Deterministic Blind Radio Networks

Artur Czumaj
University of Warwick, Coventry, UK
A.Czumaj@warwick.ac.uk

Peter Davies
University of Warwick, Coventry, UK
P.W.Davies@warwick.ac.uk

Abstract

Ad-hoc radio networks and multiple access channels are classical and well-studied models of distributed systems, with a large body of literature on deterministic algorithms for fundamental communications primitives such as broadcasting and wake-up. However, almost all of these algorithms assume knowledge of the number of participating nodes and the range of possible IDs, and often make the further assumption that the latter is linear in the former. These are very strong assumptions for models which were designed to capture networks of weak devices organized in an ad-hoc manner. It was believed that without this knowledge, deterministic algorithms must necessarily be much less efficient.

In this paper we address this fundamental question and show that this is not the case. We present *deterministic* algorithms for *blind* networks (in which nodes know only their own IDs), which match or nearly match the running times of the fastest algorithms which assume network knowledge (and even surpass the previous fastest algorithms which assume parameter knowledge but not small labels).

Specifically, in multiple access channels with k participating nodes and IDs up to L, we give a wake-up algorithm requiring $O(\frac{k \log L \log k}{\log \log k})$ time, improving dramatically over the $O(L^3 \log^3 L)$ time algorithm of De Marco et al. (2007), and a broadcasting algorithm requiring $O(k \log L \log \log k)$ time, improving over the $O(L)$ time algorithm of Gąsieniec et al. (2001) in most circumstances. Furthermore, we show how these same algorithms apply directly to multi-hop radio networks, achieving even larger running time improvements.

2012 ACM Subject Classification Theory of Computation → Design and Analysis of Algorithms → Distributed Algorithms; Networks → Network Algorithms

Keywords and phrases Broadcasting; Deterministic Algorithms; Radio Networks

Digital Object Identifier 10.4230/LIPIcs.DISC.2018.15

Funding Research partially supported by the Centre for Discrete Mathematics and its Applications (DIMAP), by EPSRC award EP/D063191/1, and by EPSRC award EP/N011163/1.

1 Introduction

In this paper we address the fundamental question in distributed computing of whether basic communication primitives can be efficiently performed in networks in which the participating nodes have no knowledge about the network structure. Our focus is on *deterministic* algorithms.

1.1 Models and problems

We consider the two classical, and related, models of distributed communication: *multiple access channels* (cf. [19, 28]) and *ad-hoc multi-hop radio networks* (cf. [2, 8, 14, 27]).

32nd International Symposium on Distributed Computing (DISC 2018).
Editors: Ulrich Schmid and Josef Widder; Article No. 15; pp. 15:1–15:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1.1.1 Multiple access channels

A set of k nodes, with unique identifiers (IDs) from $\{1, \ldots, L\}$, share a communication channel. Time is divided into discrete steps, and in every step each node chooses to either transmit a message to the channel or listen for messages. A transmission is only successful if exactly one node chooses to transmit in a given time-step; otherwise all nodes hear silence.

1.1.2 Ad-hoc multi-hop radio networks

The network is modeled by a *directed* graph $\mathfrak{N} = (V, E)$, with $|V| = n$, where nodes correspond to transmitter-receiver stations. The nodes have unique identifiers from $\{1, \ldots, L\}$. A directed edge $(v, u) \in E$ means that node v can send a message directly to node u. To make propagation of information feasible, we assume that every node in V is reachable in $\mathfrak{N}$ from any other. Time is divided into discrete steps, and in every step each node chooses to either transmit a message to all neighbors or listen for messages. A listening node only hears a transmission if exactly one neighbor transmitted; otherwise it hears silence.

It can be seen that multiple access channels are equivalent to *single-hop radio networks* (that is, radio networks in which the underlying graph is a clique).

1.1.3 Node knowledge

We study *blind* versions of these models, by which we mean that the minimum possible assumptions about node knowledge are made (and this is where our work differs most significantly from previous work): we assume nodes do not know the parameters k, L, or n, or any upper bounds thereof. In accordance with the standard model of ad-hoc radio networks (for more elaborate discussion about the model, see, e.g., [1, 2, 6, 9, 10, 16, 21, 23, 27]), we also make the assumption that a node does not have any prior knowledge about the topology of the network, its in-degree and out-degree, or the set of its neighbors. In our setting, *the only prior knowledge nodes have is their own unique ID.*

1.1.4 Tasks

In both models we consider the fundamental communication tasks of *broadcasting* (see, e.g., the survey [27] and the references therein) and *wake-up* (cf. [3, 8, 15]).

In the task of *wake-up*, nodes begin in a dormant state, and some non-empty subset of nodes spontaneously 'wake up' at arbitrary (adversarially chosen) time-steps. Nodes are also woken up if they receive messages. Nodes cannot participate (by transmitting) until they are woken up, and our goal is to ensure that eventually all nodes are awake. We assume nodes have access only to a *local clock*: they can count the number of time-steps since they woke up, but there is no global awareness of an absolute time-step number.

The task of *broadcasting* is usually described as follows: one node begins with a message, and it must inform all other nodes of this message via transmissions. However, to enable our results to transfer from multiple access channels (single-hop radio networks) to multi-hop radio networks, we will instead use broadcasting to refer to a more generalized task. Our broadcasting task will be defined similarly to wake-up, with the only difference being that nodes have access to a *global clock*, informing them of the absolute time-step number. (In multiple access channels, this task is usually also referred to as wake-up, specifying global clock access, but here we will call it broadcasting to better differentiate.)

Notice that the standard broadcasting task in radio networks is a special case of this task, in which only one node spontaneously wakes up. A global clock can be simulated by appending the current global time-step to each transmitted message (and since all message chains originate from the same source node, these time-steps will agree).

For both tasks, we wish to minimize the number of time-steps that elapse between the first node waking up, and all nodes being woken. We are not concerned with the computation performed by nodes within time-steps.

1.2 Related work

As fundamental communications primitives, the tasks of designing efficient deterministic algorithms for *broadcasting* and *wake-up* have been extensively studied for various network models for many decades.

1.2.1 Wake-up

The wake-up problem (with only local clocks) has been studied in both multiple access channels and multi-hop radio networks (often separately, though the results usually transfer directly from one to the other). It has been commonly assumed in the literature that network parameters are known, and that IDs are small ($L = n^{O(1)}$).

The first sub-quadratic deterministic wake-up protocol for radio networks was given in by Chrobak et al. [8], who introduced the concept of *radio synchronizers* to abstract the essence of the problem. They give an $O(n^{5/3} \log n)$-time protocol for the wake-up problem. Since then, there have been several improvements in running time, making use of the radio synchronizer machinery: firstly to $O(n^{3/2} \log n)$ [4], and then to $O(n \log^2 n)$ [3]. The current fastest protocol is $O(\frac{n \log^2 n}{\log \log n})$ [13]. However, without the assumption of small labels, all of these running times are increased. The algorithm of [13] as analyzed would give $O(\frac{n \log L \log(n \log L)}{\log \log(n \log L)})$ time, and similar time with k replacing n in multiple access channels. All of these algorithms, like those we present here, are non-explicit.

