Deterministic Recurrent Communication in Restricted Sensor Networks

Antonio Fernández Anta[a], Miguel A. Mosteiro[b,c], Christopher Thraves[c,1]

[a] *Institute IMDEA Networks, Leganés, Spain*
[b] *Department of Computer Science, Rutgers University, Piscataway, NJ, USA*
[c] *LADyR, GSyC, Universidad Rey Juan Carlos, Móstoles, Madrid, Spain*

Abstract

In Sensor Networks, the lack of topology information and the availability of only one communication channel has led research work to the use of randomization to deal with collisions of transmissions. However, the scarcest resource in this setting is the energy supply, and radio communication dominates the sensor node energy consumption. Hence, redundant trials of transmission as used in randomized protocols may be counter-effective. Additionally, most of the research work in Sensor Networks is either heuristic or includes unrealistic assumptions. Hence, provable results for many basic problems still remain to be given. In this paper, we study upper and lower bounds for deterministic communication primitives under the harsh constraints of sensor nodes.

Keywords: Radio Networks, Sensor Networks, deterministic communication, recurrent communication

---

☆A preliminary version of this work has appeared in [16].

☆☆This research was supported in part by the National Science Foundation (CCF-0937829, CCF-1114930), Comunidad de Madrid grant S2009TIC-1692, Spanish MICINN grant TIN2008–06735-C02-01, French ANR project Shaman, and Spanish MICINN grant Juan de la Cierva.

Email addresses: `antonio.fernandez@imdea.org` (Antonio Fernández Anta), `mosteiro@cs.rutgers.edu` (Miguel A. Mosteiro), `cbthraves@gsyc.es` (Christopher Thraves)

[1] Partially done while working as postdoc researcher at ASAP research team, INRIA Rennes – Bretagne Atlantique, France.

1. Introduction

A Sensor Network is a well-studied simplified abstraction of a radio-communication network where nodes are deployed at random over a large area in order to monitor some physical event. Sensor Networks is a very active research area, not only due to the potential applications of such a technology, but also because well-known techniques used in networks cannot be straightforwardly implemented in sensor nodes, due to harsh resource limitations.

Sensor Networks are expected to be used in remote or hostile environments. Hence, random deployment of nodes is frequently assumed. Although the density of nodes must be big enough to achieve connectivity, precise location of specific nodes cannot be guaranteed in such scenario. Consequently, the topology of the network is usually assumed to be unknown, except perhaps for bounds on the total number of nodes and the maximum number of neighbors of any node. In addition, given that in Sensor Networks only one channel of communication is assumed to be available, protocols must deal with collision of transmissions.

Most of the Sensor Network protocols use randomness to deal with collisions and lack of topology information. Randomized protocols are fast and resilient to failures, but frequently rely on redundant transmissions. Given that the most restrictive resource in a Sensor Network is energy and that the dominating factor in energy consumption is the radio communication, deterministic algorithms may yield energy-efficient solutions. In this paper, deterministic communication primitives are studied under the harsh restrictions of sensor nodes.

1.1. Model

In this work, we assume a Sensor Network where a total of n nodes are deployed over an area of interest. We model the potential connectivity of nodes as a *Geometric Graph* where n nodes are deployed in $\mathbb{R}^2$, and a pair of nodes is connected by an undirected edge if and only if they are at an Euclidean distance of at most a parameter r. It is important to stress that this topology models the *potential* connectivity of nodes. However, upon deployment, two neighboring nodes still have to establish a communication link in order to be neighbors in terms of the communication network. The geometric graph model implies a circular-range assumption, which in practice may not be true. However, whenever this is the case, the minimum radius may be taken without extra asymptotic cost.

As customary in Sensor Networks, nodes are assumed to be deployed densely enough to guarantee connectivity and coverage. For adaptive protocols, we assume that 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, which introduces only a constant factor overhead.

Although random, the deployment of sensor nodes is not the result of an uncontrolled experiment where any outcome has a positive probability. Hence, we assume that the maximum *degree*, i.e., the maximum number of nodes located within a radius of r of any node, is a known value $k - 1 < n$. Each node knows only the total size of the network n, its unique identifier in $\{1, \ldots, n\}$ and the maximum degree $k - 1$. In order to specify the results obtained to the level of constants, we further assume that $k \geq 6$. Were not this the case, the same asymptotic results can be proved using the prime number theorem [17].

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. As a general framework, we use the Weak Sensor Model, elucidated in [12], including the assumptions described below for completeness. For adaptive protocols, memory size limitations and/or adversarial node-activation schedule will be relaxed to improve in time efficiency.

- 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. Indistinctively, we say that nodes *wake-up*, *start-up*, or are *activated* adversarially.

- 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. Thus, the channel is assumed to have only two states: transmission and silence/collision.

- 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. Furthermore, we assume that only two levels of power of transmission are available, the maximum power resulting in a radius r of communication, and a reduced power of transmission that results in the biggest feasible radius smaller or equal than $r/2$, the precise value depending on the physical constraints. In the rest of the paper, we will assume a precise value of $r/2$ for simplicity.

- The memory size of each sensor node is bounded by $O(1)$ words of $O(\log n)^2$ bits, unless otherwise stated.

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

- No position information or distance estimation capabilities are available.

The Weak Sensor Model includes also limits on life cycle due to energy constraints and reliability. We specify how do we model these restrictions in this work after defining the problem and the efficiency metrics studied in Section 1.2.

In order to highlight the relevance of this work, we compare our model with previous models of node constraints. Unless otherwise stated, we model node restrictions as in the Weak Sensor Model [12]. Bar-Yehuda et al. [2] used a formal model of Radio Network, which additionally includes topology assumptions, that 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 of Kuhn et al. [21]. 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 [23].

In a time slot, an active node can be in one of two states, namely *transmission* or *reception*. We denote a temporal sequence of states of a node as a *schedule of transmissions*, or simply a *schedule* when the context is clear.

1.2. Problem Definition

An expected application of Sensor Networks is to continuously monitor some physical phenomena. Hence, in this paper, the problem we address is to guarantee that each active node can communicate with all of its neighboring active nodes infinitely many times. The actual use of such a capability will depend of course on the availability of *application messages* to be delivered. Our goal is to give guarantees on the energy cost and the time delay of the communication only, leaving aside the overhead due to queuing or other factors.

