Deterministic Communication in the Weak Sensor Model

Antonio Fernández* Miguel A. Mosteiro† Christopher Thraves‡

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 collision 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 unreallistic 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.

1 Introduction

The 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 such as energy or range of communication.

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 protocols for Sensor Networks use some form of randomness in order to deal with collisions and the lack of topology information. Randomized protocols are usually fast and resilient to failures, but they 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.

*LADyR, GSyC, Universidad Rey Juan Carlos, 28933 Móstoles, Spain; anto@gsyc.escet.urjc.es.

†LADyR, GSyC, Universidad Rey Juan Carlos, 28933 Móstoles, Spain; mosteiro@gsyc.escet.urjc.es.

‡LADyR, GSyC, Universidad Rey Juan Carlos, 28933 Móstoles, Spain; cbthraves@gmail.com.

1.1 Model

Given the limited range of sensor nodes 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.

Given the random deployment of nodes, we assume that the topology of the network is unknown. Nevertheless, such a deployment is not the result of an uncontrolled experiment where any outcome has a positive probability. Hence, we assume that the network is connected and 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 the total size of the network n, its unique identifier in $\{1,\ldots,n\}$ and the maximum degree $k-1$.

Regarding the strong resource-limitations of sensor nodes, we use the comprehensive Weak Sensor Model [11] unless otherwise stated. The following assumptions are included in this model. Time is assumed to be slotted and all nodes have the same clock frequency, but no global synchronizing mechanism is available. Furthermore, nodes are activated adversarially. 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 in a time slot. If more than one message is sent in the same time slot, a collision occurs and no collision detection mechanism is available. Sensor nodes cannot receive and transmit in the same time slot. The channel is assumed to have only two states: transmission and silence/collision. The memory size of each sensor node is bounded by $O(1)$ words of $O(\log n)$ bits. 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 due to energy constraints, short transmission range, only one channel of communication, no position information, and unreliability.

In a time slot, a node can be in one of three states, namely *transmission*, *reception*, or *inactive*. Throughout the paper, we denote a temporal sequence of states of a node as a *schedule of transmissions*, or simply a *schedule* when the context is clear. We will also refer to a node that is in the transmission or reception state as *active*.

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 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 not transmitting, 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 an extended version of the selection problem which 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 nodes are activated possibly at different times, and upon activation 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 nodes are activated possibly at different times, and upon activation 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 $R_u^i(v)$ be the number of transmissions of u between the $(i-1)^{th}$ and the i^{th} clear receptions 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. Also, 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.

Again, any of these definitions is valid for the Recurring Selection problem since a clear transmission and a clear reception is 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.

Among the assumptions of the Weak Sensor Model are limited life cycle and unreliability. These constraints imply that nodes may power on and off many times during its life time. If such a behaviour were adversarial, the delay of any protocol could be infinite. Therefore, we assume that active nodes that become inactive are not activated back.

1.3 Related Work

To the best of our knowledge, recurring deterministic communication primitives have not been studied previously even in the more general Radio Networks model. We briefly overview previous work closely related.

The question of how to diseminate information in Radio Networks has led to different well-studied important problems such as *Broadcast* [1, 19] or *Gossiping* [3, 20]. However, deterministic solutions for these problems [4, 5, 7, 9] 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 [18] 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, i.e. the time at which each node is activated, is also adversarial.

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, in each time slot a node may transmit or receive. 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 [16] Indyk gave a constructive proof of the existence of (k, n)-selective families of size $O(k \operatorname{polylog} n)$.

In the previous problem only one in every subset of k nodes must achieve a non-colliding transmission. The following problem is a generalization to $m \leq k$. Given integers m, k, n, with $1 \leq m \leq k \leq n$, we say that a boolean matrix M with t rows and n columns is a (m, k, n)-*selector* if any submatrix of M obtained by choosing arbitrarily k out of the n columns of M contains at least m distinct rows of the identity matrix I_k. The integer t is the size of the (m, k, n)-selector. In [10] Dyachkov and Rykov showed that (m, k, n)-selectors must have a size $\Omega(\min\{n, k^2 \log_k n\})$ when $m = k$. Recently in [2], De Bonis, Gąsieniec and Vaccaro showed 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 $O(k^2 \ln(n/k))$.

For the static selection problem, Komlos and Greenberg showed in [17] 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 [8] a tight lower bound of $\Omega(k \log(n/k))$ using intersection-free families. For k distinct successful transmissions, Kowalski presented in [18] an algorithm that uses $(2^{\ell-1}, 2^{\ell}, n)$-selectors for each ℓ. By combining this algorithm and the existence upper bound of [2] a $O(k \log(n/k))$ is obtained. Using Indyk's constructive selector, a $O(k \operatorname{polylog} n)$ is also proved. These results take advantage of the fact that in the selection problem nodes turn off upon successful transmission.