There has been some work on wake-up in multiple access channels without knowledge of network parameters: firstly an $O(L^4 \log^5 L)$ algorithm [15], and then an improvement to $O(L^3 \log^3 L)$ [26]. It was believed that this algorithms in this setting were necessarily much slower than those for when parameters were known; for example, [26] states "a crucial assumption is whether the processors using the shared channel are aware of the total number n of processors sharing the channel, or some polynomially related upper bound to such number. When such number n is known, much faster algorithms are possible."

There are no direct results for wake-up in radio networks with unknown parameters, but the algorithm of [26] can be applied to give $O(nL^3 \log^3 L)$ time.

We note that randomized algorithms for wake-up have also been studied, both with and without parameter knowledge; see [15, 19].

1.2.2 Broadcasting

Broadcasting is possibly the most studied problem in radio networks, and has a wealth of literature in various settings. For the model studied in this paper, *directed* radio networks with *unknown structure* and *without collision detection*, the first sub-quadratic *deterministic* broadcasting algorithm was proposed by Chlebus et al. [6], who gave an $O(n^{11/6})$-time broadcasting algorithm. After several small improvements (cf. [7, 25]), Chrobak et al. [9] designed an almost optimal algorithm that ns the task in $O(n \log^2 n)$ time, the first to be only a poly-logarithmic factor away from linear dependency. Kowalski and Pelc [21] improved this bound to obtain an algorithm of complexity $O(n \log n \log D)$ and Czumaj and Rytter [14] gave a broadcasting algorithm running in time $O(n \log^2 D)$. Here D is the eccentricity of the network, i.e., the distance between the furthest pair of nodes. De Marco [24] designed an algorithm that completes broadcasting in $O(n \log n \log \log n)$ time steps, beating [14] for general graphs. Finally, the $O(n \log D \log \log D)$ algorithm of [13] came within a log-logarithmic factor of the $\Omega(n \log D)$ lower bound [10]. Again, however, these results generally assume *small node labels* ($L = O(n)$, though some of the earlier results only require $L = O(n^c)$ for some constant c), and their running time results do not hold otherwise. The situation where node labels can be large is less well-studied, though it is easy to see that the algorithm of [9] still works, requiring $O(n \log^2 L)$ time. In multiple access channels, a $O(k \log \frac{L}{k})$ time algorithm exists [10]. Again, all of these algorithms are, like those presented here, non-explicit.

All of these results also *intrinsically require parameter knowledge.* Without knowledge of n, L, k, or D, the fastest algorithm known is the $O(L)$ time algorithm of [15] for multiple access channels. This algorithm is explicit, but has the strong added restriction that the first node wakes up at global time-step 0. It also does not transfer to multi-hop radio networks, so the best running time therein is the $O(DL^3 \log^3 L)$ given by the algorithm of [26]. Concurrently with this work, randomized algorithms for broadcasting without parameter knowledge are presented in [12], achieving a nearly optimal running time of $O(D \log \frac{n}{D} \log^2 \log \frac{n}{D} + \log^2 n)$ in the model we study here (that is, the model without collision detection).

Broadcasting, as a fundamental communication primitive, has been also studied in various related models, including undirected networks, randomized broadcasting protocols, models with collision detection, and models in which the entire network structure is known. For example, if the underlying network is undirected, then an $O(n \log D)$-time algorithm due to Kowalski [20] exists. If spontaneous transmissions are allowed and a global clock available, then deterministic broadcast can be performed in $O(L)$ time in undirected networks [6]. Randomized broadcasting has been also extensively studied, and in a seminal paper, Bar-Yehuda et al. [2] designed an almost optimal broadcasting algorithm achieving the running time of $O((D + \log n) \cdot \log n)$. This bound has been later improved by Czumaj and Rytter [14], and independently Kowalski and Pelc [22], who gave optimal randomized broadcasting algorithms that complete the task in $O(D \log \frac{n}{D} + \log^2 n)$ time with high probability, matching a known lower bound from [23]. Haeupler and Wajc [17] improved this bound for undirected networks in the model that allows spontaneous transmissions and designed an algorithm that completes broadcasting in time $O\left(\frac{D \log n \log \log n}{\log D} + \log^{O(1)} n\right)$ with high probability, improved to $O\left(\frac{D \log n}{\log D} + \log^{O(1)} n\right)$ in [11]. In the model with collision detection for undirected networks, an $O(D + \log^6 n)$-time randomized algorithm due to Ghaffari et al. [16] is the first to exploit collisions and surpass the algorithms (and lower bound) for broadcasting without collision detection.

For more details about broadcasting algorithms in various models, see e.g., [11, 14, 16, 20, 27] and the references therein.

1.3 New results

We present algorithms for the fundamental tasks of broadcasting and wake-up in multiple access channels (single-hop radio networks) and multi-hop radio networks which require no knowledge of network parameters: nodes need know only their own unique ID.

Our wake-up algorithm takes $O(\frac{k \log L \log k}{\log \log k})$ time in multiple access channels and $O(\frac{n \log L \log n}{\log \log n})$ time in multi-hop radio networks, improving dramatically over the previous best $O(L^3 \log^3 L)$ and $O(DL^3 \log^3 L)$ respective running times of [26] (recall that $k \leq n \leq L$).

This is particularly significant in the case of large labels, since dependency on L has been improved from cubic to logarithmic. Furthermore, our running time matches the $O(\frac{n \log L \log n}{\log \log n})$ time of [13], the fastest algorithm with parameter knowledge and small node labels.

Our broadcasting algorithm takes $O(k \log L \log \log k)$ time in multiple access channels and $O(n \log L \log \log n)$ time in multi-hop radio networks. This improves over the previous best $O(L)$ multiple access channel bound [15] in most cases. In radio networks the improvement is even more pronounced, beating not only the $O(DL^3 \log^3 L)$ result of [26] but also the $O(n \log^2 L)$-time algorithm of [9], which was the fastest result for large labels even when network parameters are known. When labels are small (i.e., $L = n^{O(1)}$), our result matches the best running time for known parameters ($O(n \log D \log \log D)$ from [13]) for networks of polynomial eccentricity.

We believe the primary value of our work is in challenging the *long-standing assumption that parameter knowledge is necessary for efficient deterministic algorithms* in radio networks and multiple access channels. We show that in fact, deterministic algorithms which do not assume this knowledge can reach the fastest running times for those that do.