In Radio Networks, messages are successfully delivered by means of non-colliding transmissions. Non-colliding transmissions in single-hop Radio Networks are clearly defined: the number of transmitters must be exactly one. However, in a multi-hop scenario such as Sensor Networks the same transmission may be correctly received by some nodes and collide with other transmissions at other nodes. Thus, a more precise definition is necessary. If in a given time slot exactly one of the adjacent neighbors of a node x transmits, and x itself is receiving, we say that there was a *clear reception* at x in that time slot. Whereas, in the case where a node transmits a message in a given time slot, and no other node within *two* hops of the transmitter transmits in the same time slot, we say that there was a *clear transmission*. Notice that when a clear transmission is produced by a node, all its neighbors clearly receive at the same time. Of course, in a single-hop network both problems are identical.

In this paper, our goal is to guarantee that each node communicates with all of its at most $k-1$ neighbors. Hence, a closely-related communication primitive known as *selection* is relevant for our purposes. In the selection problem, each of k *active* nodes of a single-hop Radio Network hold a different message that has to be delivered to all the active nodes. Once its message is successfully transmitted, a node becomes inactive. Given that we want to guarantee communication forever, in this paper, we give upper and lower bounds for generalizations of the selection problem that we define as follows.

Definition 1. Given a single-hop Radio Network of n nodes where k of them are activated possibly at different times, in order to solve the *Recurring Selection* problem every active node must clearly transmit infinitely many times.

For multihop networks, based on the distinction between clear reception and transmission, we define the following two problems.

Definition 2. Given a Sensor Network of n nodes and maximum degree $k-1$, where upon activation, possibly at different times, nodes stay active forever, in order to solve the *Recurring Reception* problem every active node must clearly receive from all of its active neighboring nodes infinitely many times.

Definition 3. Given a Sensor Network of n nodes and maximum degree $k-1$, where upon activation, possibly at different times, nodes stay active forever, in order to solve the *Recurring Transmission* problem every active node must clearly transmit to all of its active neighboring nodes infinitely many times.

Given that protocols for such problems run forever, we need to establish a metric to evaluate energy cost and time efficiency. Let $R_u^i(v)$, $i > 1$, be the number of transmissions of u between the $(i-1)^{th}$ and the i^{th} clear receptions of application messages from u at v, and $R_u(v) = \max_i R_u^i(v)$. In order to measure time we denote $\Delta R_u^i(v)$ the time (number of time slots) that are between the $(i-1)^{th}$ and the i^{th} clear receptions from u at v, and $\Delta R_u(v) = \max_i \Delta R_u^i(v)$. Similarly, Let $T^i(u)$ be the number of transmissions from u between the $(i-1)^{th}$ and the i^{th} clear transmissions from u, and $T(u) = \max_i T^i(u)$; and let $\Delta T^i(u)$ be the time between the $(i-1)^{th}$ and the i^{th} clear transmission from u, and $\Delta T(u) = \max_i \Delta T^i(u)$.

We define the *message complexity* of a protocol for Recurring Reception as $\max_{(u,v)} R_u(v)$, over all pairs (u, v) of adjacent nodes; and for Recurring Transmission as $\max_u T(u)$ over all nodes u. We define the *delay* of a protocol for Recurring Reception as $\max_{(u,v)} \Delta R_u(v)$, over all pairs (u, v) of adjacent nodes; and for Recurring Transmission as $\max_u \Delta T(u)$ over all nodes u. Any of these definitions is valid for the Recurring Selection problem since clear transmissions and clear receptions are the same event in a single-hop network.

Unless otherwise stated, throughout the paper we assume the presence of an adversary that gets to choose the time step of activation of each node. Additionally, for Recurring Selection, the adversary gets to choose which are the active nodes; and for Recurring Reception and Recurring Transmission, given a topology where each node has at most $k-1$ adjacent nodes, the adversary gets to choose which is the identity of each node. In other words, the adversary gets to choose which of the n schedules is assigned to each node.

Constraints such as limited life cycle and unreliability imply that nodes may power on and off many times. Were such a behaviour unrestricted and controlled by an unbounded adversary, the delay of any protocol could be infinite. Therefore, we assume that active nodes that become inactive are not activated back. The study of the problem under other models of adversarial failures is left for future work.

1.3. Related Work

In [1], Alon, Bar-Noy, Linial and Peleg gave a deterministic distributed protocol to simulate the message passing model in radio networks. Using this technique, each node receives a transmission of all its neighbors after $O(k^2 \log^2 n / \log(k \log n))$ steps. Unfortunately, simultaneous activation of nodes and $\omega(\log n)$ memory size is required. In the same paper, lower bounds for this problem are also proved by showing bipartite graphs that require $\Omega(k \log k)$ rounds. Bipartite graphs with maximum degree $\omega(1)$ are not embeddable in geometric graphs therefore these bounds do not apply to our setting.

The question of how to diseminate information in Radio Networks has led to different well-studied important problems such as *Broadcast* [2, 22] or *Gossiping* [24, 4]. However, deterministic solutions for these problems [8, 6, 10, 5] include assumptions such as simultaneous startup or the availability of a global clock, which are not feasible in Sensor Networks.

The selection problem previously defined was studied [20] in *static* and *dynamic* versions. In static selection all nodes are assumed to start simultaneously, although the choice of which are the active nodes is adversarial. Instead, in the dynamic version, the activation schedule is also adversarial. For static selection, Komlos and Greenberg showed in [19] a non-constructive upper bound of $O(k \log(n/k))$ to achieve *one* successful transmission. More recently, Clementi, Monti, and Silvestri showed for this problem in [9] a tight lower bound of $\Omega(k \log(n/k))$ using intersection-free families. For k distinct successful transmissions, Kowalski presented in [20] an algorithm that uses $(2^{\ell-1}, 2^{\ell}, n)$-selectors for each ℓ. By combining this algorithm and the existence upper bound of [3] a $O(k \log(n/k))$ is obtained. Using Indyk's constructive selector, a $O(k \text{ polylog } n)$ is also proved. These results take advantage of the fact that in the selection problem nodes turn off upon successful transmission. For dynamic selection, Chrobak, Gąsieniec and Kowalski [7] proved the existence of $O(k^2 \log n)$ for dynamic 1-selection. Kowalski [20] proved $O(k^2 \log n)$ and claimed $\Omega(k^2/\log k)$ both by using the probabilistic method, and $O(k^2 \text{ polylog } n)$ using Indyk's selector.

