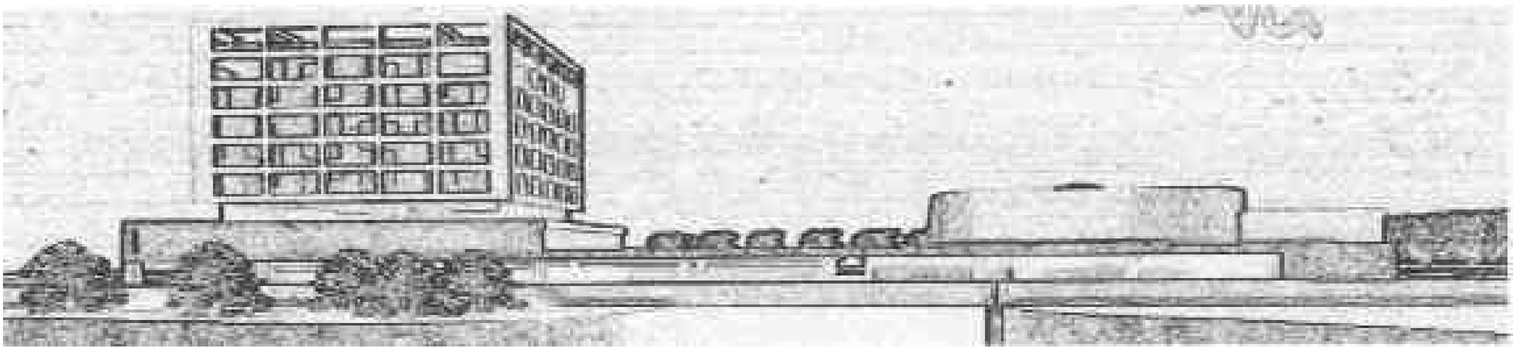
VOLUME IX, NUMBER 4



REPORTS ON SYSTEMS AND COMMUNICATIONS

GSyC

Móstoles (Madrid), Octubre 2009
Depósito Legal: M-50653-2004
ISSN: 1698-7489

Table of Contents

Optimal Memory-aware Sensor Network Gossiping *

Martín Farach-Colton, Antonio Fernández Anta, and Miguel A. Mosteiro

Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA
(farach@cs.rutgers.edu, mosteiro@cs.rutgers.edu).
LADyR, GSyC, Universidad Rey Juan Carlos, 28933 Móstoles, Madrid, Spain
(anto@gsyc.es, miguel.mosteiro@urjc.es).

Abstract. Gossiping is an important problem in Radio Networks that has been well studied, leading to many important results. 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. By exploiting the geometry of sensor node distributions, we present a reduced, optimal running time of $\Theta(D + \Delta)$ for a distributed algorithm that completes gossiping with high probability in a Sensor Network of unknown topology and adversarial start-up, where D is the diameter and Δ the maximum degree of the network. Given that an algorithm for gossiping also solves the broadcast problem, this result shows that the classic broadcast lower bound of Kushilevitz and Mansour does not hold if nodes are allowed to do preprocessing, given that the topology used in that paper is feasible for Sensor Networks. The proposed algorithm requires that a linear number of messages can be stored and transmitted in one time unit. When only a constant number of messages can be stored, an optimal distributed algorithm that solves the problem in linear time is also given.

keywords: radio networks, sensor networks, gossiping, distributed algorithms, broadcast, convergecast.

1 Introduction

The *Radio Network* is a simplified abstraction of a radio-communication network. The question of how to disseminate information within such a network has led to different well-studied problems. Those problems usually differ on the

* This research was supported in part by NSF grants CCF0621425, CCF 05414009, CCF 0632838; DIMACS, Center for Discrete Mathematics & Theoretical Computer Science, grant numbered NSF CCR 00-87022; Comunidad de Madrid grant S-0505/TIC/0285; Spanish MEC grant TIN2005-09198-C02-01, TIN2008–06735-C02-01; and EU Marie Curie International Reintegration Grant IRG 210021. A preliminary version of this article appeared in [15].

number of network nodes holding messages to transmit, the number of different messages to be transmitted and the set of nodes that must receive those messages. A *message* is the piece of information that a node holds, which must be distributed to other nodes. For settings where all nodes in the network must receive all the messages, the problems studied differ on the number of nodes that initially hold those messages, as follows. When k arbitrary nodes have a message, the problem is known as k*-selection* [23]. If $k = 1$, the problem is called *Broadcast* [1, 26], and if $k = n$, the size of the network, it is called *Gossiping* [5, 28]. A subproblem of gossiping is the opposite of broadcast, called *Convergecast* [22], where all nodes hold a message but only one node has to receive all of them. Given that the n messages are all different in general, the size of a message is assumed to be $\Omega(\log n)$ bits. Given that no central controller is feasible in Sensor Networks, a procedure to implement a protocol that solves any of these problems must be distributed. I.e., it must include n algorithms, one for each node, although some nodes may run the same algorithm. We assume that each node runs its own algorithm upon being powered up.

We study the gossiping problem in Sensor Networks. A Sensor Network is a network where n *sensor nodes* with processing, communication, and sensing capabilities are distributed randomly in an area of interest in order to self-organize as a radio-communication network. Sensor Networks are expected to be used to gather information over large remote areas in hostile environments. Sensor nodes have to transmit such information to distinguished nodes called *sinks*. The problem becomes challenging because sensor nodes are subject to strict resource limitations. Since the identity or the location of those sinks are frequently assumed to be unknown to the sensor nodes, gossiping is an important problem and its solution would yield an efficient communication primitive in this setting.

Gossiping in Radio Networks is a well-studied problem for which important results have been obtained under a myriad of models. At least one of the following two crucial assumptions is present in most of these results. (1) It is frequently assumed that the size of a message transmitted in one time slot is bounded only by the size of all the messages in the network, thus, nodes can pad their own message with all messages received and then re-transmit. (2) In addition, a usual assumption is that either nodes start simultaneously or a global clock is available. However, a Sensor Network is a very constrained setting where such assumptions are frequently not feasible. Furthermore, in the most restrictive node-capabilities models used in the literature, such as the Weak Sensor Model [14], the memory size of each node is limited, the time schedule at which nodes start-up, or *wake-up*, is adversarial, and no global clock is available. Although in the asymptotic analysis n tends to infinity, the memory limitation can be relaxed in practice when the magnitude of n is expected to be very large but bounded. However, since the deployment of nodes is produced in hostile or remote large areas, a global clock or a synchronous start of the nodes is too strong of an assumption.

1.1 Related Work

In this paper, as detailed later, we present distributed algorithms for Sensor Networks that complete gossiping in $O(D + \Delta)$ time steps (where D is the diameter and Δ the maximum degree of the network) if the number of messages that can be stored and transmitted in one time unit is linear, and in $O(n)$ time steps if that number is bounded to a constant. Both algorithms are shown to be optimal. In this section, we put these results in the context of previous work.