For dynamic selection, in [6], Chrobak, Gąsieniec and Kowalski proved the existence of $O(k^2 \log n)$ for dynamic 1-selection. In [18] Kowalski proved $O(k^2 \log n)$ and claimed $\Omega(k^2 / \log k)$ both by using the probabilistic method, and $O(k^2 \operatorname{polylog} n)$ using Indyk's selector.

Regarding randomized protocols, for a Sensor Network of diameter D, an optimal $O(D + k)$ algorithm for gossiping is presented in [13]. This algorithm includes a preprocessing phase that gives a structure to the network that allows to achieve global synchronism and to implement a collision detection mechanism. Based on that, the algorithm includes a phase in which all 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 $O(\log n + \log^2 k)$.

An non-adaptive randomized algorithm that achieves one clear transmission for each node w.h.p. in $O(k \log n)$ steps was shown in [12]. 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 [17], 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. If we weaken the adversary assuming that all nodes are activated simultaneously, the following well-known oblivious algorithm solves 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 start our analysis with oblivious protocols. We first show that the message complexity of any oblivious deterministic protocol for these problems is at least k. Then, we present a message-complexity optimal protocol, which we call Primed Selection, with delay $O(kn \log n)$. 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 $k \leq n^{1/6 \log \log n}$ 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 $O(k^2 \log k)$.

To the best of our knowledge, no lower bounds of the message complexity 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 an adaptive protocol that uses Primed Selection is given in Section 3.1. We finish with some acknowledgements in Section 4.

2 Oblivious Protocols

2.1 Message-Complexity Lower Bound

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. To see why, we map Recurring Selection into Recurring Reception and viceversa. A similar argument can be given for Recurring Transmission.

Consider a single-hop Radio Network N_S where Recurring Selection is solved and a Sensor Network N_R where Recurring Reception is solved. Consider the set of k active nodes in N_S. There is at least one node i with degree $k-1$ in N_R. Map any of the active nodes in N_S to i and the remaining $k-1$ active nodes in N_S to the neighbors of i in N_R. The adversarial choice of which are the k active nodes in N_S is equivalent to the adversarial choice of which schedules of the protocol are assigned to i and its neighbors in N_R.

Now, for the sake of contradiction, assume that for any protocol that solves Recurring Selection, the message complexity is at least s but there is a protocol $\mathcal{P}$ that solves Recurring Reception with message complexity $r < s$. Then, we can use $\mathcal{P}$ to solve Recurring Selection as follows. Consider a node u adjacent to i in N_R. By definition of Recurring Selection, it is guaranteed that i receives from u every r transmissions of u. Hence, every r transmissions of u there is at least one transmission of u that does not collide with any other node adjacent to i. Since this is true for each of the nodes adjacent to i, Recurring Selection can be solved with message complexity r which is a contradiction.

The proof of a k lower bound for message complexity is based in a simple argument and it is formalized as follows.

Theorem 4. *Any oblivious non-deterministic algorithm that solves the Recurring Selection problem, on an n-node single-hop Radio Network where k nodes are activated, perhaps at different times, has a message complexity of at least k.*

Proof. Assume for the sake of contradiction that there exists a protocol such that some node i achieves a non-colliding transmission every $t < k$ transmissions. But then, an adversary can activate each of the other $k-1$ nodes in such a way that at least one transmission collides with each transmission of i within an interval of t transmissions, which is a contradiction. □

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. We assume 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 [15]. 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 5. *Given a one-hop Radio Network with n nodes, where k nodes are activated perhaps at different times, Primed Selection solves the Recurring Selection problem with delay in $O(kn \log n)$ and the message complexity per successful transmission is k, which is optimal as shown in Theorem 4.*

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, $i = n$ and the delay is in $O(kp(n)) \in O(kn \log n)$. Since every node transmits successfully at least once every k transmissions, the message complexity is k. □

Recurring Reception. A protocol for Recurring Selection may be used to solve the Recurring Reception problem. However, two additional issues appear, namely, the restrictions of sensor nodes and the interference among one-hop neighborhoods. As mentioned before, 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. We assume the choice of which are the active nodes and the schedule of activations to be adversarial.

Theorem 6. *Given a Sensor Network with n nodes, where the maximum number of nodes adjacent to any node is $k - 1 < n$, Primed Selection solves the Recurring Reception problem with delay in $O(kn \log n)$ and the message complexity per reception is k, which is optimal as shown in Theorem 4.*

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 and the claimed delay follows. The transmission of every node is received by some neighboring node at least once every k transmissions. □

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 [14].

2.3 Delay Lower Bounds

De Bonis, Gąsieniec and Vaccaro have shown [2] 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 $1 \leq m \leq k \leq n, k < 2m-2$. 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 that a (k,m,n)-selector is defined as follows

Definition 7. *[2] 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.*

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.

If we represent 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 the set of k active nodes is equivalent to choose 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, using the same argument as in Section 2.1, a lower bound for Recurring Selection is also a lower bound for Recurring Reception and Recurring Transmission.

