Sensor Network Gossiping
or
How to Break the Broadcast Lower Bound

Martín Farach-Colton and Miguel A. Mosteiro

Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA
`{farach,mosteiro}@cs.rutgers.edu`

Abstract. Gossiping is an important problem in Radio Networks that has been well studied, leading to many important results. Due to strong resouce 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 reduced, optimal running time of $O(D + \Delta)$ for an algorithm that completes gossiping with high probability in a Sensor Network of unknown topology and adversarial wake-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, our result proves that the classic lower bound of [16] can be broken if nodes are allowed to do preprocessing.

1 Introduction

The *Radio Network* is a simplified abstraction of a radio-communication network. The question of how to diseminate information within such networks has led to different well-studied problems. Those problems differ on the number of network nodes holding messages to transmit, the number of different messages to be transmitted and the number 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 in the number of nodes that hold those messages, as follows. When k arbitrary nodes have a message the problem is known as *k-selection* [14]. If $k = 1$ the problem is called *Broadcast* [1,16], and if $k = n$, the size of the network, it is called *Gossiping* [5,17].

We study the gossiping problem in Sensor Networks, 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

---

This research was supported in part by NSF grants CCF0621425, CCF 05414009, CCF 0632838, and DIMACS, Center for Discrete Mathematics & Theoretical Computer Science, grant numbered NSF CCR 00-87022.

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. At least one of the following two crucial assumptions is present in all of these results. It is frequently assumed that the size of a message transmitted in one step 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. In addition, a usual assumption is that either nodes start simultaneously or a global clock is available. In the Weak Sensor Model [10], a harsh and comprehensive model that summarizes the literature on sensor node restrictions, none of these assumptions are feasible because memory is limited, wake-up schedule is adversarial and no global clock is available.

Although in the asymptotic analysis $n \to \infty$, 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 is produced in hostile or remote large areas, a global clock or a synchronous start is too strong of an assumption. Thus, we study gossiping in a relaxed version of the Weak Sensor Model where memory size is bounded only by a linear number of messages. We leave open the question of how to improve these results for constant-bounded memory size.

Related Work. Bar-Yehuda, Israeli and Itai [2] presented a randomized algorithm for Radio Networks with a topology modeled by an undirected graph, or *symmetric networks*, that completes gossiping in $O(n \log^2 n)$[1] on average. Briefly, their technique, used previously in [4] and later re-utilized in [12], is to build an underlying BFS tree to first collect all messages in the root node and later diseminate 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 $O(\log n)$ bit messages in synchronous time slots. However, they can store as many as needed. The same bound but with high probability[2] was proved in [5]. The algorithm relies on unbounded message size and global synchronism.

For directed graph topologies or *asymmetric networks*, Chrobak, Gąsieniec and Rytter [7] 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.

---

[1] Througout this paper, log means $\log_2$ unless otherwise stated.

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

Using the same protocol, but improving the limited broadcast by adding randomization to it, Liu and Prabhakaran [17] reduced that upper bound by a logarithmic factor. More recently, Czumaj and Rytter [9] obtained a bound of $O(n \log^2 n)$ w.h.p. for this protocol 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.

Recently, Ravelomanana [20] studied the gossiping problem 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, but the lower bound used to prove it is the well-known result of Kushilevitz and Mansour [16] where nodes are not allowed to do anything before receiving the broadcast message.

Regarding deterministic solutions, Chrobak, Gąsieniec and Rytter [6] presented a $O(n^{3/2} \log^2 n)$ algorithm for asymmetric networks. The protocol makes use of selecting sequences to ensure non-colliding transmissions. In this model nodes have different ID's in $\{1, \ldots, n\}$, the topology is modeled by a directed graph, memory and messages are unbounded and global synchronism is assumed.

Results for the broadcast problem can be used as lower bounds for gossiping because the former can be solved using an algorithm for the latter. However, it should be noticed that, in order to prove lower bounds, broadcast protocols are defined leaving out solutions that include a pre-processing stage [16]. Bruschi and Del Pinto [3] proved a $\Omega(D \log n)$ lower bound, in a model where all nodes start simultaneously and nodes know their message history. More recently, Clementi, Monti and Silvestri [8] improved the lower bound to $\Omega(n \log D)$ in symmetric networks even if nodes are not synchronized. Kowalski and Pelc [15] constructed a class of graphs of diameter 4, such that every broadcasting algorithm requires time $\Omega(n^{1/4})$ on one of these graphs. The best general lower bound for randomized protocols is $\Omega(D \log(n/D))$ obtained by Kushilevitz and Mansour [16].