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 in other problems. To the best of our knowledge, the only previous work analyzing gossiping specifically for Sensor Networks is [32]. More precisely, the problem is studied for the important class of networks with topology modeled by a random geometric graph, 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 contrasting it with the result of Kushilevitz and Mansour [26], but for this lower bound nodes are not allowed to do anything before receiving the broadcast message.

Radio Networks The Sensor Network model is an instance of the Radio Network model with specific geometric properties and node-capabilities restrictions. The prominent literature on gossiping and related problems in Radio Networks is also 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. Differences between the model used in this paper and those assumed in those results are highlighted whenever the bounds obtained can be contrasted with ours.
Upper Bounds. Bar-Yehuda, Israeli, and Itai [2] presented a randomized algorithm for Radio Networks with a topology modeled by an undirected graph [1], that completes gossiping in $O(n \log^2 n)$ on average [2]. Briefly, their technique, used previously in [4] and later re-utilized in [20], 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,

[1] Throughout this paper, the terms undirected graphs, undirected networks and symmetric networks are used indistinctively.
[2] Throughout this paper, log means $\log_2$ unless otherwise stated.

the size of the network n, and an upper bound on the maximum degree. Nodes may transmit and receive only $O(\log n)$ bit messages in synchronous time slots. However, they can store as many as needed. The same bound but with high probability[3] was proved in [5]. Their algorithm relies on unbounded message size and global synchronism. For asymmetric Radio Networks, Chrobak, Gąsieniec, and Rytter [10] showed an upper bound of $O(n \log^3 n \log(n/\epsilon))$ with probability $1-\epsilon$ and $O(n \log^4 n)$ in expectation. The main idea is to repeatedly run a limited broadcast that doubles the number of copies of each message in the network in each phase. Thus, unbounded message size is necessary as well as global synchronism. Using the same algorithm, but improving the limited broadcast by adding randomization to it, Liu and Prabhakaran [28] reduced that upper bound by a logarithmic factor. More recently, Czumaj and Rytter [12] 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 memory and messages are bounded only to the size of all messages. Global synchronism precludes the application of these results to a restricted Sensor Network.

In [7], 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.

The results described above apply to the class of randomized algorithms. Deterministic solutions may also be of interest for comparison. When memory and messages are unbounded, Chrobak, Gąsieniec, and Rytter [9] presented a $O(n^{3/2} \log^2 n)$ algorithm for asymmetric Radio Networks. The algorithm makes use of selecting sequences to ensure non-colliding transmissions. Later, for the same networks, a $O(D + \Delta \log n)$ algorithm, was presented by Gąsieniec, Peleg and Xin in [19], and Chlebus, Kowalski and Rokicki showed in [8] an upper bound of $O(\min\{k \log(n/k), n/\log n\})$, where k is the number of nodes that initially have to pass a message, was proved. In the latter, the time efficiency is given in expectation over a uniform distribution on all strongly-connected networks. For a model where only one message can be transmitted in one time slot, deterministic upper bounds of $O(n^2)$ for asymmetric Radio Networks and $O(n \log^2 n)$ for symmetric Radio Networks were shown in [20] by Gąsieniec and Potapov, and an upper bound of $O(k \log(n/k) \log n)$ on the expected time to complete gossiping in asymmetric Radio Networks was shown in [8] by Chlebus, Kowalski and Rokicki. (Again, k is the number of nodes that initially have to pass a message, and the expectation is taken over a uniform distribution on all

[3] 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 $Pr\{E_p\} \geq 1 - n^{-\kappa}$.

strongly-connected networks.) In all these models, nodes start-up simultaneously and in [19, 20] nodes know the topology. Hence, the upper bounds do not apply to harshly restricted Sensor Networks.

Lower Bounds. As for lower bounds, as stated later, in this paper it is shown that it takes at least $\Omega(\Delta + D)$ time to solve gossiping in Sensor Networks. Chlebus, Gąsieniec, Lingas, and Pangourtzis [5] proved that any deterministic oblivious gossiping algorithm requires at least $n^2/2 - n/2 + 1$ time to complete. In the same paper, for the important 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 for any integer $n \leq q \leq n^2/2$ there exists an asymmetric Radio Network such that the expected time to complete gossiping is $\Omega(q)$. Our algorithm is adaptive and is not in the class of fair algorithms so, neither of these lower bounds hold. Later, Gąsieniec and Potapov [20] showed lower bounds of $\Omega(n^2)$ for asymmetric Radio Networks and $\Omega(n \log n)$ for symmetric Radio Networks. However, the topology of the construction used for the later can not be embedded in geometric graphs, and therefore does not apply to Sensor Networks.

More recently, Chlebus, Kowalski and Radzik [7] 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 distance among any pair of them, even if the number of messages transmitted in one time slot is unbounded. As mentioned before, the problem becomes gossiping if $p = n$. The lower bound showed becomes linear when the number of participants is at least $n/2$. This case is derived from a known existential lower bound of $\Omega(p + d)$, proved using a network of degree $p - 1$. 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. Assuming that nodes do nothing before receiving the message to be broadcasted, a general lower bound for randomized algorithms of $\Omega(D \log(n/D))$ in expectation was obtained by Kushilevitz and Mansour in [26]. Under the same assumption, but using the deterministic nature of an algorithm, Bruschi and Del Pinto [3] proved the existence of networks where $\Omega(D \log n)$ steps are required to solve broadcast. Also using determinism, Dessmark and Pelc [13] proved an existential lower bound for broadcast of $\Omega(D + \log n)$ for symmetric networks even if nodes are allowed to do preprocessing. Clementi, Monti and Silvestri [11] improved the lower bound to $\Omega(n \log D)$ for asymmetric networks even if nodes are allowed to transmit before receiving the message, as well as for symmetric networks when nodes can not transmit before receiving the message. Kowalski and Pelc [24] 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 [18] that in geometric networks broadcasting time depends on the minimum distance between any pair of nodes. None of these bounds can be contrasted with our algorithm because either they do not hold for adaptive algorithms, or they make use of determinism, or the topology of the construction used for the lower bound can not be embedded in geometric graphs.

Lower bounds for convergecast also apply to gossiping. In [22], 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 distance to the closest neighbor. Given that $\max\{D, \Delta\} \in \omega(\log n)$ in our model of Sensor Network, this bound does not contradict our result.

1.2 Our Results

In this paper, the gossiping problem is studied under a strict node-restrictions model that includes the strongest limitations on node resources in the literature on Sensor Networks. As noted in previous work for Radio Networks [8], the number of messages that can be transmitted in one time slot establishes an efficiency gap for deterministic gossiping algorithms. Additionally, given the node-memory restrictions, this limitation is crucial in the Sensor Networks setting as well. Thus, we consider two scenarios. Namely, a first scenario where both, memory and messages per slot, are bounded only by a linear number of messages (a.k.a. *combined-messages model* [8]), and a second more restrictive scenario where both are limited to a constant number of messages (a.k.a. *unit-messages model* [7, 20] or *separate-messages model* [8]). Notice that, even under memory limitations, the gossiping problem is still well motivated, since the requirement is to receive all messages but not to store them. Intermediate bounds for any of these limitations are left for future work.

