent delegate [10] with one of its dependents s in the order of the chain. The delegate d passes such a message to s in slot A together with the beacon. By doing so, all of the dependents of d receive that message but only s keeps it. Upon receiving the beacon, s passes its own message to d in slot E. Upon receiving that message, d retransmits the message as an acknowledgement to s in slot F. Although such an acknowledgement is not necessary, it is used to make all the other dependents receive also that message and know which dependent has exchanged messages with d in the current communication step. Given that each dependent node knows its predecessor in its chain, they do not need to receive a specific message from the delegate to know their turn. An exception must be made to initiate this phase when, given that no message was exchanged yet, the first node in the chain exchanges its message. Further details of this phase are given in Algorithms 7 and 8.

Lemma 15. *Every node running the Shift algorithm receives all the messages within $O(n)$ time steps after the last node wakes up.*

Proof. The messages of all nodes are eventually shifted along the nodes of the tree by definition of the algorithm. In a depth-first-traversal, each node is visited at most twice. Additionally, each dependent participates in only one chain. Therefore, there is exactly one copy of each message in the network which is seen by each node $O(1)$ times. On the other hand, each communication step includes $\gamma \in O(1)$ time steps. Thus, the claim follows. □

5.4.4. Concurrency and Lack of Synchrony

Regarding concurrency, given that nodes running preprocessing use non-reserved slots there is no conflict between preprocessing and the main procedure. The main procedure is deterministic and utilizes time multiplexing, synchronized by the beacon message. Thus, there is also no conflict among nodes in the collection and dissemination phases.

Regarding asynchronism, nodes arriving later to the chain definition and shift phases can just join following the algorithms given. For the tree definition phase, given that nodes can only join but do not leave this construction,

[10] If it is the root, use the last child.

Algorithm 7: Algorithm of the shift phase for a delegate node d which is an internal node of the tree. message(d) is the message that d holds, parent(d) is the delegate parent of d, children(i,d) is the ith delegate child of d. The cases where d is the root or a leaf of the tree are omitted for clarity.

```
shift-delegate(d)

messages ← emptylist
append ⟨child(1,d), d, message(d)⟩ to messages
exchange ← null
foreach time slot do
    if the current time slot is not reserved by d then
        if ⟨d, d′, message⟩ is received then
            case d′ = child(i,d) for some i other than the last
                append ⟨child(i+1,d), d, message⟩ to messages
            case d′ is the last child of d
                append ⟨parent(d), d, message⟩ to messages
            case d′ = parent(d)
                exchange ← ⟨d, message⟩
    else
        if exchange ≠ null then
            case slot A reserved by d  (pass parent's message to next
            dependent)
                transmit exchange with radius αr
            case slot E reserved by d  (receive dependent's message)
                receive ⟨d, s, message⟩
                message(d) ← message
            case slot F reserved by d  (acknowledge dependent's
            message)
                transmit ⟨s, d, message⟩ with radius αr
            case slot G reserved by d  (pass children delegates messages)
                transmit messages with radius βr
                messages ← emptylist
                append ⟨child(1,d), d, message(d)⟩ to message
                exchange ← null
```

Algorithm 8: Algorithm of the shift phase for dependent s with chosen delegate d. message(s) is the message that s holds, predecessor(s,d) is the ID of the dependent that precedes s in the chain of d. Transmissions use radius αr.

```
shift-dependent(s)

active ← false
if s is the first node that joined the chain of d then turn ← true
else turn ← false
foreach time slot do
    case slot A reserved by d
        if receive ⟨d, message⟩ then active ← true
    case slot E reserved by d
        if turn and active then
            transmit ⟨d, s, message(s)⟩
            turn ← false
    case slot F reserved by d
        if active then
            receive ⟨s′, d, message⟩
            if s′ = predecessor(s,d) then turn ← true
```

a tree at a given time step is always a super-tree of a previous tree. After a delegate starts running the tree definition algorithm, in at most $O(n)$ time steps (the depth of the tree) after the last delegate starts this phase, a transmission from its parent delegate must be received.