As for lower bounds for gossiping, Chlebus, Gąsieniec, Lingas and Pangourtzis [5] proved that any deterministic oblivious gossiping algorithm requires at least $n^2/2 - n/2 + 1$ steps to complete. In the same paper, for the important class of fair randomized protocols, i.e., protocols where all nodes use the same probability of transmission in the same time step, it was proved that for any integer $n \leq q \leq n^2/2$ there exists an asymmetric network such that the expected time to complete gossiping is $\Omega(q)$. More recently, Gąsieniec and Potapov [12] showed lower bounds of $\Omega(n^2)$ for asymmetric networks and $\Omega(n \log n)$ for symmetric networks. The topology of the construction used for the later can not be embeded in geometric graphs, therefore does not apply to Sensor Networks.

Our Results. We study the gossiping problem in a relaxed version of the Weak Sensor Model where memory size is bounded only by a linear number of messages. We present a randomized algorithm that, given a network of n nodes, with high probability completes gossiping in $O(\Delta + D)$ time steps after the last node starts running the algorithm. Given that $\Omega(D)$ and $\Omega(\Delta)$ are lower bounds for this problem, this 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. Our result also shows that the classical broadcast lower bound of Kushilevitz and Mansour [16] can be broken if nodes are allowed to do some preprocessing before receiving a message to transmit. In that paper, the lower bound is proved using a layered structure, where a crucial assumption is that, in each layer, all nodes run the same uniform protocol upon receiving the message to be broadcasted. By the definition of a broadcast protocol given in that paper "any other processor is inactive until receiving a message for the first time". If preprocessing is allowed the protocol may be non-uniform as in our protocol.

Roadmap. In the remainder of this paper we define the models used througout in Section 2 and we present the details of our algorithm in Section 3.

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 at random 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 $\mathbb{R}^2$, 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. We also assume that the topology is unknown to the nodes and the only knowledge each node has is the number of nodes in the whole network n, its unique identifier in $\{1, \ldots, n\}$, and a constant parameter β to be defined later.

In addition to topology and connectivity models, an appropriate model of the constraints of the nodes in the network has to be defined in order to properly design and analyze protocols. Bar-Yehuda, Goldreich and Itai [1] used a formal model of a radio network that specifies many of those restrictions, including limits on contention resolution, but they make no mention of computational limits such as small memory. We use a relaxed version of the comprehensive Weak Sensor Model [10] where memory size is bounded only by $O(nm)$ where m is the message size. Briefly, the following assumptions are included in this model. The communication among neighboring nodes is through broadcast on a shared channel, where a node receives a message only if exactly one of its neighbors transmits. If more than one message is sent at the same time, a collision occurs and no collision detection mechanism is available. Sensors nodes cannot receive and transmit in the same time slot. The channel is assumed to have only two states: transmission and silence/collision. Time is assumed to be slotted and all nodes have the same clock frequency, but no global synchronizing mechanism is available. Furthermore, they *wake-up* adversarially. We assume that sensor nodes can adjust their power of transmission but only to a constant number of levels. Other limitations include: limited life cycle, short transmission range, only one channel of communication, no position information, and unreliability.

3 An Optimal Algorithm

We describe now the gossiping protocol for Sensor Networks. Although global synchronism is not required, for the sake of clarity, we assume first that nodes start simultaneously and analyze each phase separately. Later we show that such an assumption is not necessary. The protocol has the following four main phases.

1. Define nodes as *master nodes* in such a way that every node is within distance at most ar of some master, and all master nodes are separated by a distance at least ar, where r is the maximum range of transmission and a is a constant such that $0 < a < 1/3$. All non-master nodes are called *slave nodes.* Notice that any node can be the slave of at most 6 masters.
2. Every master node reserves blocks of time steps for local use, so that each master and its slaves can communicate without colliding with transmissions from any nodes within radius r.
3. Every master node maintains a set of messages received, initially containing only its own message. Using the reserved blocks, all slave nodes transmit their message to their master nodes transmitting with radius ar. Every master node adds messages received from its slaves to its set.
4. Using the reserved blocks, every master node deterministically transmits its set of messages to all master nodes within radius at most r and repeatedly adds the messages received from other masters and re-transmits.