1.4 Previous approaches

Almost all deterministic broadcasting protocols with sub-quadratic complexity (that is, since [6]) have relied on the concept of *selective families* (or some similar variant thereof, such as selectors). These are families of sets for which one can guarantee that any subset of $[k] := \{1, 2, \ldots, k\}$ below a certain size has an intersection of size exactly 1 with some member of the family [6]. They are useful in the context of radio networks because if the members of the family are interpreted to be the set of nodes which are allowed to transmit in a particular time-step, then after going through each member, any node with a participating in-neighbor and an in-neighborhood smaller than the size threshold will be informed. Most of the recent improvements in broadcasting time have been due to a combination of proving smaller selective families exist, and finding more efficient ways to apply them (i.e., choosing which size of family to apply at which time) [6, 7, 9, 14].

Applying such constructs requires coordination between nodes, which relies on a global clock, making them unsuitable for wake-up. To tackle this issue, Chrobak et al. [8] introduced the concept of *radio synchronizers*. These are a development of selective families which allow nodes to begin their behavior at different times. A further extension to *universal synchronizers* in [4] allowed effectiveness across all in-neighborhood sizes.

Another similar extension of selective families came in 2010 with a paper by De Marco [24], which used a *transmission matrix* to schedule node transmissions for broadcasting. Like radio synchronizers, the application of this matrix allowed nodes to begin their own transmission sequence at any time, and still provided a 'selective' property that guaranteed broadcasting progress. The synchronization afforded by the assumption of a global clock allowed this method to beat the time bounds given by radio synchronizers (and previous broadcasting algorithms using selective families).

The proofs of *existence* for selective families, synchronizers, and transmission matrices follow similar lines: a probabilistic candidate object is generated by deciding on each element independently at random with certain carefully chosen probabilities, and then it is proven that the candidate satisfies the desired properties with positive probability, and so *such an object must exist.* These types of proofs are all *non-constructive* (and therefore all resulting algorithms non-explicit; cf. [5, 18] for an explicit construction of selective families with significantly weaker size bounds).

In contrast, results on multiple access channels without parameter knowledge (notably

[15, 26]) have not used these types of combinatorial objects, and instead rely on some results from number theory. The algorithm of [26], for instance, is to have nodes transmit periodically, a node with ID v waiting p_v steps between transmissions, where p_v is the v^{th} smallest prime number. A number-theoretic result is then employed to show that eventually one node will transmit alone. As a result, these algorithms have the advantage of being explicit, but the disadvantage of slower running times.

1.5 Novel approach

We aim to apply the approach of using combinatorial objects proven by the probabilistic method to the setting where network parameters are not known. One way to do this would be to apply selectors (for example) of continually increasing size, until one succeeds. However, since there are two parameters which must meet the correct values for a successful application (k and L in the case of medium access channels), running times for this approach are poor. Instead, we define, and prove the existence of, *a single, infinite combinatorial object, which can accommodate all possible values of parameters at the same time.*

Another difference is that for all previous works using selective families or variants thereof, the candidate object has been generated with the same sequence of probabilities for each node. Here, however, in order to achieve good running times we need to have these probabilities depending on the node ID. In essence, this means that nodes effectively use their own ID as an estimate of the maximum ID in the network.

1.6 A note on non-explicitness

As mentioned, almost all deterministic broadcasting protocols with sub-quadratic complexity have relied on selective families or variants thereof, and have been non-explicit results. Our work here is also non-explicit, but while this is standard for deterministic radio network algorithms, it may present more of an issue here, since our combinatorial structures are infinite. It is not clear how the protocols we present could be performed by devices with bounded memory, and as such this work is more of a proof-of-concept than a practical algorithm. However, it is possible that our method could be adapted so that nodes' behavior could be generated by a finite-size (i.e., a function of ID) program; this would be a natural and interesting extension to our work, and would overcome the problem.

Another issue which would remain is that nodes must perform the protocol indefinitely, and never become aware that broadcasting has been successfully completed. However, this is unavoidable in the model: Chlebus et al. [6] prove that *acknowledged* broadcasting without parameter knowledge is impossible.

2 Combinatorial objects

In this section we present the two combinatorial objects that we wish to use in our algorithms: *unbounded universal synchronizers* and *unbounded transmission schedules*. After defining them in Sections 2.1 and 2.2, we present their main properties in Theorems 3 and 12, and then show how to apply them to obtain new deterministic algorithms for wake-up and broadcasting in multiple access channels and in radio networks (Theorems 19, 20, 22, 23).

2.1 Unbounded universal synchronizers

For the task of wake-up, i.e., in the absence of a global clock, we will define an object called an **unbounded universal synchronizer** for use in our algorithm.

We begin by defining the sets of circumstances our algorithm must account for:

▶ **Definition 1.** An (r,k)**-instance** X is a set K of k nodes with

$$\sum_{v \in K} \log v = r$$

and wake-up function $\omega : K \to \mathbb{N}$.

(By using v as an integer here, we are abusing notation to mean the ID of node v.)

Here r is the main parameter we will use to quantify how 'large' our input instance is. By using the sum of logarithms of IDs (which approximates the total number of bits needed to write all IDs in use), we capture both the number of participating nodes and the length of IDs in a single parameter. We require r to be an integer, so we round down accordingly, but we omit floor functions for clarity since they do not affect the asymptotic result.

The *wake-up function* ω maps each node to the time-step it wakes up (either spontaneously or by receiving a transmission) when our algorithm is run on this instance. This means that the wake-up function depends on the algorithm, but there is no circular dependency: whether nodes wake-up in time-step j only depends on the algorithm's behavior in previous time-steps, and the algorithm's behavior at time-step j only depends on the wake-up function up to j. We will also extend ω to sets of nodes in the instance by $\omega(K) := \min_{v \in K} \omega(v)$.

We now define the combinatorial object that will be the basis of our algorithm:

▶ **Definition 2.** For a function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, an **unbounded universal synchronizer of delay** g is a function $\mathcal{S} : \mathbb{N} \to \{0,1\}^{\mathbb{N}}$ such that for any (r,k)-instance, there is some time-step $j \le \omega(K) + g(r,k)$ with $\sum_{v \in K} \mathcal{S}(v)_{j-\omega(v)} = 1$.

The *unbounded universal synchronizer* S is a function mapping node IDs to a sequence of 0s and 1s, which tell nodes when to listen and transmit respectively. The function g, which we will call the *delay function*, tells us how many time-steps we must wait before a successful transmission is guaranteed, so this is what we want to asymptotically minimize.

We will apply this object to perform wake-up as follows: each node v transmits a message in time-step j (with its time-step count starting upon waking up) iff $S(v)_j = 1$. Then, the property guarantees that at some time-step j within $g(r,k)$ time-steps of the first node waking up, any (r,k)-instance will have a successful transmission. We call this S *'hitting' the* (r,k)*-instance at time-step* j. So, our aim is to show the existence of such an object, with g growing as slowly as possible.