Assuming that nodes can pad a linear number of messages, we present a randomized distributed algorithm that, given a network of n nodes, with high probability, completes gossiping in time $O(\Delta+D)$ after the last node starts running the algorithm. A preliminary version of this result was presented in [15]. Given that in random geometric graphs $\Delta \in O(\log n)$ and $D \in O(\sqrt{n/\log n})$ w.h.p., this result implies $O(\sqrt{n/\log n})$ under that stronger topology model, improving over [32]. Given that $\Omega(D)$ and $\Omega(\Delta)$ are lower bounds for this problem, our algorithm is optimal. This result improves over previous bounds in time efficiency and makes no assumptions about global synchronism. Rather, it exploits the geometry of the topology in Sensor Networks. This result also shows that the classical expectation lower bound for broadcast of Kushilevitz and Mansour of $\Omega(D \log(n/D))$ [26] does not hold if nodes are allowed to do some preprocessing before receiving a message to transmit. The topology used to prove that lower bound can be embedded in geometric graphs, even under a complete coverage assumption of a convex area, because nodes within each layer of their construction can be arbitrarily close. An efficiency gap between a model where nodes do nothing before receiving the message and a model where nodes may exchange messages upon start-up, sometimes named *conditional wake-up* and *spontaneous wake-up* respectively, was shown before [6, 18] but, to the best of our knowledge, 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 $O(\Delta + polylog(n))$ with high probability is included our algorithms. This primitive estimates the density by probing the channel in a fashion similar to previous work [29], but our technique improves by achieving the probabilistic guarantees needed for the overall algorithm. To do so, we define a local schedule of transmissions among neighboring nodes that implements local synchronism and collision detection.

Finally, for the case where the size of each node's memory is limited to a constant number of messages, the previous algorithm is modified to structure the network so messages can be shifted through the nodes in an ordered manner. This distributed algorithm is shown to be linear which, given the memory restriction, is also optimal.

1.3 Roadmap

In the remainder of this paper we define the models used throughout in Section 2, we establish lower bounds in Section 3, a preprocessing procedure and a communication primitive used in both algorithms are presented in Sections 4.1 and 4.2, and the details of both algorithms are given in Sections 4.3 and 4.4. Conclusions and open problems are included in Section 5.

2 The Model

Radio Networks is a vast area and there is a myriad of applications of such a technology. Depending on the application, topologies and node constraints may be very different. In Sensor Networks, nodes are expected to be deployed in large quantities over an area of interest and two nodes can communicate only if they are mutually in range. Thus, we model the topology as a *Geometric Graph*, where nodes are distributed arbitrarily in a unit convex area in $\mathbb{R}^2$, and a pair of nodes is connected by an edge if and only if they are at an Euclidean distance of at most a parameter r. 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 shortest path between any two vertices of this geometric graph. As customary in Sensor Networks, nodes are assumed to be deployed densely enough to guarantee connectivity and coverage. In this paper, nodes can adjust the transmission power among different levels. By adjusting the power of transmission a node is able to effectively adjust its radius of connectivity. Thus, we also assume that connectivity is guaranteed even while using the smallest power of transmission. The topology is assumed to be unknown to the nodes, and the only knowledge each node has is the number of nodes in the whole network n, and its unique identifier in $[1, n^\kappa]$, where $\kappa \geq 1$ is some arbitrary constant.

In addition to topology and connectivity models, an appropriate model of the constraints under which sensor nodes operate has to be defined, in order to properly design and analyze algorithms. We use a relaxed version of the Weak Sensor Model elucidated in [14] including the following assumptions.

- Low-information channel contention: The communication among neighboring nodes is through broadcast on a shared channel of communication unique throughout the network, where a node receives a message only if exactly one of its neighbors transmits. If more than one message is sent to a node at the same time, a collision occurs and the node receives the messages garbled. Furthermore, no collision detection mechanism is available and sensors nodes cannot receive and transmit in the same time slot. Therefore a node can not distinguish between a collision and no transmission in its neighborhood.
- Time is assumed to be slotted and all nodes have the same clock frequency, but no global synchronizing mechanism is available.
- It is assumed the presence of an adversary that chooses the time instant at which each node is powered up. We say that nodes *wake-up* or *start-up* adversarially.
- It is assumed that sensor nodes can adjust their power of transmission but only to a constant number of levels, always limited to cover a short range much smaller than 1, and with only one channel of communication available. By adjusting its power of transmission a node is able to effectively adjust its radius of connectivity.
- No position information or distance estimation capabilities are available.

The Weak Sensor Model includes also limits on memory size, life cycle and reliability. Regarding memory restrictions, as mentioned before, we consider two cases: node-memory and number of messages that can be transmitted in one time slot limited to a constant messages as in the Weak Sensor Model, and to a linear number of messages for settings where the size of the memory can be relaxed. Regarding power supply restrictions or node reliability, given that in order to solve the gossiping problem all nodes have to be active, we assume that no node turns off before completion of the algorithm. Any time analysis would be meaningless in presence of an adversary that turns on and off nodes forever. A study of the gossiping under a model where failures are feasible but bounded is left for future work.

In order to highlight the relevance of this work, we compare our model with respect to previous models of node constraints. Our model is a relaxation of the Weak Sensor Model [14], which includes the same set of restrictions but where failures are arbitrary and memory is always limited. As mentioned before, arbitrary failures would yield the problem unsolvable. Bar-Yehuda, Goldreich, and Itai [1] 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 [25]. Notice that the unstructured Radio Network model does not include all the restrictions of our model. For instance, that model does not include limits on the number of levels of transmission power and lack of position information. But, more importantly, the unstructured Radio Network model does not include limits on memory size, a fundamental restriction [27].

3 Lower Bounds

The following straightforward theorems establish formally lower bounds for gossiping.

Theorem 1. *Given a Sensor Network of n nodes, where D is the diameter of the network and Δ the maximum degree, under the restrictions detailed in Section 2, and independently of randomization, $\Omega(D+\Delta)$ time slots are needed in order to solve the gossiping problem.*

Proof. $\Omega(D)$ is a trivial lower bound, since messages have to traverse the diameter of the network. 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 network of maximum degree Δ, there exists a clique of such size. 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 $r/2$ 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 $r/2$ 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. *Given a Sensor Network of n nodes, each holding a different message, under the restrictions detailed in Section 2, and independently of randomization, if each node's memory size is limited to a constant number of messages, $\Omega(n)$ 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.

4 Upper Bounds

Gossiping distributed algorithms for Sensor Networks under two node-memory size scenarios, linear and constant, are presented in this section. Both algorithms include a common preprocessing part, which of course is also distributed. Preprocessing, as well as main procedures, are conceptually divided into phases. Given that nodes may be powered up at different times, these phases and the preprocessing must be executed concurrently. The overall structure of both algorithms, including preprocessing, is described in Algorithms 1 and 2. The details of each phase are given in the following sections.