The choice of the upper bound on a guarantees that communication between master and slave nodes is achieved in a time slot that is not used by any neighboring master-slave pair. Given that a is a constant, its effect is folded in the other constants of the analysis. More precisely, as we will see in Section 3.2, the more master nodes that are included in a circle of radius r the bigger is the block of reserved time slots. Although the size of such a block is still a constant, for constant-sensitive applications a must be made as big as possible. We now give the details of each phase and the analysis.

3.1 Phase 1

This phase of the protocol can be implemented distributedly running a Maximal Independent Set (MIS) algorithm with radius ar. For that purpose we use the algorithm presented in [19] which works in two stages. 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 [10]. We omit the details here for the sake of brevity.

Lemma 1. *Under the restrictions of the Weak Sensor Model, for a given node running the algorithm described above, at least one node within its transmission range joins the MIS in* $O(\log^2 n)$ *time steps and no two MIS nodes are within range of each other with probability* $1 - 1/n^{\gamma_1}$ *for some constant* $\gamma_1 > 0$.

Proof. As in [19] and [10].

□

3.2 Phase 2

We implement this phase using a counter to break symmetry as in the previous algorithm. The main idea is for each master node to reserve certain steps for deterministic transmissions in a way that there are no collisions.

Algorithm 1: Algorithm of Phase 2. α_1, α_2, α_3, α_4, α_5, b and β are constants.

```
for each master node do
    set a step counter to 0.
    while true do
        if current step was not reserved then
            transmit the counter and ID with probability $1/\alpha_1$ and radius $r$
            using non-reserved slots.
            if not transmitting in the current time slot then
                if a neighbor's counter is received and the absolute difference
                between the local and neighbor's counter is $\leq \alpha_2 \log n$ then
                    set local counter to 0.
                else
                    if a neighbor's reservation message is received then
                        keep track of slots reserved.
            increase counter if transmitted at least once.
            if the counter reached $\lceil \alpha_3 \log n \rceil$ then
                choose a block of $b$ contiguous available time slots in an interval
                of $\beta$.
                for $\alpha_5 \log n$ available steps do
                    transmit ID and the incoming time slots reserved with
                    probability $1/\alpha_4$ and radius $r$ using non-reserved slots.
                while true do
                    transmit a beacon message in the reserved slot with radius
                    $ar$.
```

The protocol, detailed in Algorithm 1, works as follows. α_1, α_2, α_3, α_4, α_5, b and β are constants. Each master node x maintains a step counter, initially set to 0. In each step still not reserved by any of the master nodes within distance r, x transmits its counter and its identity with probability $1/\alpha_1$ within a radius of r. In each step that x does not transmit, it is in receiver 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 β.

Next, x informs to the neighboring master nodes which are the slots chosen. In order to do that, x transmits a message containing the number of steps after the current step 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. Of course, the number of steps is updated appropriately in each step. As in [10], master nodes within distance of r from this node are guaranteed to receive this message within $O(\log n)$ steps and no neighboring master node can reach its final count before that w.h.p.

After $\alpha_5 \log n$ steps, the master node synchronizes its slaves by repeatedly transmitting a *beacon message*. After the first beacon message slave nodes move to phase 3.

The block of b reserved slots is big enough to include one for slave transmissions, one for master acknowledgements to slave transmissions, one for beacon messages, and one for transmissions among master nodes in the last phase. The period β is a constant big enough to ensure that each master node gets to reserve some block. As we show in Lemma 2, the number of master nodes in any circle of radius r is bounded by $O(1)$. Thus, such a constant value β exists.