Our main technical result in this section is the following:

▶ **Theorem 3.** *There exists an* ***unbounded universal synchronizer of delay*** g*, where* $g(r,k) = O\left(\frac{r \log k}{\log \log k}\right)$.

Our approach to proving Theorem 3 will be to randomly generate a candidate synchronizer, and then prove that with positive probability it does indeed satisfy the required property. Then, for this to be the case, at least one such object must exist.

Our candidate $S : \mathbb{N} \to \{0,1\}^{\mathbb{N}}$ will be generated by independently choosing each $S(v)_j$ to be **1** with probability $\frac{c \log v}{j + 2c \log v}$ and **0** otherwise, where c is some sufficiently large constant to be chosen later.

While S is drawn from an uncountable set, we will only be concerned with events that depend upon a finite portion of it, and countable unions and intersections thereof. Therefore, we can use as our underlying σ-algebra that generated by the set of all events $E_{v,j} = \{S : S(v)_j = 1\}$, where $v, j \in \mathbb{N}$, with the corresponding probabilities $\mathbb{P}[E_{v,j}] = \frac{c \log v}{j + 2c \log v}$.

We set delay function $g(r,k) = \frac{c^2 r \log k}{\log\log k}$.

To simplify our task, we begin with some useful observations:

First we note that since we only care about the asymptotic behavior of g, we can assume that r is larger than a sufficiently large constant.

We also note that we need not consider all (r,k)-instances. For a given (r,k)-instance and time-step j, let K_j be the set of nodes woken up by time j (with $k_j := |K_j|$), and r_j be defined as r but restricted to the nodes in K_j. Let t be the earliest time-step such that $t > g(r_t, k_t)$, and curtail the (r,k)-instance to the corresponding (r_t,k_t)-instance of nodes in K_t. It is easy to see that if we hit all curtailed (r_t,k_t)-instances within $g(r_t,k_t)$ time, we must hit all (r,k)-instances within $g(r,k)$ time, so henceforth we will only consider curtailed instances (i.e., we can assume that $j \le g(r_j,k_j)$ for all $j < g(r,k)$).

Finally, we observe that, since nodes' behavior is not dependent on the global clock, we can shift all (r,k)-instances to begin at time-step 0.

To analyze the probability of hitting (r,k)-instances, define the *load* of a time-step $f(j)$ to be the expected number of transmissions in that time-step:

▶ **Definition 4.** For a fixed (r,k)-instance, the **load** $f(j)$ **of a time-step** j is defined as

$$\sum_{v\in K_j} \mathbb{P}\left[v \text{ transmits}\right] = \sum_{v\in K_j} \frac{c\log v}{j-\omega(v)+2c\log v} \ .$$

We now seek to bound the load from above and below, since when the load is close to constant we have a good chance of hitting.

▶ **Lemma 5.** *All time-steps $j \le g(r,k)$ have $f(j) \ge \frac{\log\log k}{2c\log k}$.*

Proof. Fix a time-step $j \le g(r,k)$, let K_j be the set of nodes awake by time-step j, and let $k_j = |K_j|$ and $r_j = \sum_{v\in K_j} \log v$, analogous to r and k. If $k_j = k$, then

$$f(j) \ge \sum_{v\in K} \frac{c\log v}{j+2c\log v} \ge \frac{cr}{j+2cr} \ge \frac{cr}{\frac{2c^2 r\log k}{\log\log k}} \ge \frac{\log\log k}{2c\log k} \ .$$

If $k_j < k$, then due to our curtailing of instances, we have $j \le g(r_j,k_j)$. So,

$$f(j) \ge \sum_{v\in K_j} \frac{c\log v}{j+2c\log v} \ge \frac{cr_j}{j+2cr_j} \ge \frac{cr_j}{\frac{2c^2 r_j\log k_j}{\log\log k_j}} \ge \frac{\log\log k_j}{2c\log k_j} \ge \frac{\log\log k}{2c\log k} \ .$$ ◀

Having bounded load from below, we also seek to bound it from above. Unfortunately, the load in any particular time-step can be very high, but we can get a good bound on at least a constant fraction of the columns.

▶ **Lemma 6.** *Let F denote the set of time-steps $j \le g(r,k)$ such that $\frac{\log\log k}{2c\log k} \le f(j) \le \frac{\log\log k}{3}$. Then $|F| \ge \frac{cr\log k}{2\log\log k}$.*

Proof. The total load over all time-steps can be bounded as follows:

$$\sum_{j\le g(r,k)} f(j) = \sum_{j\le g(r,k)}\sum_{v\in K_j} \frac{c\log v}{j-\omega(v)+2c\log v} \le \sum_{v\in K}\sum_{\omega(v)<j\le g(r,k)} \frac{c\log v}{j-\omega(v)+2c\log v}$$

$$\le \sum_{v\in K} c\log v \sum_{j\le g(r,k)} \frac{1}{j+2c\log v} \le \sum_{v\in K} c\log v \ln\frac{2g(r,k)}{4c\log v} \ .$$

Let $K_i = \{v \in K : \frac{r}{k \cdot 2^i} \le \log v < \frac{r}{k \cdot 2^{i-1}}\}$, for $i \ge 1$, and $K' = \{v \in K : \log v \ge \frac{r}{k}\}$

If $\sum_{v \in K_i} \log v > \frac{r}{2^i}$ then $r < 2^i \sum_{v \in K_i} \log v \le 2^i \sum_{v \in K_i} \frac{r}{k \cdot 2^i} \le r$. This gives a contradiction, so we must have $\sum_{v \in K_i} \log v \le \frac{r}{2^i}$. Then,

$$
\begin{aligned}
\sum_{j \le g(r,k)} f(j) &\le \sum_{v \in K} c \log v \ln \frac{2g(r,k)}{4c \log v} \le \sum_{i \ge 1} \sum_{v \in K_i} c \log v \ln \frac{g(r,k)}{2c \log v} + \sum_{v \in K'} c \log v \ln \frac{g(r,k)}{2c \log v} \\
&\le \sum_{i \ge 1} \sum_{v \in K_i} c \log v \ln \frac{g(r,k)}{2c \frac{r}{k \cdot 2^i}} + \sum_{v \in K'} c \log v \ln \frac{g(r,k)}{2c \frac{r}{k}} \\
&= \sum_{i \ge 1} \sum_{v \in K_i} c \log v \ln \frac{ck 2^{i-1} \log k}{\log \log k} + \sum_{v \in K'} c \log v \ln \frac{ck \log k}{2 \log \log k} \\
&\le \sum_{i \ge 1} cr 2^{-i} (2 \ln k + (i-1) \ln 2) + 2cr \ln k \le 5cr \ln k \le 8cr \log k \ .
\end{aligned}
$$

Therefore, at most $\frac{24cr \log k}{\log \log k}$ time-steps have load higher than $\frac{\log \log k}{3}$. Since by Lemma 5 all time-steps have load at least $\frac{\log \log k}{2c \log k}$,we have $|F| \ge g(r,k) - \frac{24cr \log k}{\log \log k} \ge \frac{c^2 r \log k}{2 \log \log k}$ (provided we pick $c \ge 7$). ◀

Now that we have bounded load, we show how it relates to hitting probability. The following lemma, or variants thereof, has been used in several previous works such as [24], but we prove it here for completeness.