A related line of work from combinatorics is (k, n)-*selective families*. Consider the subset of nodes that transmit in each time slot. A family $\mathcal{R}$ of subsets of $\{1, \ldots, n\}$ is (k, n)-selective, for a positive integer k, if for any subset Z of $\{1, \ldots, n\}$ such that $|Z| \leq k$ there is a set $S \in \mathcal{R}$ such that $|S \cap Z| = 1$. In terms of Radio Networks, a set of n sequences of time slots where a node transmits or receives is (k, n)-selective if for any subset Z of k nodes, there exists a time slot in which exactly one node in the subset transmits. In [18] Indyk gave a constructive proof of the existence of (k, n)-selective families of size $O(k \text{ polylog } n)$. A natural generalization of selective families follows.

Definition 4. [3] Given integers k, m, and n, with $1 \leq m \leq k \leq n$, we say that a boolean matrix M with t rows and n columns is a (k, m, n)-selector if any submatrix of M obtained by choosing k out of n arbitrary columns of M contains at least m distinct rows of the identity matrix I_k . The integer t is the size of the (k, m, n)-selector.

In [11] Dyachkov and Rykov showed that (k, m, n)-selectors must have size $\Omega(\min\{n, k^2 \log_k n\})$ when $m = k$. Recently in [3], it was shown that (k, k, n)-selectors must have size $t \geq (k-1)^2 \log n/(4 \log(k-1) + O(1))$ using superimposed codes. In the same paper, it was shown the existence of (k, k, n)-selectors of size $O(k^2 \ln(n/k))$.

Regarding randomized protocols, an optimal $O(D+k)$-algorithm for gossiping in a Sensor Network of diameter D was presented in [14]. The algorithm includes a preprocessing phase that allows to achieve global synchronism and to implement a collision detection mechanism. After that, nodes transmit their message to all neighboring nodes within $O(k + \log^2 n \log k)$

steps with high probabiliy. The expected message complexity of such phase is $O(\log n + \log^2 k)$. A non-adaptive randomized algorithm that achieves one clear transmission for each node w.h.p. in $O(k \log n)$ steps was shown in [13]. The expected message complexity of such a protocol is $O(\log n)$. In the same paper it was shown that such a running time is optimal for fair protocols, i.e., protocols where all nodes are assumed to use the same probability of transmission in the same time slot.

1.4. Our Results

Our objective is to find deterministic algorithms that *minimize the message complexity* and, among those, algorithms that attempt to minimize the delay. As in [19], we say that a protocol is *oblivious* if the sequence of transmissions of a node does not depend on the messages received. Otherwise, we call the protocol *adaptive*. We study deterministic oblivious and adaptive protocols for Recurring Selection, Recurring Reception and Recurring Transmission. These problems are particularly difficult due to the arbitrary activation schedule of nodes. In fact, the study of oblivious protocols is particularly relevant under adversarial activation of nodes, given their simplicity as compared with adaptive protocols where usually different phases need to be synchronized. If we were able to weaken the adversary assuming that all nodes are activated simultaneously, as it is customary in the more general Radio Network model, the following well-known oblivious algorithm would solve these problems optimally.

For each node i,

node i transmits in time slot $t = i + jn$, $\forall j \in \mathbb{N} \cup \{0\}$.

The message complexity for this algorithm is 1 which of course is optimal. To see why the delay of n is optimal for a protocol with message complexity 1, assume that there is an algorithm with smaller delay. Then, there are at least two nodes that transmit in the same time slot. If these nodes are placed within one-hop their transmissions will collide, hence increasing the message complexity.

We first study oblivious protocols. We show that the message complexity of any oblivious deterministic protocol for these problems is at least k. Then, we present a message-complexity optimal oblivious deterministic protocol, which we call Primed Selection, with delay at most $k(n + k)(\ln(n + k) + \ln\ln(n+k))$. We then evaluate the time efficiency of such a protocol studying lower bounds for these problems. Since a lower bound for Recurring Selection is also a lower bound for Recurring Reception and Recurring Transmission, we concentrate on the first problem. By giving a mapping between (m, k, n)-selectors and Recurring Selection, we establish that $\Omega(k^2 \log n/ \log k)$ is a lower bound for the delay of any protocol that solves Recurring Selection. Maintaining the optimal message complexity may be a good approach to improve this bound. However, the memory size limitations motivates the study of protocols with some form of periodicity. Using a simple argument we show that the delay of any protocol that solves Recurring Selection is in $\Omega(kn)$, for the important class of *equiperiodic protocols,* i.e., protocols where each node transmits with a fixed frequency. Finally, we show that choosing appropriately the periods that nodes use, for $k \leq n^{1/(2\log\log n)} - \log n$ Primed Selection is also optimal delay-wise for equiperiodic protocols. Given that most of the research work in Sensor Networks assumes a logarithmic one-hop density of nodes, Primed Selection is optimal in general for most of the values of k and the delay is only a logarithmic factor from optimal for arbitrary graphs.

Moving to adaptive protocols, we show how to implement a preprocessing phase using Primed Selection so that the delay is reduced to $O(k^2 \log k)$ relaxing the node-memory size and to an asymptotically optimal $O(k)$ additionally limiting the adversarial wake-up schedule.

To the best of our knowledge, no message-complexity lower bounds for recurring communication with randomized oblivious protocols have been proved. Nevertheless, the best algorithm known to solve Recurring Selection w.h.p. is to repeatedly transmit with probability $1/k$ which solves the problem with delay $O(k \log n)$ and expected message complexity in $O(\log n)$. Therefore, deterministic protocols outperform this randomized algorithm for $k \in o(\log n)$ and for settings where the task has to be solved with probability 1.

1.5. Roadmap

Oblivious and adaptive protocols are studied in Sections 2 and 3 respectively. Lower bounds are studied for message complexity in Section 2.1 and for the delay in Section 2.3. The Primed Selection oblivious protocol is presented and analyzed in Section 2.2. An improvement of this algorithm for most of the values of k is shown in Section 2.4 whereas adaptive protocols that use Primed Selection are given in Sections 3.1 and 3.2. We finish with some acknowledgements.

2. Oblivious Protocols

2.1. Message-Complexity Lower Bound

The message complexity for Recurring Selection is at least k. To see why, consider the adversarial node-activation schedule. For any given protocol, the adversary may choose the time of activation of some subset of k nodes so that, within an interval of time steps where one of them produces its first $k+1$ transmissions, only the first and the last one are successful, and the others fail due to collision with other $k-1$ transmissions. We establish formally this observation in the following theorem.