Lemma 2. *There are at most* $3\lceil 2/a\sqrt{3}\rceil(\lceil 2/a\sqrt{3}\rceil+1)$ *master nodes within distance* r *of any master node with high probability.*

Proof. All master nodes are separated by a distance of at least ar with high probability as a result of phase 1. Consider the smallest regular hexagon whose side is a multiple of ar and covers completely a circle of radius r. Consider a tiling of such hexagon with equilateral triangles of side ar. As proved by Fejes-Tóth in 1940 [11], 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 master nodes at a distance r of a master node located in the center of such a hexagon. That number is $3\lceil 2/a\sqrt{3}\rceil(\lceil 2/a\sqrt{3}\rceil + 1)$.

□

Lemma 3. *After* $O(\log n)$ *time steps running the algorithm described above, any master node reserves a block of* $b \in O(1)$ *steps every* $\beta \in O(1)$ *steps for local use, i.e., this block does not overlap with the block of any other master node separated by a distance at most* r*, with probability* $1 - 1/n^{\gamma_2}$ *for some constant* $\gamma_2 > 0$.

Proof. The running time of the algorithm can be proved as in [10].

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 master 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 [10] it can be shown that the probability of that event is $O(1/n^{\gamma_3})$ for some constant $\gamma_3 > 2$, which ensures that the probability of failure over all possible pairs is low.

Case 2: we assume for the sake of contradiction that a master node x can not reserve a contiguous block of b slots. This implies that after some neighboring master nodes reserve their blocks, there are no contiguous b slots available. As proved in Lemma 2 there are at most $3\lceil 2/a\sqrt{3}\rceil(\lceil 2/a\sqrt{3}\rceil + 1)$ master nodes within r distance of any master node w.h.p. But, making $\beta \geq (2b-1)(1 + 3\lceil 2/a\sqrt{3}\rceil(\lceil 2/a\sqrt{3}\rceil + 1))$ there is always a block of b contiguous slots available w.h.p.

□

3.3 Phase 3

In this phase we need to guarantee that all slave nodes transmit their message to their master. A simple randomized algorithm can achieve this task in $O(\Delta \log n)$ but we show in this section that it can be done faster using the synchronism achieved in the previous phase.

In order to implement this phase efficiently, we could use an approach similar to the algorithm presented in [13]. 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 protocol requires that nodes know h, in our problem Δ. So, instead, we use an scheme where the only topology information is the size of the whole network n.

As explained before, slave nodes periodically receive a beacon message from their master node indicating the forecoming available slots for local use. A block of reserved slots includes, among others, a slot for slave transmissions and a slot for master acknowledgement. This acknowledgement informs a node that its transmission was successful, implementing a collision detection mechanism. Thus, we can take advantage of local synchronism achieved by the beacon message and collision detection implemented by the acknowledgement. For the sake of clarity, we focus here in the description of the algorithm ignoring these details and the fact that nodes use only the reserved slots for transmissions.

The protocol, detailed in Algorithm 2, works as follows. The algorithm is window-based, i.e., nodes repeatedly choose uniformly one time slot within an interval, or *window*, of time slots to transmit its message. Regarding the size of such a window, the protocol follows a back-on/back-off strategy, i.e., the window is increased in an outer loop by the master and decreased in an inner loop by the slaves. The master informs the slaves of the current window size in the beacon message. In order to succeed with high probability when $o(\log n)$ messages are left, $\Theta(\log^2 n)$ steps where nodes repeatedly transmit with probability $1/\log n$ are included at the end of each phase of the outer loop.

Algorithm 2: Algorithm of Phase 3. α is a positive constant.

```
for $i = \{\lfloor \log\log n \rfloor, \lfloor \log\log n \rfloor + 1, \lfloor \log\log n \rfloor + 2, \ldots\}$ do
    Each master node transmits $i$ in the beacon message.
    Each slave node does the following.
    for $j = 0$ to $i - 1$ do
        Choose uniformly a step within the next $2^{i-j}$ steps.
        Transmit the message in such a step and receive messages in all the other steps.
    for $\alpha \log^2 n$ steps do
        Transmit the message with probability $1/\log n$.
        Receive messages if not transmitting.
```

The intuition for the algorithm is as follows. Assume nodes know the number of nodes in their one-hop neighborhood, call it δ. Then, we think of the problem as a random process where δ balls (modelling the messages) are dropped uniformly in δ bins (modelling time slots). We will show that, for large enough δ, with high probability a constant fraction of the balls fall alone in a bin. Now, we can repeat the process removing this constant fraction of balls and bins until all balls have fallen alone. Since nodes do not know the size of their neighborhood, the outer loop increasing the number of bins is necessary.