▶ **Lemma 7.** *Let x_i, $i \in [n]$, be independent $\{0,1\}$-valued random variables with $\mathbb{P}[x_i = 1] \le \frac{1}{2}$, and let $f = \sum_{i \in [n]} \mathbb{P}[x_i = 1]$. Then $\mathbb{P}\left[\sum_{i \in [n]} x_i = 1\right] \ge f 4^{-f}$.*

Proof.

$$
\begin{aligned}
\mathbb{P}\left[\sum_{i \in [n]} x_i = 1\right] &= \sum_{j \in [n]} \mathbb{P}[x_j = 1 \wedge \forall_{i \ne j}\ x_i = 0] \ge \sum_{j \in [n]} \mathbb{P}[x_j = 1] \cdot \mathbb{P}[\forall_i\ x_i = 0] \\
&\ge f \cdot \mathbb{P}[\forall_i\ x_i = 0] = f \cdot \prod_{i \in [n]} (1 - \mathbb{P}[x_i = 1]) \ge f \cdot \prod_{i \in [n]} 4^{-\mathbb{P}[x_i = 1]} \\
&= f \cdot 4^{-\sum_{i \in [n]} \mathbb{P}[x_i = 1]} = f 4^{-f} \ .
\end{aligned}
$$

◀

We can use Lemma 7 to show that the probability that we hit on our 'good' time-steps (those in F) is high:

▶ **Lemma 8.** *For any time-step $j \in F$, the probability that j hits is at least $\frac{\log \log k}{3c \log k}$.*

Proof. $\frac{\log \log k}{2c \log k} \le f(j) \le \frac{\log \log k}{3}$, and so $f(j) 4^{-f(j)}$ is minimized at $f(j) = \frac{\log \log k}{2c \log k}$ and is therefore at least $\frac{\log \log k}{2c \log k} 4^{-\frac{\log \log k}{2c \log k}} \ge \frac{\log \log k}{3c \log k}$. ◀

We now bound the number of possible instances we must hit:

▶ **Lemma 9.** *For any (sufficiently large) r, the number of unique (r,k)-instances is at most 2^{5r}.*

Proof. The total number of bits used in all node IDs in the instance is at most r. There are at most 2^{r+1} possible bit-strings of length at most r, and at most 2^r ways of dividing each of these into substrings (for individual IDs), giving at most 2^{2r+1} sets of node IDs. The number of possible wake-up functions $\omega : K \to \mathbb{N}$ is at most $g(r,k)^k$, since all nodes must be awake by $g(r,k)$ time or the instance would have been curtailed.

$$g(r,k)^k = 2^{k \log g(r,k)} \leq 2^{1.1k \log r} = 2^{1.1(k \log k + k \log \frac{r}{k})} \leq 2^{1.3(k \log(k^{0.9})+r)} \leq 2^{2.9r} \quad .$$

So, the total number of possible (r,k)-instances is at most $2^{2r+1+2.9r} \leq 2^{5r}$. ◀

▶ **Lemma 10.** *For any (sufficiently large) r, the probability that S does not hit all (r,k)-instances is at most 2^{-3r}*

Proof. Fix some (r,k)-instance. The probability that it is not hit within $g(k,r)$ time-steps is at most

$$\prod_{j \in F} (1 - \frac{\log \log k}{3c \log k}) \leq e^{-|F| \frac{\log \log k}{3c \log k}} \leq e^{-\frac{2}{3}cr} = 2^{-\frac{2cr}{3 \ln 2}} \quad ,$$

by Lemma 8. Hence, if we set $c = 9$, by a union bound the probability that any (r,k)-instance is not hit is at most $2^{5r} \cdot 2^{-\frac{18r}{3 \ln 2}} \leq 2^{-3r}$. ◀

We can now prove our main theorem on unbounded universal synchronizers (Theorem 3):

Proof. By Lemma 10 and a union bound over r, the probability that S does not hit all instances is at most $\sum_{r \in \mathbb{N}} 2^{-3r} < 1$. Therefore S is an unbounded universal synchronizer of delay g with non-zero probability, so such an object must exist. ◀

2.2 Unbounded transmission schedules

For the task of broadcasting, i.e., when a global clock is available, we can make use of the global clock to improve our running time. We again define an infinite combinatorial object, which we will call an **unbounded transmission schedule**. We use the same definition of (r,k)-instances as in the previous section.

▶ **Definition 11.** For a function $h : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, an **unbounded transmission schedule of delay** h is a function $T : \mathbb{N} \times \mathbb{N} \to \{0,1\}^{\mathbb{N}}$ such that $T(v, \omega(v))_j = 0$ for any $j < \omega(v)$, and for any (r,k)-instance there is some time-step $j \leq \omega(K) + h(r,k)$ with $\sum_{v \in K} T(v, \omega(v))_j = 1$.

The difference here from an unbounded universal synchronizer is that nodes now know the global time-step in which they wake up, and so their transmission patterns can depend upon it. This is the second argument of the function T. The other difference in the meaning of the definition is that the output of T now corresponds to absolute time-step numbers, rather than being relative to each node's wake-up time as for unbounded universal synchronizers. That is, the j^{th} entry of a node's output sequence tells it how it should behave in global time-step j, rather than j time-steps after it wakes up.

Our existence result for unbounded transmission schedules is the following:

▶ **Theorem 12.** *There exists an **unbounded transmission schedule of delay** h, where $h(r,k) = O(r \log \log k)$.*

Our method will again be to randomly generate a candidate unbounded transmission schedule T, and then prove that it satisfies the required property with positive probability, so some such object must exist.

Let d be a constant to be chosen later. Our candidate object T will be generated as follows: for each node v, we generate a transmission sequence $s_{v,j}$, $j \in \mathbb{N}$, with $s_{v,j}$ independently chosen to be 1 with probability $\frac{d \log v \log \log j}{j + 2d \log v \log \log j}$ and 0 otherwise.

However, these will not be our final probabilities for generating T. To make use of our global clock, we will also divide time into short *phases* during which transmission probability will decay exponentially. The lengths of these phases will be based on a function $z(j) := 2^{\lceil 1+\log\log\log j\rceil}$, i.e., $\log\log j$ rounded up to the next-plus-one power of 2. An important property to note is that for all i, $z(i)|z(i+1)$. We also generate a sequence $p_{v,j}$, $j \in \mathbb{N}$ of *phase values*, each chosen independently and uniformly at random from the real interval $[0, 1]$. These, combined with the global time-step number and current phase length, will give us our final generation probabilities.