It is important to notice that our main goal is to minimize the message complexity. Hence, this lower bound might be improved if we maintain the constraint of a message complexity of at most k. 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 4. Therefore, there exist a node with delay at least kn, which we formalize in the following theorem.

Theorem 8. *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 for $k \leq n^{1/6 \log\log n}$

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 4, it is enough to use a set of n periods that verify the following property. 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 a set of periods that verifies this property and, when used as periods in Primed Selection, gives optimal delay for equiperiodic protocols when $k \leq n^{1/6 \log\log n}$.

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 9. *Given a positive integer $k \leq n$ and a compact set C defined as above, for all pairs $u, v \in C, u \neq v$ 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 $v/\gcd(u,v) \geq v_i \geq k$ which is a contradiction. □

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 in $O(n)$ when $k \leq n^{1/6 \log\log n}$, which we do in the following lemma.

Lemma 10. *Given a positive integer $k \leq n^{1/6 \log\log n}$ and a compact set C defined as above,* $\max_{c\in C}\{c\} \in O(n)$.

Proof. Consider the prime number $p_{k+\log n}$. Using the prime number theorem, it can be shown that the number of primes in the interval $[k, p_{k+\log n}]$ is bigger than $\log n$. Hence, in order to prove the claim, it is enough to prove $(p_{k+\log n})^{\log\log n} \in O(n)$. Thus, using the prime number theorem, for some constants α, β we want to prove

$$(\beta(k + \log n)\log(k + \log n))^{\log\log n} \leq \alpha n.$$

Replacing $k \leq n^{1/6 \log\log n}$, the inequality is true for large enough values of n. □

Now we are in conditions to state the main theorem for Recurring Selection which can be proved with a combination of the preceeding lemmas and Theorems 4 and 8. As before, this theorem can be extended to Recurring Reception and Recurring Transmission.

Theorem 11. *Given a one-hop Radio Network with n nodes, where $k \leq n^{1/6 \log\log n}$ nodes are activated perhaps at different times, Primed Selection using a compact set of periods solves the Recurring Selection problem with optimal message complexity k and delay in $O(kn)$ which is 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

3.1 Reducing the Delay using 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 introducing a pre-processing phase in which nodes self-assign those primes appropriately. Given that in this protocol it is necessary to maintain two sets of k primes, we relax the Weak Sensor Model assuming that the memory size of each node is bounded only by $O(k + \log n)$ bits. We further assume that nodes are deployed densely enough so that if we reduce the radius of transmission by a constant factor 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 $k \leq n$, 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, hence, we do nothing.

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 a radius of transmission of $r/2$ 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 in $O(kn \log n)$. 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 the correctnes of Algorithm 1, we have to prove that every node becomes decided and that no pair of neighboring nodes choose the same prime.

Lemma 12. *Given any node u that becomes decided in the time slot t, the 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 that 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 6, v must receive from u within T steps. Then, v must have been reset in the interval $[t - T, t]$, which is a contradiction. □

Algorithm 1: Primed Selection with pre-processing.

```
for each node with assigned prime number p(i) do
    initialize a used-small-primes set to empty;
    initialize local counter to 0;
    while counter < 2T do
        if a bigger counter is received then
            reset local counter to 0;
        if a small prime is received then
            update the local used-small-primes set;
        transmit counter with period p(i) and radius r;
        increase counter;
    end
    choose an available small prime p_j;
    while true do
        transmit p_j with period p(i) and radius r;
        transmit the message with period p_j and radius r/2;
        if in a time step these transmissions coincide then
            do not transmit;
    end
end
```

Theorem 13. *Given a Sensor Network with n nodes, where the maximum degree is $k-1<n$, if nodes run Algorithm 1, 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 12 and Theorem 6. 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, after a node becomes decided, it does not reset the counter of any other node. Applying the same argument recursively the claim follows. □

Theorem 14. *Given a Sensor Network with n nodes, where the maximum degree is $k-1<n$, after the pre-processing, the delay of Algorithm 1 is $O(k^2 \log k)$ and the message complexity is k.*

Proof. As in Theorem 6. □

4 Acknowledgements

We thank Dariusz Kowalski for useful discussions. This research was supported in part by Comunidad de Madrid grant S-0505/TIC/0285; Spanish MEC, grant TIN2005-09198-C02-01; DIMACS, Center for Discrete Mathematics & Theoretical Computer Science, grant NSF CCR 00-87022; NSF EIA 02-05116; Alfred P. Sloan Foundation 99-10-8; and Universidad de Chile, Mecesup fellowship and Anillo en Redes.

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] A. D. 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*, 2003.

[3] 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*, 2001.

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

[5] 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*, 2000.

[6] 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*, 2004.

[7] 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.

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

[11] 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.

[12] 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*, 2007. To appear.

[13] M. Farach-Colton and M. A. Mosteiro. Sensor network gossiping or how to break the broadcast lower bound. Manuscript, 2007.

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

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

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

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

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

[19] 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.

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