We now concentrate in the analysis of this phase. First, we need the following lemma about bins and balls.

Lemma 4. *For $\delta \geq (2e/(1-e\epsilon)^2)(1+(\gamma_4+1/2)\ln n)$, if δ balls are dropped in δ bins uniformly at random, the probability that the number of bins with exactly one ball is smaller than $\epsilon\delta$ is at most $1/n^{\gamma_4}$ for some constants $\gamma_4 > 0$ and $0 < \epsilon < 1$.*

Proof. The probability for a given ball to fall in a given bin is $(1/\delta)(1-1/\delta)^{\delta-1} \geq 1/e\delta$. Let X_i be a random variable that indicates if there is exactly one ball in bin i. Then, $Pr(X_i = 1) \geq 1/e$. To handle the dependencies that arise in balls and bins problems, we approximate the joint distribution of the number of balls in all bins by assuming the load in each bin is an independent Poisson random variable with mean 1. Let X be a random variable that indicates the total number of bins with exactly one ball. Then, $\mu = E[X] = \delta/e$. Using Chernoff-Hoeffding bounds,

$$Pr(X \leq \epsilon\delta) \leq exp\left(-\frac{\delta}{2e}\left(1-e\epsilon\right)^2\right).$$

As shown in [18], any event that takes place with probability p in the Poisson case takes place with probability at most $pe\sqrt{\delta}$ in the exact case. Then, we want to show

$$exp\left(-\frac{\delta}{2e}(1-e\epsilon)^2\right)e\sqrt{\delta} \leq n^{-\gamma}.$$

Which is true for

$$\delta \geq \frac{2e}{(1-e\epsilon)^2}\left(1+\left(\frac{1}{2}+\gamma\right)\ln n\right).$$

□

Lemma 5. *The algorithm described above guarantees that a master node receives the message of all its slaves within $O(\Delta + \log^2 n \log \Delta)$ steps with probability $1 - 1/n^{\gamma_5}$ for some constant $\gamma_5 > 0$.*

Proof. If $\delta \in O(\log n)$, as shown in [10], all slave nodes achieve a non-colliding transmission within $O(\log^2 n)$ steps with probability at least $1 - n^{-\gamma_i}$ for some constant $\gamma_i > 0$.

If $\delta \in \omega(\log n)$, after the master node transmits a window size in $\Theta(\delta)$, a constant fraction ϵ of slave nodes succeed in each step j of the inner loop with probability $1 - n^{-\gamma_j}$ as shown in Lemma 4.

Taking each γ_j small enough and ϵ big enough and telescoping the running time of each loop, the claim follows.

□

3.4 Phase 4

Each master node maintains a set of messages, initially containing only its own message, adding all messages received in phases 3 and 4, and deterministically re-transmiting this set in the time slots reserved for that purpose.

Lemma 6. *Any master node running the algorithm of phase 4 receives all messages from other master nodes within $O(D)$ time steps, where D is the diameter of the network.*

Proof. Given that the master nodes form a maximal independent set, the diameter of the subgraph induced by them is in $O(D)$. Since master nodes re-transmit all messages ever received deterministically every $\beta \in O(1)$ steps, the claim follows.

□

3.5 Overall Analysis

Two important restrictions of the Weak Sensor Model are that nodes start running the algorithm, or wake-up for short, according to an adversarial schedule, and that their power supply is unrealiable resulting in potential on/off cycles. In this section, we remove the assumption of simultaneous wake-up used in the analysis and we show that in fact the algorithm and its efficiency are still the same. Given that in order to solve the gossiping problem all nodes have to be active, we analyze the time after the last node starts running the algorithm and we assume that no node turns off before completion. Otherwise, any time analysis would be meaningless in presence of an adversary that turns on and off nodes forever.