For the sake of clarity, the details of each phase are presented separately and the efficiency is analyzed assuming that nodes start each phase synchronously. I.e., all nodes start running each phase at the same time. After each analysis, it is shown that such an assumption can be removed. As mentioned before, given that in order to solve the gossiping problem all nodes have to be active, 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-slug(i)
        dissemination-slug(i)
    coend
```

4.1 Preprocessing

Both gossiping algorithms include a preprocessing part composed by 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 at least 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 *slug nodes*.
- **Slots Reservation.** Every delegate node reserves blocks of b consecutive time slots for local use, so that each delegate and its slugs can communicate without colliding with transmissions from other nodes within radius r. (We use $b = 7$ in this paper.)

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-slug(i)
        shift-slug(i)
    coend
```

Each slug node is at a distance αr of at least one delegate node. The delegate nodes at distance at most αr of a slug node are the slug's delegates. Communication between slugs and delegates uses a transmission radius of αr. The choice of the upper bound on α guarantees that, once slot reservation has taken place, communication between delegate and slug nodes is achieved in a time slot that is not used by any delegate-slug 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 α and β are constants, its effect is folded in the other constants of the analysis. More precisely, it is assumed that the density of nodes is adjusted by a constant factor in order to still achieve connectivity and coverage using these reduced radii of transmissions.

Given that the hexagonal lattice is the densest of all possible plane packings [16], every slug 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. After that, the details of each phase of preprocessing are given.

Lemma 1. *There are at most* $\mathcal{D}(\rho) = 3\lceil 2\rho/\alpha\sqrt{3}\rceil(\lceil 2\rho/\alpha\sqrt{3}\rceil + 1)$ *delegate nodes within distance* ρr *of any delegate node with high probability.*

Proof. All delegate nodes are separated by a distance of at least αr with high probability as a result of the hierarchy definition phase. 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 [16], the hexagonal lattice is indeed the densest of all possible plane packings. 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 $3\lceil 2\rho/\alpha\sqrt{3}\rceil(\lceil 2\rho/\alpha\sqrt{3}\rceil + 1)$.

Hierarchy Definition This phase of preprocessing can be implemented distributedly running a Maximal Independent Set (MIS) algorithm with radius αr. Each member of the MIS becomes a delegate and all the other nodes become slugs. For that purpose the algorithm presented in [31] is used. In an initial bounding stage, the number of neighboring nodes that will participate in the second stage is upper bounded to $O(\log n)$. 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 [14].We refer the reader to those papers for further details.

Lemma 2. *For each node running the algorithm described above, at least one node within its transmission range of αr joins the MIS in $O(\log^2 n)$ time slots and no two MIS nodes are within range of each other with high probability.*

Proof. As in [31] and [14].

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, x transmits its counter and its identity with probability $1/\alpha_1$ within a 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 less than $\alpha_2 \log n$, x resets its counter to 0. Upon reaching a final count of $\alpha_3 \log n$, 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 message is repeatedly transmitted with probability $1/\alpha_4$, radius r, and using only non-reserved slots. As in [14],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. After $\alpha_2 \log n$ non-reserved slots, the delegate node synchronizes its slugs by repeatedly transmitting a *beacon message* with radius αr. After the first beacon message, slug 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 slug transmissions, delegate acknowledgements to slug transmissions, the beacon message, and transmissions among delegate nodes. The specific number of slots of each