5.4.5. Overall Analysis

A straightforward application of the lemmata of previous sections, gives the main theorem of this section.

Theorem 16. *Given a network of n nodes, after the last node starts running the algorithm described in this section, the Gossiping problem can be solved distributedly with high probability in $\Theta(n)$ time steps.*

Proof. Using Lemmas 4, 5, 10, 14 and 15 the overall complexity of the algorithm including preprocessing is in $O(\Delta + \log^3 n + n)$ with high probability. Therefore, the upper bound follows and, given the lower bound of Theorem 2, it is tight. □

References

[1] M. Farach-Colton, M. A. Mosteiro, Sensor network gossiping or how to break the broadcast lower bound, in: Proc. of the 18th Intl. Symp. on Algorithms and Computation, Vol. 4835 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2007, pp. 232–243.

[2] B. Chlebus, Randomized communication in radio networks, in: P. Pardalos, S. Rajasekaran, J. Reif, J. Rolim (Eds.), Handbook on Randomized Computing, Kluwer Academic Publishers, 2001, Ch. 11.

[3] D. R. Kowalski, On selection problem in radio networks, in: Proc. 24th Ann. ACM Symp. on Principles of Distributed Computing, 2005, pp. 158–166.

[4] R. Bar-Yehuda, O. Goldreich, A. Itai, On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization, Journal of Computer and System Sciences 45 (1992) 104–126.

[5] E. Kushilevitz, Y. Mansour, An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks, SIAM Journal on Computing 27 (3) (1998) 702–712.

[6] D. Liu, M. Prabhakaran, On randomized broadcasting and gossiping in radio networks, Lecture Notes in Computer Science 2387 (2002) 340–349.

[7] B. Chlebus, L. Gąsieniec, A. Lingas, A. Pagourtzis, Oblivious gossiping in ad-hoc radio networks, in: Proc. of 5th Intl. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 2001, pp. 44–51.

[8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci, Wireless sensor networks: A survey., Computer Networks 38 (4) (2002) 393–422.

[9] B. S. Chlebus, D. R. Kowalski, M. A. Rokicki, Average time complexity of gossiping in radio networks, in: Proc. of the 13th Colloquium on Structural Information and Communication Complexity, 2006.

[10] L. Gąsieniec, I. Potapov, Gossiping with unit messages in known radio networks, in: Proc. of 17th IFIP World Computer Congress / 2nd IFIP Intl. Conference on Theoretical Computer Science, 2002.

[11] B. S. Chlebus, D. R. Kowalski, T. Radzik, Many-to-many communication in radio networks, Algorithmica 54 (1) (2009) 118–139.

[12] V. Ravelomanana, Optimal initialization and gossiping algorithms for random radio networks, IEEE Trans. Parallel Distr. Syst. 18 (1) (2007) 17–28.

[13] L. Gąsieniec, A. Pelc, D. Peleg, Y. Emek, E. Kantor, C. Su, Broadcasting in udg radio networks with unknown topology, in: Proc. 26th Ann. ACM Symp. on Principles of Distributed Computing, 2007.

[14] B. S. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, W. Rytter, Deterministic broadcasting in ad hoc radio networks, Distributed Computing 15 (2002) 27–38.

[15] C. U. Martel, Maximum finding on a multiple access broadcast network, Inf. Process. Lett. 52 (1994) 7–13.

[16] R. Bar-Yehuda, A. Israeli, A. Itai, Multiple communication in multihop radio networks, SIAM Journal on Computing 22 (1993) 875–887.

[17] I. Chlamtac, O. Weinstein, The wave expansion approach to broadcasting in multihop networks, in: Proc. of INFOCOM, 1987.

[18] M. Chrobak, L. Gąsieniec, W. Rytter, A randomized algorithm for gossiping in radio networks, Networks 43 (2) (2004) 119–124.