Theorem 1. *Any oblivious deterministic algorithm that solves the Recurring Selection problem, on an n-node single-hop Radio Network where k nodes are activated, possibly at different times, has a message complexity of at least k.*

A lower bound on the message complexity of any protocol that solves Recurring Selection is also a lower bound for Recurring Reception and Recurring Transmission, an observation that we formalize in the following theorem.

Theorem 2. *Given an n-node multihop Radio Network, where the maximum degree is $k-1<n$, and nodes are activated possibly at different times, any oblivious deterministic algorithm that solves the Recurring Reception problem, and any oblivious deterministic algorithm that solves the Recurring Transmission problem, has a message complexity of at least k.*

Proof. We concentrate on proving the claim for Recurring Reception. The same argument can be used for Recurring Transmission. For the sake of contradiction, assume that there exists a protocol $\mathcal{P}$ that solves Recurring Reception with message complexity $t<k$. As argued in Section 2.2, given that at most k nodes are activated in a one-hop Radio Network where the Recurring Selection problem must be solved, the maximum degree on such network is at most $k-1$. Thus, the protocol $\mathcal{P}$ can be used to solve Recurring Selection. But this is contradiction with Theorem 1 where it was proved that the message complexity of any Recurring Selection protocol is at least k. □

2.2. A Message-Complexity-Optimal Protocol: Primed Selection

In the following sections we present our *Primed Selection* protocol for deterministic communication. Such a protocol solves Recurring Selection, Recurring Reception and Recurring Transmission with the same asymptotic cost. For clarity, we first analyze the protocol for Recurring Selection, then we extend the analysis to Recurring Reception and finally we argue why Recurring Transmission is solved with the same asymptotic cost.

A static version of the Recurring Selection problem, where k nodes are activated simultaneously, may also be of interest. For the case $k = 2$, a $(k \log_k n)$-delay protocol can be given recursively applying the following approach. First, evenly split the nodes in two subsets. Then, in the first step one subset transmits and the other receives and in the next one the roles are reversed. Finally, recursively apply the same process to each subset.

Recurring Selection. Recall that the choice of which are the active nodes and the schedule of activations is adversarial. In principle, k different schedules might suffice to solve the problem. However, if only s different schedules are used, for any $s < n$ there exists a pair of nodes with the same schedule. Then, since the protocols are oblivious, if the adversary activates that pair at the same time the protocol would fail. Instead, we define a set of schedules such that each node in the network is assigned a different one.

We assume that, for each node with ID i, a prime number $p(i)$ has been stored in advance in its memory so that $p(1) = p_j < p(2) = p_{j+1} \ldots p(n) = p_{j+n-1}$. Where p_ℓ denotes the ℓ-th prime number and p_j is the smallest prime number bigger than k. Notice that the biggest prime used is $p(n) < p_{n+k} \in O(n \log n)$ by the prime number theorem [17]. Hence, its bit size is in $O(\log n)$. Thus, this protocol works in a small-memory model. The algorithm, which we call *Primed Selection* is simple to describe.

For each node i with assigned prime number $p(i)$,

node i transmits with period $p(i)$.

Theorem 3. *Given a one-hop Radio Network with n nodes, where k nodes are activated perhaps at different times and $6 \leq k \leq n$, Primed Selection* *solves the Recurring Selection problem with delay at most* $k(n+k)(\ln(n+k)+\ln\ln(n+k))$ *and optimal message complexity per successful transmission of* k.

Proof. If no transmission collides with any other transmission we are done, so let us assume that there are some collisions. Consider a node i whose transmission collides with the transmission of a node $j \neq i$ at time t_c. Since $p(i)$ and $p(j)$ are coprimes, the next collision among them occurs at $t_c + p(i)p(j)$. Since $p(i)p(j) > p(i)k$, j does not collide with i within the next $kp(i)$ steps. Node i transmits at least k times within the interval $(t_c, t_c + kp(i)]$. There are at most $k-1$ other active nodes that can collide with i. But, due to the same reason, they can collide with i only once in the interval $[t_c, t_c+kp(i)]$. Therefore, i transmits successfully at least once within this interval. In the worst case $i = n$, so the delay is at most $kp(n) < kp_{n+k}$. Given that $p_x < x(\ln x + \ln\ln x)$ for any $x \geq 6$ [26], the claimed time delay follows. Since every node transmits successfully at least once every k transmissions, the message complexity is k, which is optimal as shown in Theorem 1. □

Recurring Reception.

Although Recurring Selection is defined for a one-hop network of n nodes and, consequently, the maximum degree is potentially $n-1$, the definition of the problem restricts the number of active nodes to k. Thus, the maximum degree is limited to $k-1$ as in Recurring Reception and Recurring Transmission. Thus, a protocol for any of the latter problems can be used to solve

Recurring Selection. The reverse is not so clear because two additional issues appear: the restrictions of sensor nodes and the interference among one-hop neighborhoods. As mentioned, Primed Selection works under the constraints of the Weak Sensor Model. We show in this section that interference is also handled.

Recall that in the Recurring Reception problem n nodes of a Sensor Network are activated, possibly at different times, the maximum number of neighbors of any node is bounded by some value $k-1<n$, and every active node must receive from all of its active neighboring nodes periodically forever. The non-active nodes do not participate in the protocol. Recall that the choice of which are the active nodes and the schedule of activations is adversarial.

Theorem 4. *Given a Sensor Network with n nodes, where the maximum number of nodes adjacent to any node is $k-1$ and $6 \leq k \leq n$, Primed Selection solves the Recurring Reception problem with delay at most $k(n+k)(\ln(n+k)+\ln\ln(n+k))$ and optimal message complexity per reception of k.*

Proof. Consider any node u and the set of its adjacent nodes $N(u)$. If u receives the transmissions of all its neighbors without collisions we are done. Otherwise, consider a pair of nodes $i,j \in N(u)$ that transmit –hence, collide at u– at time t_c. Since $p(i)$ and $p(j)$ are coprimes, the next collision among them at u occurs at time $t_c+p(i)p(j)$. Since $p(i)p(j)>p(i)k$, j does not collide with i at u within the next $kp(i)$ steps. Node i transmits at least k

times within this interval. There are at most $k-2$ other nodes adjacent to u that can collide with i at u, and of course u itself can collide with i at u. But, due to the same reason, they can collide with i at u only once in the interval $[t_c, t_c + kp(i)]$. Therefore, i transmits without collision at u at least once within this interval. Since $i \leq n$, the delay is at most $kp(n) < kp_{n+k}$, which is at most $k(n+k)(\ln(n+k)+\ln\ln(n+k))$ as proved in [25] for $n+k \geq 6$. Thus, the claimed time delay follows. The transmission of every node is received by some neighboring node at least once every k transmissions, which is optimal as shown in Theorem 1. $\square$

Recurring Transmission. Observe that Primed Selection solves the Recurring Transmission problem also, modulo an additional factor of 7 in the analysis, because any two-hop neighborhood has at most $7k$ nodes, by a simple geometric argument based on the optimality of an hexagonal packing [15].