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, 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 $1/\alpha_1$ and radius $r$
            increase i
            if $i = \lceil \alpha_3 \log n \rceil$ then
                reserved ← b contiguous non-reserved slots in an interval of $\gamma$
                update incoming-reserved-slots with reserved
                status ← decided
    foreach time slot while $j < \lfloor \alpha_2 \log n \rfloor$ do
        if current slot is not reserved then
            transmit ⟨d, reserved⟩ with probability $1/\alpha_4$ 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 $\alpha r$ 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 $|i - sc| \le \lfloor \alpha_2 \log n \rfloor$ then
                i ← 0
end task 2
```

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 1, 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.

Lemma 3. *After $O(\log n)$ time slots running the algorithm described above, any delegate node reserves a block of $b \in O(1)$ slots every $\gamma \in O(1)$ slots for local use, i.e., this block does not overlap with the block of any other delegate node separated by a distance at most r, with high probability.*

Proof. The running time of the algorithm can be proved as in [14]replacing *"become MIS member"* for *"reserve a block"* and changing the probabilities of transmission used in the algorithm appropriately.

To complete the proof we consider two cases.

Case 1: we assume for the sake of contradiction that the blocks reserved by some pair of delegate nodes separated by a distance at most r overlap. This implies that at least one of them did not receive the message of the other. But, using the techniques in [14]it can be shown that the probability of that event is $O(1/n^{\psi_{(3)}})$ for some constant $\psi_{(3)} > 2$, which ensures that the probability of failure over all possible pairs is low.

Case 2: we assume for the sake of contradiction that a delegate node x can not reserve a contiguous block of b slots. This implies that after some neighboring delegate nodes reserve their blocks, there are no contiguous b slots available. As proved in Lemma 1 there are at most $\mathcal{D}(1)$ delegate nodes within r distance of any master node w.h.p. But, making $\gamma \geq (2b-1)(\mathcal{D}(1)+1)$ there is always a block of b contiguous slots available w.h.p.

Concurrency and Lack of Synchrony Nodes in the Hierarchy Definition phase use non-reserved slots. As shown, choosing γ big enough, there is always some non-reserved slot available every γ slots. If while running the Hierarchy Definition phase a non-delegate node receives the beacon of some delegate node, the node leaves the Hierarchy Definition phase immediately.

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 within any two-hop neighborhood (Lemma 1) and that each delegate in the Slot Reservation phase transmits with constant probability $1/\alpha_1$ or $1/\alpha_4$, the silence probability in a two-hop neighborhood (i.e., the probability that all these nodes do not transmit in a time slot $(1-1/\alpha_1)^{\mathcal{D}(2)}$ or $(1-1/\alpha_4)^{\mathcal{D}(2)}$) is a constant. As for the nodes running the Hierarchy Definition phase, they either use constant probability of transmission after becoming delegates, or they are upper bounded to 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 [14].

4.2 Local Communication Primitive

In order to efficiently implement gossiping algorithms on top of the delegate-slug hierarchy defined, a fast communication primitive to send messages from slug nodes to their delegates is needed. More precisely, let a *neighborhood* be the set given by a delegate node and its slug nodes. We say that a *successful* transmission occurs in a neighborhood if exactly one of the slugs transmit in a time slot. We consider the subproblem where all slug nodes hold one message that must be successfully transmitted to their delegates.

As explained before, slug 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 slug transmission and a slot for delegate acknowledgement. This acknowledgement informs a node 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, using the local synchronism achieved in preprocessing.

In order to implement this primitive efficiently, an approach similar to the algorithm presented in [21] 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, in our problem Δ, which is not our case. Other constructive upper bounds potentially useful [29], require availability of collision detection. So, 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 delegate algorithm (Algorithm 4), each iteration of the inner cycle of Part 1 is a communication step. Let each iteration of the outer cycle be called a *round*. Each iteration of the external while loop is an execution of the local communication primitive.

For any neighborhood $\mathcal{N}$ and any communication step, the delegate broadcasts to the slugs an estimator of the density of slugs that still have a message to transmit in $\mathcal{N}$. Upon receiving such an estimator, the slug nodes transmit with probability the inverse of it (Algorithm 5). If there is a successful transmission, the delegate acknowledges such an event. Upon receiving such an acknowledgement that slug 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 the first part of the delegate algorithm, 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 solution suboptimal 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 execution of the algorithm, the total cost of non-successful communication steps is in $O(\log^2 n)$ whereas the total cost of successful transmissions is in $O(\Delta)$. It will be shown that a constant number of executions 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 number of messages to be sent. Therefore, a second part is included in the algorithm to take care of transmissions whenever the number of messages left is smaller. It is shown in the analysis that the overhead introduced by this part does not affect the complexity of the gossiping algorithm.

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

Fact 1 *[30, §2.68]* $e^{-x} \geq 1 - x \geq e^{-x/(1-x)}, 0 < x < 1.$

Lemma 4. *For any neighborhood $\mathcal{N}$ where the number of slug nodes that still hold a message is at most $\tau = 7(96 \log n + 1)$, the second part of the delegate algorithm, together with the slug algorithm solve the problem with probability at least $1 - 1/n^2$ in $O(\log^3 n)$ time steps.*

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` **do**
  Part 1:
  for *round* $= 1$ *to* $\log n - \log\log n$ **do**
    $j \leftarrow 1$
    $\tilde{\delta} \leftarrow n/2^{\mathsf{round}-1}$
    while $j \leq 96 \log n$ **do**
      transmit a density estimator $\langle d, \tilde{\delta} \rangle$ in slot A (beacon)
      if $\langle d, s, message \rangle$ *is received from slug* 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 slug* s *in slot* B **then**
      transmit acknowledgement $\langle d, s \rangle$ in slot C

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

`collection-slug(`s`)`

repeat
  upon receiving $\langle d, \tilde{\delta} \rangle$ in slot A (beacon)
  transmit $\langle d, s, message \rangle$ with probability $1/\tilde{\delta}$ in slot B
until *an acknowledgement* $\langle d, s \rangle$ *is received in slot* C

Proof. Recall that $\kappa = e^{(\tau-1)/(\log^2 n-1)}$. The probability for any of those slug 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 slugs 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 slug 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 5. *For any round r where $\tilde{\delta}_{r,1} \leq 4\delta_{r,1}/5$, $Pr(X_{r,t} = 1)$ is monotonically non-increasing with respect to t as long as more than two messages are left to transmit.*

Proof. Consider any communication step t in r. The lemma is proved by induction on the number of successful transmissions $\sigma_{r,t}$. For the base case, assume that $\sigma_{r,t} = 0$ then $Pr(X_{r,i} = 1) = Pr(X_{r,t} = 1), \forall i < t$ because $\delta_{r,i} = \delta_{r,t}$ and $\tilde{\delta}_{r,i} = \tilde{\delta}_{r,t}$, $\forall i < t$. For the inductive step, assume as inductive hypothesis that $Pr(X_{r,t} = 1)$ is monotonically non-increasing in t for some $\sigma_{r,t} = \sigma \geq 0$, we want to prove it for $\sigma_{r,t} = \sigma + 1$. Let $t' < t$ be the communication step at which the last successful transmission occurred before t. Then, $\sigma_{r,t'} = \sigma$ and, by inductive hypothesis, the claim holds for that step. Thus, it is enough to show that $Pr(X_{r,t'} = 1) \geq Pr(X_{r,i} = 1)$, $\forall i \in (t', t]$. Then, we want to show

$$\frac{\delta_{r,t'}}{\tilde{\delta}_{r,t'}}\left(1 - \frac{1}{\tilde{\delta}_{r,t'}}\right)^{\delta_{r,t'}-1} \geq \frac{\delta_{r,i}}{\tilde{\delta}_{r,i}}\left(1 - \frac{1}{\tilde{\delta}_{r,i}}\right)^{\delta_{r,i}-1}, \forall i \in (t', t].$$

Given that all nodes are awake, $\delta_{r,i} = \delta_{r,t'} - 1$ and $\tilde{\delta}_{r,i} = \tilde{\delta}_{r,t'} - 1$, $\forall i \in (t', t]$. Thus, it is enough to show

$$\frac{\delta_{r,t'}}{\tilde{\delta}_{r,t'}}\left(1 - \frac{1}{\tilde{\delta}_{r,t'}}\right)^{\delta_{r,t'}-1} \geq \frac{\delta_{r,t'}-1}{\tilde{\delta}_{r,t'}-1}\left(1 - \frac{1}{\tilde{\delta}_{r,t'}-1}\right)^{\delta_{r,t'}-2}$$