[19] A. Czumaj, W. Rytter, Broadcasting algorithms in radio networks with unknown topology, in: Proc. of the 44th IEEE Ann. Symp. on Foundations of Computer Science, 2003.

[20] L. Gąsieniec, D. Peleg, Q. Xin, Faster communication in known topology radio networks, in: Proc. 24th Ann. ACM Symp. on Principles of Distributed Computing, 2005.

[21] M. Chrobak, L. Gąsieniec, W. Rytter, Fast broadcasting and gossiping in radio networks, in: Proc. of the 41st IEEE Ann. Symp. on Foundations of Computer Science, 2000.

[22] D. Bruschi, M. D. Pinto, Lower bounds for the broadcast problem in mobile radio networks, Distributed Computing 10 (3) (1997) 129–135.

[23] A. Dessmark, A. Pelc, Deterministic broadcasting in ad hoc radio networks, Journal of Discrete Algorithms 5 (2007) 187–201.

[24] A. Clementi, A. Monti, R. Silvestri, Selective families, superimposed codes, and broadcasting on unknown radio networks, in: Proc. of the 12th Ann. ACM-SIAM Symp. on Discrete Algorithms, 2001.

[25] D. R. Kowalski, A. Pelc, Time complexity of radio broadcasting: Adaptiveness vs. obliviousness and randomization vs. determinism, Theoretical Computer Science 333 (2005) 355–371.

[26] A. Kesselman, D. R. Kowalski, Fast distributed algorithm for convergecast in ad hoc geometric radio networks, Journal of Parallel and Distributed Computing 66 (2006) 578–585.

[27] M. Farach-Colton, R. J. Fernandes, M. A. Mosteiro, Bootstrapping a hop-optimal network in the weak sensor model, ACM Transactions on Algorithms 5 (4) (2009) 1–30.

[28] F. Kuhn, T. Moscibroda, R. Wattenhofer, Initializing newly deployed ad hoc and sensor networks, in: Proc. of the 10th ACM Ann. Intl. Conference on Mobile Computing and Networking, 2004.

[29] P. Levis, E. Brewer, D. Culler, D. Gay, S. Madden, N. Patel, J. Polastre, S. Shenker, R. Szewczyk, A. Woo, The emergence of a networking primitive in wireless sensor networks, Commun. ACM 51 (7) (2008) 99–106.

[30] Z. Zhou, S. R. Das, H. Gupta, Variable radii connected sensor cover in sensor networks, ACM Transactions on Sensor Networks 5 (2009) 8:1–8:36. doi:http://doi.acm.org/10.1145/1464420.1464428.
URL `http://doi.acm.org/10.1145/1464420.1464428`

[31] W. Song, Y. Wang, X. Li, O. Frieder, Localized algorithms for energy efficient topology in wireless ad hoc networks, in: Proc. of the 5th ACM international Symp. on Mobile ad hoc networking and computing, 2004, pp. 98–108.

[32] L. Fejes-Tóth, über einen geometrischen satz, Mathematische Zeitschrift 46 (1) (1940) 83–85.

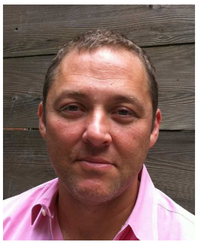
[33] T. Moscibroda, R. Wattenhofer, Maximal independent sets in radio networks, in: Proc. 24th Ann. ACM Symp. on Principles of Distributed Computing, 2005, pp. 148–157.

[34] M. Gerèb-Graus, T. Tsantilas, Efficient optical communication in parallel computers, in: 4th Ann. ACM Symp. on Parallel Algorithms and Architectures, 1992, pp. 41–48.

[35] D. S. Mitrinović, Elementary Inequalities, P. Noordhoff Ltd. - Groningen, 1964.

[36] R. E. Tarjan, U. Vishkin, An efficient parallel biconnectivity algorithm, SIAM Journal on Computing 14 (1985) 862–874.