We set $T(v, \omega(v))_j$ to equal 1 iff $s_{v,j-\omega(v)} = 1$ and $p_{v,j-\omega(v)} \leq 2^{-j \bmod z(j-\omega(v))}$.

It can then be seen that

$$\mathbb{P}\left[T(v, \omega(v))_j = 1\right] = \frac{d \log v \log\log(j - \omega(v))}{(j - \omega(v) + 2d \log v \log\log(j - \omega(v)))2^{j \bmod z(j-\omega(v))}} \ .$$

The reason we split the process of random generation into two steps (using our phase values) is that now, if we shift all wake-up times in an (r, k)-instance by the same multiple of $z(x)$, then node behavior in the first x time-steps after $\omega(K)$ does not change. We will require this property when analyzing T.

Our probabilistic analysis is with respect to the σ-algebra generated by all events $E_{v,\omega(v),j} = \{T : T(v, \omega(v))_j = 1\}$, with $v, \omega(v), j \in \mathbb{N}$, and with the corresponding probabilities given above.

Let **load** $f(j)$ **of a time-step** j be again defined as the expected number of transmissions in that time-step:

$$f(j) := \sum_{v \in K_j} \frac{d \log v \log\log(j - \omega(v))}{(j - \omega(v) + 2d \log v \log\log(j - \omega(v)))2^{j \bmod z(j-\omega(v))}} \ .$$

We will set our delay function $h(r, k) = d^2 r \log\log k$.

Again we make some observations that allow us to narrow the circumstances we must consider: firstly that we can again assume that r is larger than a sufficiently large constant, and secondly that we need only look at curtailed instances (i.e., we can assume that $j - \omega(K) \leq h(r_j, k_j)$ for all $j < h(r, k)$). This time, however, we cannot shift instances to begin at time-step 0, because node behavior is dependent upon global time-step number.

We follow a similar line of proof as before, except with some extra complications in dealing with phases. We first consider only time-steps at the beginning of each phase, i.e., time-steps $\omega(K) < j \leq \omega(K) + h(r, k)$ with $j \bmod z(h(r, k)) \equiv 0$, and we will call these *basic* time-steps. Notice that for basic time-steps,

$$f(j) = \sum_{v \in K_j} \frac{d \log v \log\log(j - \omega(v))}{j - \omega(v) 2d \log v \log\log(j - \omega(v))} \ .$$

We bound the load of basic time-steps from below:

▶ **Lemma 13.** *All basic time-steps j have $f(j) \geq \frac{1}{2d}$.*

Proof. Fix a basic time-step j, let K_j be the set of nodes awake by time-step j, and let $k_j = |K_j|$ and $r_j = \sum_{v \in K_j} \log v$, analogous to r and k. If $k_j = k$, then

$$\begin{aligned} f(j) &\geq \sum_{v \in K} \frac{d \log v \log\log(j - \omega(v))}{j - \omega(v) + 2d \log v \log\log(j - \omega(v))} \geq \sum_{v \in K} \frac{d \log v \log\log h(r, k)}{h(r, k) + 2d \log v \log\log h(r, k)} \\ &\geq \sum_{v \in K} \frac{d \log v \log\log k}{2d^2 r \log\log k} \geq \frac{dr}{2d^2 r} = \frac{1}{2d} \ . \end{aligned}$$

If $k_j < k$, then due to our curtailing of instances, we have $j - \omega(K) \leq h(r_j, k_j)$. So,

$$f(j) \geq \sum_{v \in K_j} \frac{d \log v \log\log(j - \omega(v))}{j - \omega(v) + 2d \log v \log\log(j - \omega(v))} \geq \sum_{v \in K} \frac{d \log v \log\log h(r_j, k_j)}{h(r_j, k_j) + 2d \log v \log\log h(r, k)}$$

$$\geq \sum_{v \in K} \frac{d \log v \log\log k_j}{2d^2 r_j \log\log k_j} \geq \frac{dr_j}{2d^2 r_j} = \frac{1}{2d} \ . \qquad ◀$$

We next examine time-steps at the end of phases, i.e., with $\omega(K) < j \leq \omega(K) + h(r, k)$ and $j \bmod z(h(r, k)) \equiv -1$, and call these *ending* time-steps. Note that for ending time-steps,

$$f(j) = \sum_{v \in K_j} \frac{d \log v \log\log(j - \omega(v))}{(j - \omega(v) + 2d \log v \log\log(j - \omega(v)))2^{z(j - \omega(v)) - 1}} \ .$$

We bound the load of (a constant fraction of) ending time-steps from above:

▶ **Lemma 14.** *Let $\mathcal{F}$ denote the set of ending time-steps j such that $f(j) \leq 1$. Then $|\mathcal{F}| \geq \frac{d^2 r}{2}$.*

Proof. Let t be the first ending time-step. The total load over all ending time-steps can be bounded as follows:

$$\sum_{\text{ending timestep } j} f(j) \leq \sum_{i=0}^{h(r,k)/z(h(r,k))} f(t + iz(h(r, k))) \qquad \leq \sum_{i=0}^{d^2 r} f(t + iz(h(r, k))) \ .$$

Applying the definition of f, $f(t + iz(h(r, k)))$ is equal to:

$$\sum_{v \in K_{t+iz(h(r))}} \frac{d \log v \log\log(t + iz(h(r, k)) - \omega(v))2^{-z(t+iz(h(r,k)) - \omega(v)) - 1}}{(t + iz(h(r, k)) - \omega(v) + 2d \log v \log\log(t + iz(h(r, k)) - \omega(v)))} \ ,$$

which is bounded from above when $t - \omega(v) = 0$:

$$f(t + iz(h(r, k))) \leq \sum_{v \in K_{t+iz(h(r))}} \frac{d \log v \log\log(iz(h(r)))}{(iz(h(r, k)) + 2d \log v \log\log(iz(h(r, k))))2^{z(iz(h(r,k)))}}$$

$$\leq \sum_{v \in K_{t+iz(h(r,k))}} \frac{d \log v \log\log(iz(h(r, k)))}{iz(h(r, k))2^{z(iz(h(r,k)))}} \ .$$