$$\frac{\delta_{r,t'}}{\tilde{\delta}_{r,t'}}\left(\frac{\tilde{\delta}_{r,t'}-1}{\tilde{\delta}_{r,t'}}\right)^{\delta_{r,t'}-1} \geq \frac{\delta_{r,t'}-1}{\tilde{\delta}_{r,t'}-2}\left(\frac{\tilde{\delta}_{r,t'}-2}{\tilde{\delta}_{r,t'}-1}\right)^{\delta_{r,t'}-1}$$

$$\frac{\tilde{\delta}_{r,t'}-2}{\tilde{\delta}_{r,t'}}\left(\frac{(\tilde{\delta}_{r,t'}-1)^2}{\tilde{\delta}_{r,t'}(\tilde{\delta}_{r,t'}-2)}\right)^{\delta_{r,t'}-1} \geq \frac{\delta_{r,t'}-1}{\delta_{r,t'}}.$$

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 $\tilde{\delta}_{r,t'} = \tilde{\delta}_{r,1} - \sigma \leq 4\delta_{r,1}/5 - \sigma \leq 4\delta_{r,t'}/5$, it is enough to show

$$\frac{4\delta_{r,t'}/5-2}{4\delta_{r,t'}/5}\left(\frac{(4\delta_{r,t'}/5-1)^2}{4\delta_{r,t'}/5(4\delta_{r,t'}/5-2)}\right)^{\delta_{r,t'}-1} \geq \frac{\delta_{r,t'}-1}{\delta_{r,t'}}$$

$$\frac{4\delta_{r,t'}-10}{4\delta_{r,t'}-4}\left(\frac{(4\delta_{r,t'}-5)^2}{4\delta_{r,t'}(4\delta_{r,t'}-10)}\right)^{\delta_{r,t'}-1} \geq 1$$

Again using calculus, it can be seen that the left-hand side is monotonically non-decreasing for any $\delta_{r,t'} \geq 3$. 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 6. *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 $\sigma_{r,1,t} \leq \delta_{r,1}/7 - 1$, 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

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

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

$$\begin{aligned}
\frac{\delta_{r,1} - \sigma_{r,1,t}}{2\delta_{r,1}/5 - \sigma_{r,1,t}} \left(1 - \frac{1}{2\delta_{r,1}/5 - \sigma_{r,1,t}}\right)^{\delta_{r,1}-1-\sigma_{r,1,t}} &\geq \frac{1}{12} \\
\frac{5}{2} \left(1 - \frac{1}{2\delta_{r,1}/5 - \sigma_{r,1,t}}\right)^{\delta_{r,1}-1-\sigma_{r,1,t}} &\geq \frac{1}{12}, \text{ using Fact 1 }, \\
\frac{5}{2} \left(\exp\left(\frac{\delta_{r,1} - \sigma_{r,1,t} - 1}{2\delta_{r,1}/5 - \sigma_{r,1,t} - 1}\right)\right)^{-1} &\geq \frac{1}{12} \\
\frac{\delta_{r,1} - \sigma_{r,1,t} - 1}{2\delta_{r,1}/5 - \sigma_{r,1,t} - 1} &\leq \ln 30 \\
5\delta_{r,1} - 5\sigma_{r,1,t} - 5 &\leq 2\delta_{r,1} \ln 30 - 5\sigma_{r,1,t} \ln 30 - 5 \ln 30 \\
\delta_{r,1} &\geq \frac{5 \ln 30 - 5}{2 \ln 30 - 5} \sigma_{r,1,t} + \frac{5 \ln 30 - 5}{2 \ln 30 - 5} \\
\delta_{r,1} &\geq \frac{5 \ln 30 - 5}{2 \ln 30 - 5} (\sigma_{r,1,t} + 1)
\end{aligned}$$

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

The following lemma, shows the efficiency and correctness of the first part of the delegate algorithm, together with the slug algorithm for any neighborhood where the number of slugs with a message to transmit is more than $7(96 \log n + 1)$.

Lemma 7. *For any neighborhood $\mathcal{N}$ where the number of slug nodes that still hold a message to transmit is more than τ, after running the first part of the delegate algorithm together with the slug algorithm for $O(\Delta + \log^2 n)$ time steps, the number of slug nodes that still hold a message to transmit is at most τ with probability at least $1 - 1/n^{\psi_{(7)}}$, where $\psi_{(7)} > 1$.*

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 execution of the algorithm, either all nodes have achieved a successful transmission or such a round r exists. 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 6, the expected number of successful transmissions in the first $96 \log n$ communication steps is $E[Y_1] \geq 8 \log n$. By Lemma 5, the random variables $X_{r,i}$ are negatively 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 $\varepsilon = 7/8$,

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

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 $\sigma_{r,1,t} \leq \delta_{r,1}/7 - 1$ or the remaining slug 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 slugs in one neighborhood is bounded by Δ, 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

$$\begin{aligned}
Pr(E_1E_2E_3\ldots E_s) &= Pr(E_1)Pr(E_2|E_1)Pr(E_3|E_1E_2)\ldots Pr(E_s|E_1E_2E_3\ldots E_{s-1}) \\
&\geq \left(1-\frac{1}{n^3}\right)^{n/\log n}, \text{ using Fact 1 }, \\
&\geq e^{-n/(n^3-1)\log n} \\
&\geq 1-\frac{n}{(n^3-1)\log n} \\
&\geq 1-\frac{1}{n^{\psi_{(7)}}}, \text{ for some constant } \psi_{(7)} > 1.
\end{aligned}$$

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 8. *Running the local communication primitive, all delegate nodes receive the message of its slug nodes within $O(\Delta + \log^3 n)$ time steps with high probability.*

Proof. Consider any neighborhood $\mathcal{N}$. If the number of slug nodes in $\mathcal{N}$ that still have a message to transmit 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 4. If instead that number is more than $7(96 \log n + 1)$, as shown in Lemma 7, after $O(\Delta + \log^2 n)$ steps it is reduced to at most $7(96 \log n + 1)$ with probability at least $1 - 1/n^{\psi_{(7)}}$ for some $\psi_{(7)} > 1$, after which Lemma 4 applies. Given that there are at most n neighborhoods in the whole network, using the union bound the claim follows.

4.3 An Optimal Algorithm for Linear Memory

We describe now a gossiping algorithm for Sensor Networks for the case when each node memory can hold at least a linear number of messages. The algorithm has the following two phases.

- **Collection.** Every delegate node maintains a set of messages received, initially containing only its own message. Each slug node transmits its message to its delegate nodes. Every delegate node adds messages received from its slugs 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, slug 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 nodes start each phase synchronously. Later, we remove such assumption.

For the collection phase, we need to guarantee that all slug nodes transmit their message to their delegates. A straightforward application of the local communication primitive presented in Section 4.2 solves the problem in $O(\Delta + \log^3 n)$ time steps w.h.p. as shown in Lemma 8.

Lemma 9. *Any delegate node running the dissemination algorithm as described receives all messages from other delegate nodes within $O(D)$ time steps, 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 coverage, 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 [4] 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.

[4] 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.

Concurrency and Lack of Synchrony Nodes might wake up and become slugs 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 node wakes up. By Lemmas 2 and 3 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 executions of the primitive after that time step.

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.

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

Theorem 3. *Given a network of n nodes, after the last node wakes up, the gossiping problem can be solved distributedly with high probability in $\Theta(D + \Delta)$ time steps.*

Proof. Using Lemmas 2, 3, 8, and 9 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 $\Omega(n^{\psi_{(3)}})$, for some $\psi_{(3)} > 0$. Thus, the upper bound follows and, given the lower bound of Theorem 1, it is tight.

4.4 An Optimal Algorithm for Constant Memory

In this section we study the gossiping problem in Sensor Networks where the memory of each node is restricted to $\Theta(\log n)$ bits. I.e., nodes can not store in memory (and hence transmit in one slot) more than a constant number of pieces information, namely, node identifiers, messages, etc. As mentioned before, the extant literature for the gossiping problem does not include limits on memory size. However, the unlimited availability of such a resource in an unstructured setting such as a Sensor Network notably simplifies the problem.

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 [33] 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-slug hierarchy as defined in preprocessing (Section 4.1). Each communication step, i.e., the γ slots reserved for local use, include a slot for the delegate beacon, two slots for slug 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 slug 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, we remove such assumption.

Chain Definition The slugs of a delegate are ordered in a list, or *chain*, such that each of them knows its predecessor slug in the chain. Notice that each delegate node has only one chain but a node may be a slug of more than one delegate. Therefore, each slug node chooses one of its delegates arbitrarily to participate in its chain. To define the chain, each slug node requests to its chosen 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 slugs 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 slug in the chain. Delegate nodes only need to keep track of the last slug to grant new requests of appending. On the other hand, each slug node keeps track of its predecessor in the chain. If a slug 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 slug that first requested to be appended. The following lemma is an immediate consequence of Lemma 8.

Lemma 10. *Running the algorithm for chain definition, all slug nodes join a chain within* $O(\Delta + \log^3 n)$ *time steps with high probability.*

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 sink nodes. Without loss of generality, we assume that the sink node is a delegate (Otherwise, one of the sink delegates 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, such bookkeeping is feasible.

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
```