Authors Biographies



Martín Farach-Colton is a Professor in the Computer Science Department at Rutgers University and CTO of Tokutek Inc. He received his PhD in Computer Science from the University of Maryland in 1991 and his MD from the Johns Hopkins School of Medicine. His primary research interest is in memory-sensitive algorithms for high-performance databases and file systems, particularly Cache-Oblivious Algorithms.

Antonio Fernández Anta is a Senior Researcher at the Institute IMDEA Networks in Leganés, Spain, on leave from his position of Professor at the Universidad Rey Juan Carlos. Previously, he was a member of the faculty at the Universidad Politécnica de Madrid. He graduated in Computer Science from the Universidad Politécnica de Madrid in 1991. He got his PhD in Computer Science from the University of Louisiana at Lafayette in 1994 and was a postdoc at the Massachusetts Institute of Technology from 1995 to 1997. He is Senior Member of the IEEE and the ACM. His research interests include data communications, computer networks, distributed processing, algorithms, and discrete and applied mathematics.

Miguel A. Mosteiro is a Research Professor in the Department of Computer Science at Rutgers University. He is also affiliated to the Department of Telematic Systems and Computer Science at Universidad Rey Juan Carlos, Spain. He received his MS and PhD degrees in Computer Science from Rutgers University in 2003 and 2006 respectively. His primary research interests are in algorithmic problems in Radio and Optical Networks, Distributed Computing and Algorithmic Game Theory.

Appendix A. Auxiliary Lemmas for Proof of Lemma 5

Lemma 17. *For any delegate d_1 running Algorithm 3, with probability at least $1 - n^{-\kappa}$, where $\kappa > 1$, d_1 reserves a block of $b \in O(1)$ slots every $\gamma \in O(1)$ slots, within $O(\log^2 n)$ time slots.*

Proof. For clarity, we omit the reserved time slots, which add only a constant factor to the time complexity. Also, we assume $\log n$ and the various constants $\alpha_\bullet$ to be integers for clarity.

Consider an undecided delegate d_1 that starts running Algorithm 3 at time t_1. Let C_1 be the circle of radius r centered at d_1. Let $\mathcal{E}$ be the event that some delegate node d_2 located in C_1 transmits successfully while its counter is bigger than $\alpha_2 \log n$. We show first that, by time $t_1 + O(\log^2 n)$, there is some delegate node located in C_1 that has reserved a block w.h.p. In order to do that, we show first that, within $O(\log n)$ time steps, there is a constant probability that $\mathcal{E}$ occurs. Notice that if $\mathcal{E}$ occurs all the delegate nodes in a circle C_2 of radius r centered at d_2 reset their counter to 0 (Line 31). (Also notice that even if some delegate starts running Algorithm 3 later it will start with a 0 counter.) Hence, even if d_2 receives the transmission of any of those delegates later, d_2 will not reset its counter until reaching the final count in Line 13 and become decided.

The event $\mathcal{E}$ occurs if (a) d_2 achieves a successful transmission and (b) no other delegate in C_2 transmits its counter (Line 11) for more than $\alpha_2 \log n$ steps before such transmission. Given that there are at most $\mathcal{D}(2)$ delegate nodes in a circle of radius $2r$ and at least one delegate node in C_1, the probability that some delegate node d_2 achieves a successful transmission in one step is at least $(1/(\alpha_1 \log n))\,(1 - 1/(\alpha_1 \log n))^{\mathcal{D}(2)-1}\,(1-1/\alpha_4)^{\mathcal{D}(2)-1}$, whereas the probability that none of the delegates in C_2 transmit its counter for k slots is at least $(1 - 1/(\alpha_1 \log n))^{\mathcal{D}(1)k}$. (Notice that decided delegates do not transmit their counter.) Then,