So,

$$\sum_{\text{ending timestep } j} f(j) \leq \sum_{i=0}^{d^2 r} \sum_{v \in K_{t+iz(h(r,k))}} \frac{d \log v \log\log(iz(h(r, k)))}{iz(h(r, k))2^{z(iz(h(r,k)))}}$$

$$\leq \sum_{v \in K} \sum_{i=0}^{d^2 r} \frac{d \log v \log\log(iz(h(r, k)))}{iz(h(r, k))2^{z(iz(h(r,k)))}}$$

$$\leq \sum_{v \in K} \sum_{i=0}^{d^2 r} \frac{2d \log v \log\log(iz(h(r, k)))}{iz(h(r, k)) \log^2(iz(h(r, k)))}$$

$$\leq \sum_{v \in K} 2d \log v \sum_{i=0}^{\infty} \frac{\log\log(iz(h(r, k)))}{iz(h(r, k)) \log^2(iz(h(r, k)))} \leq 10dr \ .$$

Here the last inequality follows since the second sum converges to a constant less than 5, which can be seen by inspection of the first three terms and using the integral bound $\int_2^\infty \frac{\log\log x}{x\log^2 x} < 2$

Therefore, at most $10dr$ ending time-steps have load higher than 1, and so at least $d^2r - 10dr \geq \frac{d^2r}{2}$ (provided we set $d \geq 5$) ending time-steps have $f(j) \leq 1$. ◀

We can use Lemma 14 to show that we have sufficiently many time-steps with load within the interval $[\frac{1}{2d}, 1]$:

▶ **Lemma 15.** *Let $\mathcal{F}$ be the set of time-steps $\omega(K) < j \leq \omega(K) + h(r,k)$ with $\frac{1}{2d} \leq f(j) \leq 1$. Then $|\mathcal{F}| \geq \frac{d^2r}{2}$.*

Proof. It can be seen that load decreases by at most a multiplicative factor of 3 between consecutive time-steps (since the contribution of each node decreases by at most a factor of 3). So, since every base time-step has load at least $\frac{1}{2d}$, for every ending timestep j with $f(j) \leq 1$, there is some j' in the preceding phase with $\frac{1}{2d} \leq f(j') \leq 1$. ◀

Since these time-steps have constant load, they have constant probability of hitting:

▶ **Lemma 16.** *For any time-step $j \in \mathcal{F}$, the probability that j hits is at least $\frac{1}{3d}$.*

Proof. By Lemma 7, the probability that j hits is at least $f(j)4^{-f(j)}$. This is minimized over the range $[\frac{1}{2d}, 1]$ at $f(j) = \frac{1}{2d}$ and is therefore at least $\frac{4^{-\frac{1}{2d}}}{2d} \geq \frac{1}{3d}$. ◀

We now need to know how many unique (r,k)-instances we must hit. Since we are only concerned with the first $h(r,k)$ time-steps after the first node wakes up, we need only consider (r,k)-instances which are unique within this time range.

▶ **Lemma 17.** *For any (sufficiently large) r, the number of unique (up to the first $h(r,k)$ steps after activation) (r,k)-instances is at most 2^{5r}.*

Proof. As before (in Lemma 9) there are at most 2^{2r+1} sets of node IDs. The number of possible wake-up functions $\omega : K \to \mathbb{N}$ for a fixed $\omega(K)$ is again at most $h(r,k)^k$, and though we are using a different delay function to the previous section, the calculations used to prove Lemma 9 still hold.

$$h(r,k)^k = 2^{k\log h(r,k)} \leq 2^{1.1k\log r} = 2^{1.1(k\log k + k\log\frac{r}{k})} \leq 2^{1.3(k\log(k^{0.9})+r)} \leq 2^{2.9r} \quad .$$

However, now our object does depend on $\omega(K)$, though as we noted we can shift all activation times by a multiple of $z(h(r,k))$ and behavior during the time-steps we analyze is unchanged. So our total number of instances to consider is multiplied by $z(h(r,k))$, and is upper bounded by $2^{2r+1+2.9r}z(h(r,k)) \leq 2^{5r}$. ◀

▶ **Lemma 18.** *For any (sufficiently large) r, the probability that T does not hit all (r,k)-instances is at most 2^{-3r}.*

Proof. Fix some (r,k)-instance. The probability that it is not hit within $h(r,k)$ time-steps is at most

$$\prod_{j\in\mathcal{F}}(1-\frac{1}{3d}) \leq e^{-\frac{|\mathcal{F}|}{3d}} \leq e^{-\frac{dr}{6}} = 2^{-\frac{dr}{6\ln 2}} \quad .$$

Hence, if we set $d = 34$, by a union bound the probability that any (r,k)-instance is not hit is at most $2^{5r} \cdot 2^{-\frac{34r}{6\ln 2}} \leq 2^{-3r}$. ◀

We can now prove our main theorem on unbounded transmission schedules (Theorem 12):

Proof. By Lemma 18 and a union bound over r, the probability that T does not hit all instances is at most $\sum_{r\in\mathbb{N}} 2^{-3r} < 1$. Therefore T is an unbounded transmission schedule of delay h with non-zero probability, so such an object must exist. ◀

3 Algorithms for multiple access channels

Armed with our combinatorial objects, our algorithms are now extremely simple, and are the same for multiple access channels as for multi-hop radio networks.

Let S be an unbounded universal synchronizer of delay g, where $g(r,k) = O\left(\frac{r\log k}{\log\log k}\right)$, and T be an unbounded transmission schedule of delay h, where $h(r,k) = O(r\log\log k)$.

Our algorithms are simply applications of these objects.

Algorithm 1 Wake-up at a node v

> **for** j from 1 to ∞, in time-step $\omega(v)+j$, **do**
> v transmits iff $S(v)_j = 1$
> **end for**

▶ **Theorem 19.** *Algorithm 1 performs wake-up in multiple access channels in time $O\left(\frac{k\log L\log k}{\log\log k}\right)$, without knowledge of k or L.*

Proof. By the definition of an unbounded universal synchronizer, there is some time-step within

$$g(r,k) = O\left(\frac{r\log k}{\log\log k}\right) = O\left(\frac{k\log L\log k}{\log\log k}\right)$$

time-steps of the first activation in which only one node transmits, and this completes wake-up. ◀

Algorithm 2 Broadcasting at a node v

> **for** j from 1 to ∞, in time-step j, **do**
> v transmits iff $T(v,\omega(v))_j = 1$
> **end for**

▶ **Theorem 20.** *Algorithm 2 performs broadcasting in multiple access channels in time $O(k\log L\log\log k)$, without knowledge of k or L.*

Proof. By the definition of an unbounded transmission schedule, there is some time-step within $h(r,k) = O(r\log\log k) = O(k\log L\log\log k)$ time-steps of the first activation in which only one node transmits, and this completes broadcasting. ◀

4 Algorithms for radio networks

To see how our results on multiple access channels (Theorems 19 and 20) transfer directly to multi-hop radio networks, we apply the analysis method of [13] for radio network protocols. The idea is that we fix a shortest path $p = (p_0, p_1, \ldots p_d)$ from some *source* node to some *target* node, and then classify all nodes into *layers* depending on the furthest node along the path to which they are an in-neighbor, i.e., layer $L_i = \{v : \max j \text{ such that } (v, p_j) \in E = i\}$. We wish to quantify how long a layer can remain *leading*, that is, the furthest layer to contain awake nodes. The key point is that we can regard these layers as multiple access channels: though they are not necessarily cliques, all we need is for a single node in the layer to transmit and then the layer ceases to be leading. Once the final layer ceases to be leading, the target node must be informed, and since this node was chosen arbitrarily we obtain a time-bound for informing the entire network. Thereby the problem is reduced to a sequence of at most D single-hop instances, whose sizes sum to at most n. For full details of this analysis method see [13].