Lemma 11. *Any delegate node running the algorithm described 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. Since 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, the claim follows.

Shift Phase The intuition of this phase follows. Omit first the messages held by the slug nodes. In order to shift messages among delegate nodes, using the slot G of their communication steps, each delegate passes its message to a neighboring delegate according to the order given by a depth-first-search traversal of the tree. Now consider the messages held by slug nodes. Each delegate node d introduces the messages of the slugs in its chain in the dissemination by exchanging the message received from its parent delegate [5] with one of its slugs 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 slugs 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 slugs receive also that message and know which slug has exchanged messages with d in the current communication step. Given that each slug 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 12. *Every node running the algorithm described receives all 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 slug 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.

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, 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.

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

[5] If it is the root, use the last children.

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 slug)
                transmit exchange with radius αr
            case slot E reserved by d  (receive slug's message)
                receive ⟨d, s, message⟩
                message(d) ← message
            case slot F reserved by d  (acknowledge slug'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 slug s with chosen delegate d. message(s) is the message that s holds, predecessor(s,d) is the ID of the slug that precedes s in the chain of d. Transmissions use radius αr.

```
shift-slug(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
```

Theorem 4. *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 2, 3, 8, 11 and 12 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.

5 Conclusions and Open Problems

In this paper we have explored the problem of gossiping in Sensor Networks. We have given an optimal $\Theta(D + \Delta)$ algorithm when the memory of each sensor can hold the n messages. This result implies that the lower bound for broadcasting derived by Kushilevitz and Mansour [26] does not hold if preprocessing is allowed. We have also given an algorithm that solves gossiping in optimal $\Theta(n)$ time if each sensor can only hold a constant number of messages in its memory. Both algorithms use as a building block an algorithm that allows the successful transmission of Δ competing sensors to a distinguished node in $O(\Delta + polylog(n))$ time, with high probability. This algorithm only needs the availability of a collision detection mechanism and to know (a bound on) n. We believe this communication primitive can be useful to solve other problems in Radio Networks.

The most interesting question that this paper leaves open is a study of the trade-off between sensor memory size and gossiping time. Results in this paper answer the question for the extreme cases, namely, for linear and constant memory size in terms of number of messages. These cases lead to very different time complexities. An analysis for intermediate memory sizes is left for future work. Likewise, it would be interesting to explore models where node failures are neither arbitrary nor inexistent. Another interesting line is to reuse the infrastructure created as preprocessing in this paper to efficiently solve other problems. In fact, this has already been done to provide an early-terminating aggregation algorithm for Sensor Networks in [17].

References

1. R. Bar-Yehuda, O. Goldreich, and 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), pp. 104–126.
2. R. Bar-Yehuda, A. Israeli, and A. Itai, *Multiple communication in multihop radio networks*, SIAM Journal on Computing, 22 (1993), pp. 875–887.
3. D. Bruschi and M. D. Pinto, *Lower bounds for the broadcast problem in mobile radio networks*, Distributed Computing, 10 (1997), pp. 129–135.
4. I. Chlamtac and O. Weinstein, *The wave expansion approach to broadcasting in multihop networks*, in Proc. of INFOCOM, 1987.
5. B. Chlebus, L. Gąsieniec, A. Lingas, and 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.
6. B. S. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, and W. Rytter, *Deterministic broadcasting in ad hoc radio networks*, Distributed Computing, 15 (2002), pp. 27–38.
7. B. S. Chlebus, D. R. Kowalski, and T. Radzik, *Many-to-many communication in radio networks*, Algorithmica, 54 (2009), pp. 118–139.
8. B. S. Chlebus, D. R. Kowalski, and M. A. Rokicki, *Average time complexity of gossiping in radio networks*, in Proc. of the 13th Colloquium on Structural Information and Communication Complexity, 2006.
9. M. Chrobak, L. Gąsieniec, and W. Rytter, *Fast broadcasting and gossiping in radio networks*, in Proc. of the 41st IEEE Ann. Symp. on Foundations of Computer Science, 2000.
10. ——, *A randomized algorithm for gossiping in radio networks*, Networks, 43 (2004), pp. 119–124.
11. A. Clementi, A. Monti, and 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.
12. A. Czumaj and W. Rytter, *Broadcasting algorithms in radio networks with unknown topology*, in Proc. of the 44th IEEE Ann. Symp. on Foundations of Computer Science, 2003.
13. A. Dessmark and A. Pelc, *Deterministic broadcasting in ad hoc radio networks*, Journal of Discrete Algorithms, 5 (2007), pp. 187–201.
14. M. Farach-Colton, R. J. Fernandes, and M. A. Mosteiro, *Bootstrapping a hop-optimal network in the weak sensor model*, in Proc. of the 13th Ann. European Symp. on Algorithms, vol. 3669 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2005, pp. 827–838.