$$\begin{aligned}
Pr(\mathcal{E}) &\geq \sum_{k=1+\alpha_2 \log n}^{2\alpha_2 \log n} \left(1 - \frac{1}{\alpha_1 \log n}\right)^{\mathcal{D}(1)k} \frac{1}{\alpha_1 \log n} \left(1 - \frac{1}{\alpha_1 \log n}\right)^{\mathcal{D}(2)} \left(1 - \frac{1}{\alpha_4}\right)^{\mathcal{D}(2)} \\
&\geq \left(1 - \frac{1}{\alpha_1 \log n}\right)^{2\mathcal{D}(1)\alpha_2 \log n} \sum_{k=1+\alpha_2 \log n}^{2\alpha_2 \log n} \frac{1}{\alpha_1 \log n} \left(1 - \frac{1}{\alpha_1 \log n}\right)^{\mathcal{D}(2)} \left(1 - \frac{1}{\alpha_4}\right)^{\mathcal{D}(2)} \\
&= \left(1 - \frac{1}{\alpha_1 \log n}\right)^{2\mathcal{D}(1)\alpha_2 \log n + \mathcal{D}(2)} \frac{\alpha_2}{\alpha_1} \left(1 - \frac{1}{\alpha_4}\right)^{\mathcal{D}(2)} .
\end{aligned}$$

Using Fact 1,

$$Pr(\mathcal{E}) \geq \exp\left(-\frac{2\mathcal{D}(1)\alpha_2 \log n + \mathcal{D}(2)}{\alpha_1 \log n - 1}\right) \frac{\alpha_2}{\alpha_1} \left(1 - \frac{1}{\alpha_4}\right)^{\mathcal{D}(2)} .$$

Which is lower bounded by a constant as we claimed. Now fix some constant $\kappa > 1$ and let $a = (\kappa + \log_n \mathcal{D}(1))/\log(1/(1 - Pr(\mathcal{E}))) \in O(1)$.

Then, the probability that $\mathcal{E}$ does not occur by $t_1 + 2\alpha_2 a \log^2 n$ is at most $(1 - Pr(\mathcal{E}))^{a \log n} = n^{-a \log(1/(1-Pr(\mathcal{E})))}$.

We have shown that, by time $t_1 + 2\alpha_2 a \log^2 n$, there is some delegate node located in C_1 that has reserved a block w.h.p. Given that there are at most $\mathcal{D}(1)$ delegates in C_1, by time $t_1 + 2\mathcal{D}(1)\alpha_2 a \log^2 n \in t_1 + O(\log^2 n)$, all delegates in C_1, including d_1, have reserved a block with probability at least $1 - \mathcal{D}(1)/n^{a \log(1/(1-Pr(\mathcal{E})))} = 1 - n^{-\kappa}$. Hence, the claim follows. □

Lemma 18. *Given any delegate node d_i that runs Algorithm 3 and becomes decided at time t_i, with probability at least $1 - n^{-\kappa}$, for some $\kappa > 1$, the block reserved by d_i does not overlap with the block reserved by any other delegate in a circle C_i of radius r centered at d_i that became decided until t_i.*

Proof. To prove this lemma, we prove the following three claims simultaneously by induction on the order in which the delegates become decided, with ties broken arbitrarily. We assume $\log n$ and the various constants $\alpha_\bullet$ to be integers for clarity.

(i) The counter of all undecided delegates in C_i is at most $\alpha_3 \log^2 n - \alpha_2 \log n$ at time t_i.

(ii) All delegates in C_i receive d_i's reservation message within $\alpha_2 \log n$ unreserved time slots after t_i.

(iii) The block reserved by d_i does not overlap with the block reserved by any delegate in C_i decided until time t_i.

Base case: Consider the first delegate within the whole network that becomes decided. Call such delegate d_1 and t_1 the time slot at which d_1 becomes decided.