2.3. Delay Lower Bounds

De Bonis, Gąsieniec and Vaccaro have shown [3] a lower bound of $((k-m+1)\lfloor(m-1)/(k-m+1)\rfloor^2/(4\log(\lfloor(m-1)/(k-m+1)\rfloor)+O(1)))\log(n/(k-m+1))$ on the size of (k,m,n)-selectors when $1 \leq m \leq k \leq n$ and $k < 2m-2$. When $m = k > 2$, this lower bound gives a lower bound of $\Omega(k^2 \log n/\log k)$ for the delay of any protocol that solves Recurring Selection. To see why, recall Definition 4.

Now, assume that there exists a protocol $\mathcal{P}$ for Recurring Selection with delay in $o(k^2 \log n/\log k)$. Recall that a protocol for Recurring Selection is a set of schedules of transmissions. Assuming that all nodes start simultaneously, consider such a set of schedules. By definition of Recurring Selection, for each choice of k schedules of $\mathcal{P}$, i.e., active nodes, there exists a positive integer $t \in o(k^2 \log n / \log k)$ such that in every time interval of length t each active node must achieve at least one non-colliding transmission.

Representing a transmission with a 1 and a reception with a 0, the set of schedules can be mapped to a matrix M where each time step is a row of M and each schedule is a column of M. The arbitrary choice of k active nodes is equivalent to choosing k arbitrary columns of M. The time steps where each of the k active nodes achieve non-colliding transmissions gives the $m = k$ distinct rows of the identity matrix I_k in M. Therefore, there exists a (k, k, n)-selector of size in $o(k^2 \log n / \log k)$ which violates the aforementioned lower bound. Thus, $\Omega(k^2 \log n / \log k)$ is a lower bound for the delay of any protocol that solves Recurring Selection and, as shown before, a lower bound for Recurring Selection is a lower bound for Recurring Reception and Recurring Transmission.

Recall that our main goal is to minimize the message complexity. Hence, this lower bound might be increased if we maintain the constraint of k message complexity. Nevertheless, in order to obtain a better lower bound, we will use the memory size constraint present in the Weak Sensor Model (and any Radio Network for that matter) which leads to protocols with some form of periodicity.

We define an *equiperiodic protocol* as a set of schedules of transmissions where, in each schedule, every two consecutive transmissions are separated by the same number of time slots. A simple lower bound of $\Omega(kn)$ steps for the delay of any equiperiodic protocol that solves Recurring Selection can be observed as follows. n different periods are necessary otherwise two nodes can collide forever. At least k transmissions are necessary within the delay to achieve one reception successfully as proved in Theorem 1. Therefore, there exist a node with delay at least kn, which we formalize in the following theorem.

Theorem 5. *Any oblivious equiperiodic protocol that solves Recurring Selection in a one-hop Radio Network with n nodes, where k of them are activated possibly at different times, has delay at least kn.*

2.4. A Delay-Optimal Equiperiodic Protocol

In Primed Selection, the period of each node is a different prime number. However, in order to achieve optimal message complexity as proved in Theorem 1, it is enough to use a set of n periods such that, for each pair of distinct periods u, v, it holds that $v/\gcd(u,v) \geq k$ and $u/\gcd(u,v) \geq k$. In this section, we define such a set of periods so that, when used as periods in Primed Selection, gives optimal delay for equiperiodic protocols when $k \leq n^{1/(2\log\log n)} - \log n$.

The idea is to use a set of composite numbers each of them formed by $\log\log n$ prime factors taken from the smallest $\log n$ primes bigger than k. More precisely, we define a *compact set* C as follows. Let p_ℓ denote the ℓ-

th prime number. Let p_μ be a prime number such that $p_\mu = 2$ if $k \leq 2$, and $p_{\mu-1} < k \leq p_\mu$ otherwise. Let P be the set of prime numbers $P = \{p_\mu, p_{\mu+1}, \ldots, p_{\mu+\log n-1}\}$. Let $\mathcal{F}$ be a family of sets such that $\mathcal{F} = \{F | (F \subset P) \wedge (|F| = \log\log n)\}$. Make C a set of composite numbers such that $C = \{c_F | c_F = (\prod_{i \in F} i) \wedge (F \in \mathcal{F})\}$. The following lemma shows that the aforementioned property holds in a compact set.

Lemma 1. *Given a positive integer $k \leq n$ and a compact set C defined as above, $\forall u, v \in C, u \neq v$ it holds that $v/\gcd(u,v) \geq k$ and $u/\gcd(u,v) \geq k$.*

Proof. For the sake of contradiction, assume that there exists a pair $u, v \in C, u \neq v$ such that either $v/\gcd(u,v) < k$ or $u/\gcd(u,v) < k$. Let $U = \{u_1, u_2, \ldots, u_{\log\log n}\}$ and $V = \{v_1, v_2, \ldots, v_{\log\log n}\}$ be the sets of prime factors of u and v respectively. Given that the prime factorization of a number is unique and that $|U| = |V|$, there must exist $u_i \in U$ and $v_j \in V$ such that $u_i \notin V$ and $v_j \notin U$. But then $u/\gcd(u,v) \geq u_i \geq k$ and $v/\gcd(u,v) \geq v_i \geq k$. □

We assume that, for each node with ID i, a number $P(i) \in C$ has been stored in advance in its memory so that no two nodes have the same number. It can be derived that $|C| = \binom{\log n}{\log\log n} \geq n$ for large enough values of n. Hence, C is big enough as to assign a different number to each node.

In order to show the delay-optimality of this assignment it remains to be proved that the biggest period is at most n, which we do in the following lemma.

Lemma 2. *Given a positive integer* $6 \leq k \leq n^{1/(2\log\log n)} - \log n$ *and a compact set* C *defined as above,* $\max_{c \in C}\{c\} \leq n$.