15. M. Farach-Colton and 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.
16. L. Fejes-Tóth, *über einen geometrischen satz*, Mathematische Zeitschrift, 46 (1940), pp. 83–85.
17. A. Fernández Anta, M. A. Mosteiro, and C. Thraves, *Brief announcement: An early-stopping protocol for computing aggregate functions in sensor networks*, in Proc. 22nd Ann. EATCS International Symposium on Distributed Computing, vol. 5218 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2008, pp. 504–506.
18. L. Gąsieniec, A. Pelc, D. Peleg, Y. Emek, E. Kantor, and C. Su, *Broadcasting in udg radio networks with unknown topology*, in Proc. 26th Ann. ACM Symp. on Principles of Distributed Computing, 2007.
19. L. Gąsieniec, D. Peleg, and Q. Xin, *Faster communication in known topology radio networks*, in Proc. 24th Ann. ACM Symp. on Principles of Distributed Computing, 2005.
20. L. Gąsieniec and 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.
21. M. Gerèb-Graus and T. Tsantilas, *Efficient optical communication in parallel computers*, in 4th Ann. ACM Symp. on Parallel Algorithms and Architectures, 1992, pp. 41–48.
22. A. Kesselman and D. R. Kowalski, *Fast distributed algorithm for convergecast in ad hoc geometric radio networks*, Journal of Parallel and Distributed Computing, 66 (2006), pp. 578–585.
23. D. R. Kowalski, *On selection problem in radio networks*, in Proc. 24th Ann. ACM Symp. on Principles of Distributed Computing, 2005, pp. 158–166.
24. D. R. Kowalski and A. Pelc, *Time complexity of radio broadcasting: Adaptiveness vs. obliviousness and randomization vs. determinism*, Theoretical Computer Science, 333 (2005), pp. 355–371.
25. F. Kuhn, T. Moscibroda, and 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.
26. E. Kushilevitz and Y. Mansour, *An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks*, SIAM Journal on Computing, 27 (1998), pp. 702–712.
27. P. Levis, E. Brewer, D. Culler, D. Gay, S. Madden, N. Patel, J. Polastre, S. Shenker, R. Szewczyk, and A. Woo, *The emergence of a networking primitive in wireless sensor networks*, Commun. ACM, 51 (2008), pp. 99–106.
28. D. Liu and M. Prabhakaran, *On randomized broadcasting and gossiping in radio networks*, Lecture Notes in Computer Science, 2387 (2002), pp. 340–349.
29. C. U. Martel, *Maximum finding on a multiple access broadcast network*, Inf. Process. Lett., 52 (1994), pp. 7–13.
30. D. S. Mitrinović, *Elementary Inequalities*, P. Noordhoff Ltd. - Groningen, 1964.
31. T. Moscibroda and R. Wattenhofer, *Maximal independent sets in radio networks*, in Proc. 24th Ann. ACM Symp. on Principles of Distributed Computing, 2005, pp. 148–157.
32. V. Ravelomanana, *Optimal initialization and gossiping algorithms for random radio networks*, IEEE Trans. Parallel Distr. Syst., 18 (2007), pp. 17–28.
33. R. E. Tarjan and U. Vishkin, *An efficient parallel biconnectivity algorithm*, SIAM Journal on Computing, 14 (1985), pp. 862–874.

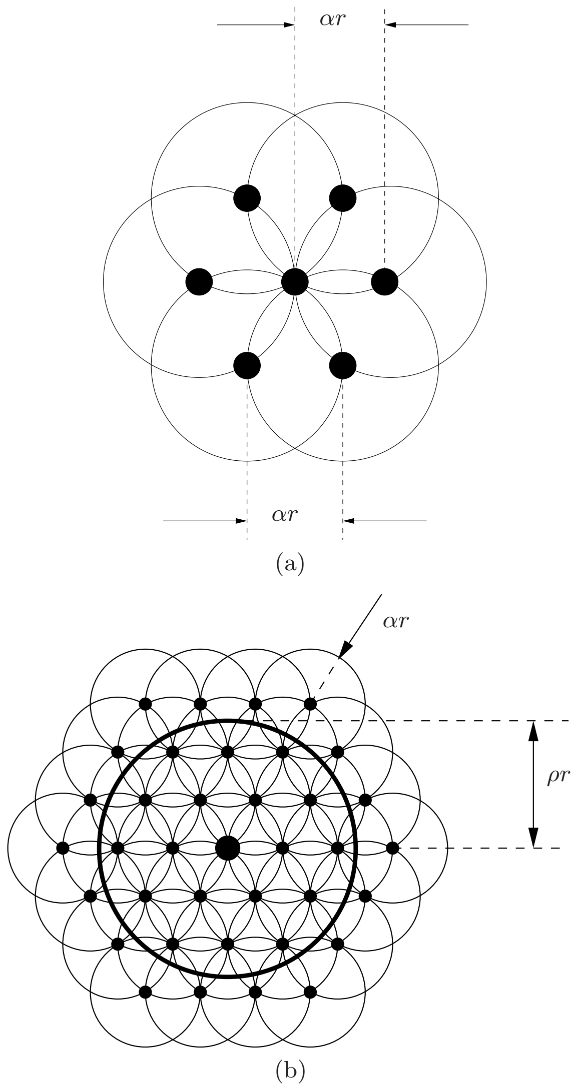


Fig. 1. Illustration of maximum degree.

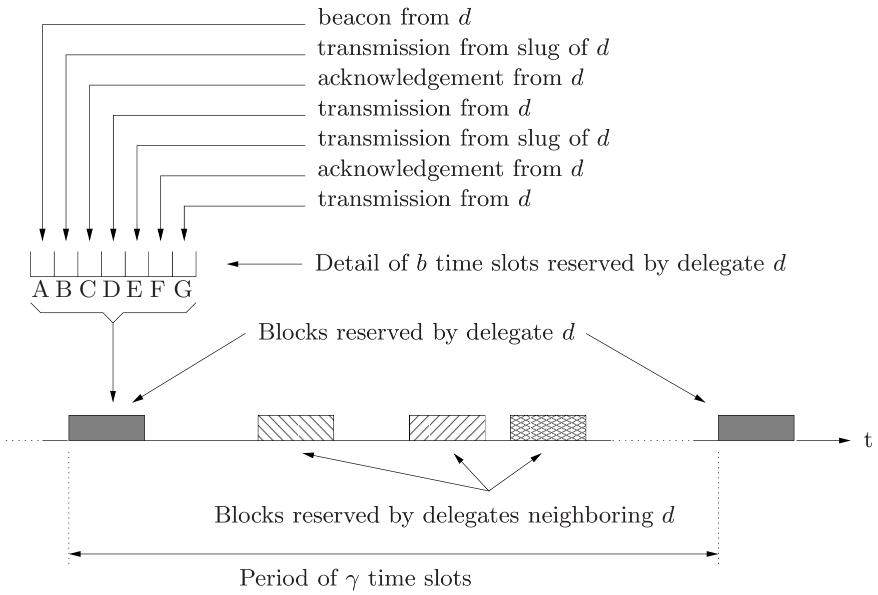


Fig. 2. Illustration of time slots usage by delegate node d and some of its neighboring nodes. The algorithm for unrestricted memory uses only slots A to D, whereas the algorithm for restricted memory uses all slots.

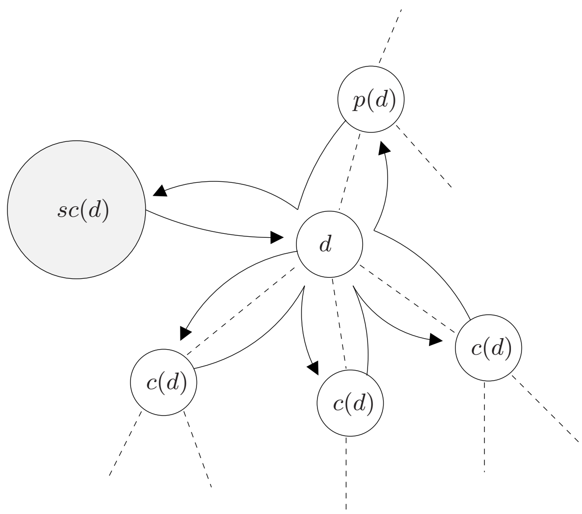


Fig. 3. Illustration of Algorithm 6. d: internal node, $p(d)$: parent of d, $c(d)$: child of d, $sc(d)$: slug chain of d. The arrows indicate the flow of each message.