(i) Assume that there is a delegate d_x located within range of d_1 whose counter is greater than $\alpha_3 \log^2 n - \alpha_2 \log n$ at t_1. Because it became decided at t_1, d_1 was running the loop of Line 9 in Algorithm 3 $\alpha_3 \log^2 n$ unreserved time slots before t_1. Because its counter is greater than $\alpha_3 \log^2 n - \alpha_2 \log n$ at t_1, d_x was running the loop of Line 9 in Algorithm 3 at some time slot within the next $\alpha_2 \log n$ unreserved time slots after d_1 did. Afterwards, neither d_1 nor d_x have transmitted without collision, otherwise, one of their counters would have been reset to 0 in Line 31. Let $\mathcal{E}(k)$ denote the event that neither d_1 nor d_x have transmitted without collision within k unreserved time slots. Recall that

in order to receive successfully from d_x at d_1 in a given time slot, all delegates in C_1 must be silent, and viceversa to receive from d_1 at d_x. Using the fact that there are at most $\mathcal{D}(2)$ delegates within distance $2r$ of any delegate shown in Lemma 3 we have

$$Pr(\mathcal{E}(\alpha_3 \log^2 n - \alpha_2 \log n)) \leq \left(1 - 2\frac{1}{\alpha_1 \log n}\left(1 - \frac{1}{\alpha_1 \log n}\right)^{\mathcal{D}(2)}\right)^{\alpha_3 \log n^2 - \alpha_2 \log n}.$$

Using Fact 1,

$$Pr(\mathcal{E}(\alpha_3 \log^2 n - \alpha_2 \log n)) \leq \left(\exp\left(-2\frac{\alpha_3 \log n - \alpha_2}{\alpha_1 \log n} \exp\left(-\frac{\mathcal{D}(2)}{\alpha_1 \log n - 1}\right)\right)\right)^{\log n}.$$

Which is in $O(n^{-\gamma_1})$, for some constant $\gamma_1 > 1$.

(ii) For at least $\alpha_2 \log n$ unreserved time slots after the delegate d_1 decides which block to reserve, no other delegates in C_1 decides w.h.p. as shown above. Thus, only d_1 will be transmitting with probability $1/\alpha_4$ in that period. Let $\mathcal{E}(k)$ denote the event that d_1 does not transmit successfully in k time slots. The probability of failure in $\alpha_2 \log n$ time slots is

$$Pr(\mathcal{E}(\alpha_2 \log n)) \leq \left(1 - \frac{1}{\alpha_4}\left(1 - \frac{1}{\alpha_1 \log n}\right)^{\mathcal{D}(2)}\right)^{\alpha_2 \log n}.$$

Using Fact 1,

$$Pr(\mathcal{E}(\alpha_2 \log n)) \leq \left(\exp\left(-\frac{\alpha_2}{\alpha_4} \exp\left(-\frac{\mathcal{D}(2)}{\alpha_1 \log n - 1}\right)\right)\right)^{\log n}.$$

Which is in $O(n^{-\gamma_2})$, for some constant $\gamma_2 > 1$.

(iii) By definition of d_1 no other delegate node becomes decided before t_1. With respect to other delegates deciding at t_1, as shown above none of them could be located in C_1, in which case the claim holds trivially.

Inductive Step: For $i > 1$, consider the ith delegate that becomes decided. Let such delegate be called d_i, and the time slot at which d_i becomes decided be called t_i. Let C_i be a circle of radius r centered at d_i. We want to show that the claim holds at the end of time slot t_i after d_i became decided assuming the following.

Inductive hypothesis: For any given delegate d_j such that $j < i$, that becomes decided at time t_j, the following holds with probability at least $1 - n^{-\kappa}$, for some $\kappa > 1$. Let C_j be a circle of radius r centered at d_j.

(i) The counter of all undecided delegates in C_j is at most $\alpha_3 \log^2 n - \alpha_2 \log n$ at time t_j.

(ii) All delegates in C_j receive d_j's reservation message within $\alpha_2 \log n$ unreserved time slots after t_j.

(iii) The block reserved by d_j does not overlap with the block reserved by any delegate in C_j decided until time t_j.