The MIS algorithm used in phase 1 includes an initial waiting period, long enough to ensure that nodes waking-up do not interfere with nodes already running the algorithm. We extend this waiting period to the duration of the first two phases of the protocol. If during the waiting period a node becomes a slave of some master node, the slave waits for the beacon message doing nothing and goes directly to the third phase after receiving it. If a node does not become a slave during its waiting period, it starts phase 1 using slots still available, i.e., slots that were not reserved by some master node within radius r. Choosing $\beta \in O(1)$ big enough, there is always some slot available every β slots. Choosing the constant factors of the probabilities appropriately, nodes running phases 1 and 2 do not interfere with each other w.h.p. as proved in [10].

Nodes in phase 3 are synchronized by their master beacon-message. However, if a node is woken up late enough with respect to its slave neighbors, it could reach this phase after the window size of the outer loop is in $\omega(\Delta)$. In order to avoid this situation, whenever the master node does not receive transmissions after receiving transmissions for a given window size, it resets the outer loop. Given that the running time is analyzed after the last node is woken up the claimed running time still holds.

Given that nodes running phases 1 and 2 use all slots not reserved there is also no conflict with nodes running phases 3 and 4. The last two phases are deterministic and utilize time multiplexing, synchronized by the beacon message. Thus, there is also no conflict among nodes in these phases.

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

Theorem 1. *Given a network of n nodes, after the last node starts running the algorithm described in this section, the gossiping problem is solved with high probability in $O(D + \Delta)$.*

Proof. Using Lemmas 1, 3, 5, and 6 the overall complexity of the algorithm including preprocessing is $O(\log^2 n + \log n + \Delta + \log^2 n \log \Delta + D)$ 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, D and Δ can not be simultanoeusly in $o(n^c)$ for any constant $c > 0$ and the claim follows.

□

Given that $\Omega(D)$ and $\Omega(\Delta)$ are lower bounds for this problem, the previous result is tight.

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*, 45:104–126, 1992.
2. R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple communication in multihop radio networks. *SIAM Journal on Computing*, 22:875–887, 1993.

3. D. Bruschi and M. D. Pinto. Lower bounds for the broadcast problem in mobile radio networks. *Distributed Computing*, 10(3):129–135, 1997.
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 International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 44–51, 2001.
6. M. Chrobak, L. Gąsieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. In *Proc. of the 41st IEEE Annual Symp. on Foundation of Computer Science*, 2000.
7. M. Chrobak, L. Gąsieniec, and W. Rytter. A randomized algorithm for gossiping in radio networks. *Networks*, 43(2):119–124, 2004.
8. A. Clementi, A. Monti, and R. Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In *Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001.
9. A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *Proc. of the 44th IEEE Annual Symp. on Foundation of Computer Science*, 2003.
10. 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 Annual European Symposium on Algorithms*, volume 3669 of *Lecture Notes in Computer Science*, pages 827–838. Springer-Verlag, Berlin, 2005.
11. L. Fejes-Tóth. über einen geometrischen satz. *Mathematische Zeitschrift*, 46(1):83–85, 1940.
12. L. Gąsieniec and I. Potapov. Gossiping with unit messages in known radio networks. In *Proc. of 17th IFIP World Computer Congress / 2nd IFIP International Conference on Theoretical Computer Science*, 2002.
13. M. Gerèb-Graus and T. Tsantilas. Efficient optical communication in parallel computers. In *4th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 41–48, 1992.
14. D. R. Kowalski. On selection problem in radio networks. In *Proceedings 24th Annual ACM Symposium on Principles of Distributed Computing*, pages 158–166, 2005.
15. D. R. Kowalski and A. Pelc. Time of deterministic broadcasting in radio networks with local knowledge. *SIAM Journal on Computing*, 33(4):870–891, 2004.
16. E. Kushilevitz and Y. Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM Journal on Computing*, 27(3):702–712, 1998.
17. D. Liu and M. Prabhakaran. On randomized broadcasting and gossiping in radio networks. *Lecture Notes in Computer Science*, 2387:340–349, 2002.
18. M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.
19. T. Moscibroda and R. Wattenhofer. Maximal independent sets in radio networks. Technical Report TR-478, Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland, December 2004.
20. V. Ravelomanana. Optimal initialization and gossiping algorithms for random radio networks. *IEEE Trans. Parallel Distr. Syst.*, 18(1):17–28, 2007.