Proof. Given that $p_k > k \log k$ for any $k \geq 1$ [26], in order to form the compact set C it is enough to use the prime numbers $\{p_k, \ldots, p_{k+\log n}\}$. Hence, in order to prove the claim, it is enough to prove $(p_{k+\log n})^{\log\log n} \leq n$. Given that $p_x < x(\ln x + \ln\ln x)$ when $x \geq 6$ [26], we want to prove

$$\big((k + \log n)(\ln(k + \log n) + \ln\ln(k + \log n))\big)^{\log\log n} \leq n.$$

Manipulating, it can be verified that the inequality is true for $k \leq n^{1/(2\log\log n)} - \log n$. ☐

Now we are in conditions to state the main theorem for Recurring Selection which can be proved using Lemmas 1 and 2 and Theorems 1 and 5, and can be extended to Recurring Reception and Recurring Transmission.

Theorem 6. *Given a one-hop Radio Network with* n *nodes, where* $k \leq n^{1/(2\log\log n)} - \log n$ *nodes are activated perhaps at different times, using a compact set of periods Primed Selection solves the Recurring Selection problem with optimal message complexity* k *and* kn *delay, optimal for equiperiodic protocols.*

The good news is that this value of k is actually very big for most of the applications of Sensor Networks, where a logarithmic density of nodes in any one-hop neighborhood is usually assumed.

3. Adaptive Protocols

In this section we study adaptive protocols for recurrent communication. First, we present a protocol that improves the delay upper bound over Primed Selection by utilizing a bigger node-memory size. Then, we present a delay optimal protocol by additionally restricting the adversarial wake-up schedule. Given that in these protocols nodes run a pre-processing phase without delay guarantees, the efficiency metrics defined for oblivious protocols are reused, but only after nodes have finished that phase.

3.1. Reduced Primed-Selection

The same technique used in Primed Selection yields a reduced delay if we use only $O(k)$ coprime periods in the whole network as long as we guarantee that, for every node u, every pair of nodes $i, j \in N(u) \cup \{u\}$ use different coprimes. However, given that the topology is unknown, it is not possible to define an oblivious assignment that works under our adversary.

In this section, we show how to reduce the delay for Recurring Reception introducing a pre-processing phase in which nodes make use of Primed Selection to self-assign those primes appropriately. As argued in Section 2.2, a protocol for Recurring Reception can also be used to solve Recurring Selection without extra cost and Recurring Transmission with constant overhead. Given that in this protocol it is necessary to maintain a set of k primes, we relax the node-memory constraint of the Weak Sensor Model as follows. If the computational power of each node is such that the computation time is negligible with respect to the communication time (so that prime numbers in an interval can be easily computed) it is enough to maintain membership to that set. Thus, we assume that the memory size of each node is bounded only by $O(k + \log n)$ bits. If that is not the case, we assume that nodes have already in memory those prime numbers. Hence, by the prime number theorem, the memory bound becomes $O(k^2 \log k + \log n)$. We further assume that nodes are deployed densely enough so that if we reduce the radius of transmission to $r/2$ the network is still connected. This assumption introduces only an additional constant factor in the total number of nodes to be deployed n and the maximum degree $k - 1$.

We first give the intuition of the protocol. As before, we use prime numbers bigger than k but, additionally, the smallest k of them are left available. More precisely, each node with ID $i \in 1, \ldots, n$ is assigned a *big* prime number $p(i)$ so that $p(1) = p_{j+k} < p(2) = p_{j+k+1} \ldots p(n) = p_{j+k+n-1}$. Where p_ℓ is the ℓ-th prime number and p_j is the first prime number bigger than k. Again, given that $k \leq n$ and using the prime number theorem [17], the size in bits of the biggest prime is still in $O(\log n)$.

Using their big prime as a period of transmission nodes first compete for one of the k *small* primes left available. Once a node chooses one of these small primes, it announces its choice with period its big prime and transmits its messages with period its small prime. If at a given time slot these transmissions coincide, it is equivalent to the event of a collision of the transmissions of two different nodes. We choose to produce one of them arbitrarily.

In order to prevent two nodes from choosing the same small prime, each node maintains a counter. A node chooses an available small prime upon reaching a final count. When a node reaches its final count and chooses, it is guaranteed that all neighboring nodes lag behind enough so that they receive the announcement of its choice before they can themselves choose a small prime.

In order to ensure the correctness of the algorithm, no two nodes within two hops should choose the same small prime. Therefore, we use as radius of transmission $r/2$ for message communication and r for small-prime announcements.

The protocol is detailed in Algorithm 1. It was shown before that the delay of Primed Selection is at most $k(n+k)(\ln(n+k)+\ln\ln(n+k))$ if $n+k \geq 6$, when the first n primes bigger than k are used. Given that in Reduced Primed-Selection we leave available the smallest k primes bigger than k, the delay of this modified version of Primed Selection is at most $k(n+2k)(\ln(n+2k)+\ln\ln(n+2k))$. For clarity of the presentation, we denote this value as T.

Let us call a node that has chosen a small prime a *decided* node and *undecided* otherwise. In order to prove correctnes, we have to prove that every node becomes decided and that no pair of neighboring nodes choose the same prime.

Lemma 3. *Given any node u that becomes decided in the time slot t, the*

Algorithm 1: Reduced Primed-Selection. Pseudocode for node x with assigned prime number $p(x)$. $T = k(n+2k)(\ln(n+2k) + \ln\ln(n+2k))$. The binary relation $\preceq$ represents component-wise lexicographic order.

```
my-counter ← 0
my-time-awake ← 1
used-small-primes ← ∅
once per time slot while my-counter < 2T do
    if my-time-awake ≡ 0 (mod p(x)) then                                   //
        transmit ⟨count, my-counter, x⟩ with radius r
    else if ⟨count, c, i⟩ is received and (my-counter, x) ⪯ (c, i) then     //
        my-counter ← 0
    else if ⟨prime, p⟩ is received then                                    //
        used-small-primes ← used-small-primes ∪ {p}
    increase my-counter and my-time-awake
end
my-small-prime-period ← q ∉ used-small-primes        // x becomes decided
once per time slot do
    if my-time-awake ≡ 0 (mod p(x)) then                                   //
        transmit ⟨prime, my-small-prime-period⟩ with radius r
    else if my-time-awake ≡ 0 (mod my-small-prime-period) then             //
        transmit ⟨app-message⟩ with radius r/2
    increase my-time-awake
end
```

counter of every undecided node $v \in N(u)$ is at most T in the time slot t.