If d_i is separated from decided delegates by a distance bigger than $2r$, all claims can be proved using the same argument as in the base case. Otherwise, the interference produced by decided delegates must be taken into account, which we do as follows.

(i) Assume that there is an undecided delegate d_y within range of d_i whose counter is greater than $\alpha_3 \log^2 n - \alpha_2 \log n$ at t_i. Because it became decided at t_i, d_i was running the loop of Line 9 in Algorithm 3 $\alpha_3 \log^2 n$ unreserved time slots before t_i. Because its counter is greater than $\alpha_3 \log^2 n - \alpha_2 \log n$ at t_i, d_y was running the loop of Line 9 in Algorithm 3 in a time slot within the next $\alpha_2 \log n$ unreserved time slots after d_i did. Afterwards, neither d_i nor d_y have sent without collision, otherwise, one of their counters would have been reset to 0 in Line 31. Let $\mathcal{E}(k)$ be the event that neither d_i nor d_y send without collision for k unreserved time slots. Then,

$$Pr(\mathcal{E}(\alpha_3 \log^2 n - \alpha_2 \log n)) \leq$$
$$\left(1 - 2\frac{1}{\alpha_1 \log n}\left(1 - \frac{1}{\alpha_1 \log n}\right)^{\mathcal{D}(2)}\left(1 - \frac{1}{\alpha_4}\right)^{\mathcal{D}(2)}\right)^{\alpha_3 \log^2 n - \alpha_2 \log n}.$$

Using Fact 1,

$$Pr(\mathcal{E}(\alpha_3 \log^2 n - \alpha_2 \log n)) \leq$$
$$\left(\exp\left(-2\frac{\alpha_3 \log n - \alpha_2}{\alpha_1 \log n} \exp\left(-\frac{\mathcal{D}(2)}{\alpha_1 \log n - 1}\right) \exp\left(-\frac{\mathcal{D}(2)}{\alpha_4 - 1}\right)\right)\right)^{\log n}.$$

Which is in $O(n^{-\gamma_3})$, for some constant $\gamma_3 > 1$.

(ii) Let $\mathcal{E}(k)$ denote the event that d_i does not transmit without collision in k time slots. The probability of failure in $\alpha_2 \log n$ time slots is

$$Pr(\mathcal{E}(\alpha_2 \log n)) \leq \left(1 - \frac{1}{\alpha_4}\left(1 - \frac{1}{\alpha_1 \log n}\right)^{\mathcal{D}(2)}\left(1 - \frac{1}{\alpha_4}\right)^{\mathcal{D}(2)}\right)^{\alpha_2 \log n}.$$

Using Fact 1,

$$Pr(\mathcal{E}(\alpha_2 \log n)) \leq \left(\exp\left(-\frac{\alpha_2}{\alpha_4}\exp\left(-\frac{\mathcal{D}(2)}{\alpha_1 \log n - 1}\right)\exp\left(-\frac{\mathcal{D}(2)}{\alpha_4 - 1}\right)\right)\right)^{\log n}.$$

Which is in $O(n^{-\gamma_4})$, for some constant $\gamma_4 > 1$.

(iii) By inductive hypothesis, we know that, for each delegate d_j that is a neighbor of d_i and became decided at time $t_j < t_i$, the counter of d_i at time t_j was behind the counter of d_j enough so that d_i has received d_j reservation message before t_j with probability at least $1 - n^{-\kappa}$. There are a constant number of delegates neighboring d_i, hence the same probability guarantee holds for all delegate neighbors of d_i. Setting $\gamma \geq (2b - 1)(\mathcal{D}(1) + 1)$, there is a block of b unreserved time slots within the next γ time slots even if all delegate neighbors of d_i have reserved blocks before. Then, the block reserved by d_i does not overlap with the block reserved by any delegate neighboring d_i, with probability at least $1 - n^{-\gamma_5}$, for some constant $\gamma_5 > 1$.

□