Therefore we can use the following lemma from [13] (paraphrased to fit our notation) to analyze how our algorithms perform in radio networks.

▶ **Lemma 21.** ***(Lemma 10 of [13])*** *Let $X : [n] \to \mathbb{N}$ be a non-decreasing function, and define $Y(n)$ to be the supremum of the function $\sum_{i=1}^{n} X(\ell_i)$, where non-negative integers ℓ_i satisfy the constraint $\sum_{i=1}^{n} \ell_i \le n$. If a broadcast or wake-up protocol ensures that any layer of size ℓ remains leading for no more than $X(\ell)$ time-steps, then all nodes wake up within $Y(n)$ time-steps.*

▶ **Theorem 22.** *Algorithm 1 performs wake-up in radio networks in time $O(\frac{n \log L \log n}{\log \log n})$, without knowledge of n or L.*

Proof. By Theorem 19, any layer of size ℓ remains leading for no more than $X(\ell)$ time-steps, where $X(\ell) = O(\frac{\ell \log L \log \ell}{\log \log \ell})$. $Y(n, h)$ is then maximized by setting $\ell_1 = n$ and $\ell_i = 0$ for every $i > 1$. So, by Lemma 21, wake-up is performed for the entire radio network in $O(\frac{n \log L \log n}{\log \log n})$ time. ◀

▶ **Theorem 23.** *Algorithm 2 performs broadcasting in radio networks in $O(n \log L \log \log n)$ time, without knowledge of n or L.*

Proof. By Theorem 20, any layer of size ℓ remains leading for no more than $X(\ell)$ time-steps, where $X(\ell) = O(\ell \log L \log \log \ell)$. $Y(n, h)$ is then maximized by setting $\ell_1 = n$ and $\ell_i = 0$ for $i > 1$. So, by Lemma 21, broadcasting is performed for the entire radio network in $O(n \log L \log \log n)$ time. ◀

5 Conclusions

We have shown that *deterministic algorithms* for communications primitives in multiple access channels and multi-hop radio networks *need not assume parameter knowledge, or small IDs*, to be efficient. One possible next step would be to show a means by which nodes could generate efficient transmission schedules based on some finite (i.e., with size bounded by some function of ID) advice string; this would go some way towards making the algorithm practical.

References

1 N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43(2):290–298, 1991.

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(1):104–126, 1992.

3 B. Chlebus, L. Gasieniec, D. R. Kowalski, and T. Radzik. On the wake-up problem in radio networks. In *Proceedings of the 32nd Annual International Colloquium on Automata, Languages and Programming (ICALP)*, pages 347–359, 2005.

4 B. Chlebus and D. R. Kowalski. A better wake-up in radio networks. In *Proceedings of the 23rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 266–274, 2004.

5 B. Chlebus and D. R. Kowalski. Almost optimal explicit selectors. In *Proceedings of the 15th International Symposium on Fundamentals of Computation Theory (FCT)*, pages 270–280, 2005.

6 B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in unknown radio networks. *Distributed Computing*, 15(1):27–38, 2002.

7 B. S. Chlebus, L. Gasieniec, A. Östlin, and J. M. Robson. Deterministic radio broadcasting. In *Proceedings of the 27th Annual International Colloquium on Automata, Languages and Programming (ICALP)*, pages 717–728, 2000.

8 M. Chrobak, L. Gasieniec, and D. R. Kowalski. The wake-up problem in multihop radio networks. *SIAM Journal on Computing*, 36(5):1453–1471, 2007.

9 M. Chrobak, L. Gasieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. *Journal of Algorithms*, 43(2):177–189, 2002.

10 A. E. F. Clementi, A. Monti, and R. Silvestri. Distributed broadcasting in radio networks of unknown topology. *Theoretical Computer Science*, 302(1–3):337–364, 2003.

11 A. Czumaj and P. Davies. Exploiting spontaneous transmissions for broadcasting and leader election in radio networks. In *Proceedings of the 36th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 3–12, 2017.

12 A. Czumaj and P. Davies. Brief announcement: Randomized blind radio networks. In *Proceedings of the 32nd International Symposium on Distributed Computing (DISC)*, pages 44:1–44:3, 2018.

13 A. Czumaj and P. Davies. Deterministic communication in radio networks. *SIAM Journal on Computing*, 47(1):218–240, 2018.

14 A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 492–501, 2003.

15 L. Gasieniec, A. Pelc, and D. Peleg. The wakeup problem in synchronous broadcast systems. *SIAM Journal on Discrete Mathematics*, 14(2):207–222, 2001.

16 M. Ghaffari, B. Haeupler, and M. Khabbazian. Randomized broadcast in radio networks with collision detection. In *Proceedings of the 32nd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 325–334, 2013.

17 B. Haeupler and D. Wajc. A faster distributed radio broadcast primitive. In *Proceedings of the 35th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 361–370, 2016.

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

19 T. Jurdziński and G. Stachowiak. Probabilistic algorithms for the wakeup problem in single-hop radio networks. *Theory of Computing Systems*, 38(3):347–367, 2005.

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

21 D. Kowalski and A. Pelc. Faster deterministic broadcasting in ad hoc radio networks. *SIAM Journal on Discrete Mathematics*, 18(2):332–346, 2004.

22 D. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. *Distributed Computing*, 18(1):43–57, 2005.

23 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.

24 G. De Marco. Distributed broadcast in unknown radio networks. *SIAM Journal on Computing*, 39(6):2162–2175, 2010.

25 G. De Marco and A. Pelc. Faster broadcasting in unknown radio networks. *Information Processing Letters*, 79(2):53–56, 2001.

26 G. De Marco, M. Pelegrini, and G. Sburlati. Faster deterministic wakeup in multiple access channels. *Discrete Apllied Mathematics*, 155(8):898–903, 2007.

27 D. Peleg. Time-efficient broadcasting in radio networks: A review. In *Proceedings of the 4th International Conference on Distributed Computing and Internet Technology (ICDCIT)*, pages 1–18, 2007.

28 D. E. Willard. Log-logarithmic selection resolution protocols in a multiple access channel. *SIAM Journal on Computing*, 15(2):468–477, 1986.