Proof. Consider a node u that becomes decided at time t. For the sake of contradiction, assume there is an undecided node $v \in N(u)$ whose counter is greater than T at t. By the definition of the algorithm, v did not receive a bigger counter for more than T steps before t, and u did not receive a bigger counter for $2T$ steps before t. In the interval $[t-T, t]$ the local counter of u is larger than the local counter of v. As shown in Theorem 4, v must receive from u within T steps. But then, v must have been reset in the interval $[t-T, t]$. □

Theorem 7. *Given a Sensor Network with n nodes, where the maximum degree is $k-1$ and $6 \leq k \leq n$, if nodes run Reduced Primed-Selection, no pair of neighboring nodes choose the same small prime and every node becomes decided within $O(Tn^2)$ steps after starting running the algorithm.*

Proof. The first statement is a direct conclusion of Lemma 3 and Theorem 4. For the second statement, if a node u is not reset within T steps no neighbor of u has a bigger counter and u will become decided within $2T$ steps. Thus, it takes at most $(n+1)T$ steps for the first node in the network that becomes decided. By definition of the algorithm, a decided node does not reset the counter of any other node. Applying the same argument recursively the claim follows. □

Theorem 8. *Given a Sensor Network with n nodes, where the maximum degree is $k-1$ and $6 \leq k \leq n$, after the pre-processing, Reduced Primed-*

Selection solves the Recurring Reception problem with delay at most $2k^2(\ln(2k) + \ln\ln(2k))$ *and optimal message complexity of* k*.*

Proof. As in Theorem 4. □

3.2. Optimal Delay

Due to the pigeonhole principle, any protocol for recurring communication as defined has a delay of at least $k-1$. Using adaptiveness, it was shown in Section 3.1 how to obtain a delay of $O(k^2 \log k)$ relaxing the memory size restrictions. A natural question is how to improve further the delay guarantee, perhaps at the cost of relaxing other restrictions. In this section, we show that an asymptotically optimal delay of $O(k)$ can be achieved by restricting the adversarial node-activation schedule, assuming that each node memory-size is bounded only by $O((2(n+k)\ln(n+k))^k)$ bits.

We assume the presence of an adversary that activates each node at an arbitrary time slot, but only τ time slots may separate the first and last node-activation times. Nodes that are not activated during this time frame will not become active at all, for instance due to failures. As in Section 3.1, we further assume that nodes are deployed densely enough so that, if the radius of transmission is reduced by a constant factor, the network is still connected, introducing only a constant factor overhead in n and k. As before, we focus in Recurring Reception given that Recurring Selection can be solved without extra cost and Recurring Transmission introduces only a constant factor overhead, as shown in Section 2.2.

The protocol presented in this section, named *Prime-Compressed Selection,* includes the same preprocessing technique used in Reduced Primed-Selection (Section 3.1). Thus, in the description that follows, we focus on the differences with respect to Reduced Primed-Selection and we reuse previous proofs.

Upon starting up, nodes compete using Primed Selection, i.e., transmitting with their assigned prime period and radius r, although in this case we use prime numbers bigger than $2k$. Instead of competing for a small prime as in Reduced Primed-Selection, each node competes to reserve some slots among those left available by the schedule of transmissions of Primed Selection. To decide when it is safe to choose slots for reservation, each node uses a counter as in Reduced Primed-Selection. After choosing and announcing its choice, each node uses those reserved slots to produce all its future recurrent transmissions, using as radius of transmission $r/2$ to avoid the hidden-terminal problem. Reusing previous notation, a node that has already chosen slots is called *decided* node and *undecided* otherwise.

In order to disseminate the information needed to choose the slots left available, upon activation, each node repeatedly transmits its pre-assigned prime period, so that neighboring nodes within radius r keep track of slots used. Waiting long enough, a node receives this information from all its neighboring nodes that will be ever active. Armed with this information, each node u creates a dynamic map, in its local memory, of the transmissions scheduled in its neighborhood in the next interval of length $p(u)\prod_{i\in N(u)} p(i)$, i.e., the period of the schedule of transmissions of u and its neighbors. The map is dynamic in the sense that u updates the schedule to the next interval for each time step and, additionally, u adds the information about slots used by its neighbors upon receiving it. Based on this map, upon reaching its final count, u makes its choice of available slots and announces its reservation of incoming slots. The map of scheduled transmissions can be stored in u's memory because $\prod_{i \in N(u)} p_i < p_{n+k}^k < ((n+k)(\ln(n+k) + \ln\ln(n+k)))^k$ by [26].

Further details about the Prime-Compressed Selection protocol can be found in Algorithm 2. We denote the upper bound on the delay of Primed Selection using the smallest n primes bigger than $2k$ as in the previous section $T = k(n+2k)(\ln(n+2k) + \ln\ln(n+2k))$. In the rest of the section, we prove correctness and efficiency.

Theorem 9. *Given a Sensor Network with n nodes, where the maximum degree is $k-1$ and $6 \leq k \leq n$, if nodes run Prime-Compressed Selection, no pair of neighboring nodes choose the same slot and every node becomes decided within $O(\tau + Tn^2)$ steps after starting running the algorithm.*

Proof. We have to prove that every node becomes decided and that no pair of neighboring nodes choose the same slot. The latter can be proved as in Theorem 7. For the former, it has to be proved that all nodes have slots available to make their choice, which we prove as follows.

Claim 1. For any set of k out of n nodes running Prime-Compressed Selec-

Algorithm 2: Prime-Compressed Selection. Pseudocode for node x with assigned prime number $p(x)$. $T = k(n+2k)(\ln(n+2k)+\ln\ln(n+2k))$. The binary relation $\preceq$ represents component-wise lexicographic order.

```
my-time-awake ← 1
set a circular queue of bits neighboring-transmissions according to p(x)
once per time slot while my-time-awake < τ + T do
    if my-time-awake ≡ 0 (mod p(x)) then                                  //
        transmit ⟨prime, p(x)⟩ with radius r
    else if ⟨prime, p⟩ or ⟨count, c, p⟩ is received then                  //
        update neighboring-transmissions according to p
    else if ⟨slots, s, p⟩ is received then                                //
        update neighboring-transmissions according to s and p
    increase my-time-awake
    shift neighboring-transmissions for the next time step
end
my-counter ← 0
once per time slot while my-counter < 2T do
    if my-time-awake ≡ 0 (mod p(x)) then                                  //
        transmit ⟨count, my-counter, p(x)⟩ with radius r
    else if ⟨prime, p⟩ is received then                                   //
        update neighboring-transmissions according to p
    else if ⟨count, c, p⟩ is received then                                //
        if (my-counter, x) ⪯ (c, p) then my-counter ← 0
        update neighboring-transmissions according to p
    end
    else if ⟨slots, s, p⟩ is received then                                //
        update neighboring-transmissions according to s and p
    increase my-counter and my-time-awake
    shift neighboring-transmissions for the next time step
end
set a circular queue of bits my-reserved-slots according to neighboring-transmissions
once per time slot do
    if my-time-awake ≡ 0 (mod p(x)) then                                  //
        transmit ⟨slots, my-reserved-slots, p(x)⟩ with radius r
    else if my-time-awake is in my-reserved-slots then                    //
        transmit ⟨app-message⟩ with radius r/2
    increase my-time-awake
    shift my-reserved-slots for the next time step
end
```

tion and for any interval of $2k$ time slots, there are at least k slots that are not used for preprocessing transmissions.

Proof. For the sake of contradiction, assume the claim is false. Then, for some set of k nodes, there is an interval of length $2k$ such that more than k slots are used for transmissions with period a prime number. However, given that the smallest prime is $\geq 2k$, each of those transmissions correspond to a different node. Given that there are at most k nodes in any neighborhood, this is a contradiction. □

Regarding the running time, waiting for $\tau + T$ time slots is enough to guarantee that each node knows the schedule of transmissions of its neighborhood, because within τ slots all neighboring nodes will be activated and a non-colliding transmission is received from all neighboring nodes within Primed Selection maximum delay of T. On the other hand, $O(Tn^2)$ steps are enough for all nodes to become decided as shown in Theorem 7. Thus, the claim follows. □

The following theorem establishes the result presented in this section.

Theorem 10. *Given a Sensor Network with n nodes, where the maximum degree is $k-1$ and $6 \leq k \leq n$, after the pre-processing, Prime-Compressed Selection solves the Recurring Reception problem with message complexity of 1 and delay in $O(k)$.*

4. Conclusions

In this paper, deterministic communication under restricted models of Radio Networks has been studied. The metrics used to establish efficiency were the overhead on the number of transmissions that a node has to produce to effectively communicate with its neighbors, and the delay produced by this overhead. Simple pigeonhole-principle arguments yielded lower bounds for these metrics, both matched for the important class of equiperiodic protocols, and in message complexity for oblivious protocols. Relaxing the node-memory size and the node-activation schedule constraints, it was shown how to improve the time delay using adaptiveness, matching the lower bound. We leave for future work the study of upper bounds that hold even without these relaxations or lower bounds that show that such goal is not feasible. Another important line of work is to consider models were the total number of nodes n or the number of nodes activated k is unknown beforehand.

References

[1] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. Single round simulation in radio networks. *Journal of Algorithms*, 13:188–210, 1992.

[2] 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.

[3] A. De Bonis, L. Gąsieniec, and U. Vaccaro. Generalized framework for selectors with applications in optimal group testing. In *Proc. of 30th International Colloquium on Automata Languages and Programming*, volume 2719 of *Lecture Notes in Computer Science*, pages 81–96, 2003.

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

[5] B. S. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in unknown radio networks. *Distributed Computing*, 15:27–38, 2002.

[6] B. S. Chlebus, L. Gąsieniec, A. Östlin, and J. M. Robson. Deterministic radio broadcasting. In *Proc. of 27th International Colloquium on Automata Languages and Programming*, pages 717–728, 2000.

[7] M. Chrobak, L. Gąsieniec, and D. Kowalski. The wake-up problem in multi-hop radio networks. In *Proc. of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 985–993, 2004.

[8] 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*, page 575, 2000.

[9] A.E.F. 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*, pages 709–718, 2001.

[10] A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. *Journal of Algorithms*, 60(2):115–143, 2006.

[11] A.G. Dyachkov and V.V. Rykov. A survey of superimposed code theory. *Problems Control and Inform. Theory*, 12(4), 1983.

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

[13] M. Farach-Colton and M. A. Mosteiro. Initializing sensor networks of non-uniform density in the weak sensor model. In *Proc. of 10th International Workshop on Algorithms and Data Structures*, volume 4619 of *Lecture Notes in Computer Science*, pages 565–576. Springer-Verlag, Berlin, 2007.

[14] 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*, volume 4835 of *Lecture Notes in Computer Science*, pages 232–243. Springer-Verlag, Berlin, 2007.

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

[16] A. Fernández Anta, Miguel A. Mosteiro, and Christopher Thraves. Deterministic communication in the weak sensor model. In *Proceedings of the 11th International Conference on Principles of Distributed Systems*, pages 119–131, 2007.

[17] G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers.* Oxford University Press, 1979.

[18] P. Indyk. Explicit constructions of selectors and related combinatorial structures, with applications. In *Proc. of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 697–704, 2002.

[19] J. Komlòs and A.G. Greenberg. An asymptotically nonadaptive algorithm for conflict resolution in multiple-access channels. *IEEE Trans. Inf. Theory*, 31:303–306, 1985.

[20] D. R. Kowalski. On selection problem in radio networks. In *Proceedings 24th Annual ACM Symposium on Principles of Distributed Computing*, pages 158–166, 2005.

[21] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing newly deployed ad hoc and sensor networks. In *Proc. of the 10th ACM Annual International Conference on Mobile Computing and Networking*, pages 260–274, 2004.

[22] 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.

[23] Philip Levis, Eric Brewer, David Culler, David Gay, Samuel Madden, Neil Patel, Joe Polastre, Scott Shenker, Robert Szewczyk, and Alec Woo. The emergence of a networking primitive in wireless sensor networks. *Commun. ACM*, 51(7):99–106, 2008.

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

[25] B. Rosser. Explicit bounds for some functions of prime numbers. *American Journal of Mathematics*, 63(1):211–232, 1941.

[26] J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6(1):64–94, 1962.

Authors Biographies

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

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

Christopher Thraves is Juan de la Cierva researcher at Universidad Rey Juan Carlos, Madrid, Spain. Previously, he was postdoc researcher at INRIA, France. He spent one year with the research team CEPAGE at Bordeaux and two years with the research team ASAP, Rennes. He got his PhD in Computer Science from the Universidad Rey Juan Carlos in March 2008. His research interests include graph theory, design and analysis of algorithms, and discrete mathematics.