DETERMINISTIC COMMUNICATION IN RADIO NETWORKS*

ARTUR CZUMAJ† AND PETER DAVIES‡

1

Abstract. In this paper we improve the deterministic complexity of two fundamental communication primitives in the classical model of ad-hoc radio networks with unknown topology: broadcasting and wake-up. We consider an unknown radio network, in which all nodes have no prior knowledge about network topology, and know only the size of the network n, the maximum in-degree of any node Δ, and the eccentricity of the network D.

For such networks, we first give an algorithm for wake-up, based on the existence of small universal synchronizers. This algorithm runs in $O(\frac{\min\{n, D\Delta\}\log n \log \Delta}{\log\log \Delta})$ time, the fastest known in both directed and undirected networks, improving over the previous best $O(n \log^2 n)$-time result across all ranges of parameters, but particularly when maximum in-degree is small.

Next, we introduce a new combinatorial framework of block synchronizers and prove the existence of such objects of low size. Using this framework, we design a new deterministic algorithm for the fundamental problem of broadcasting, running in $O(n \log D \log\log \frac{D\Delta}{n})$ time. This is the fastest known algorithm for the problem in directed networks, improving upon the $O(n \log n \log\log n)$-time algorithm of De Marco (2010) and the $O(n \log^2 D)$-time algorithm due to Czumaj and Rytter (2003). It is also the first to come within a log-logarithmic factor of the $\Omega(n \log D)$ lower bound due to Clementi et al. (2003).

Our results also have direct implications on the fastest *deterministic leader election* and *clock synchronization* algorithms in both directed and undirected radio networks, tasks which are commonly used as building blocks for more complex procedures.

Key words. Radio Networks, Broadcasting, Wake-up, Deterministic

AMS subject classifications. 68M10, 68W15, 05C85

1. Introduction.

1.1. Model of communication networks.
We consider the classical model of *ad-hoc radio networks* with *unknown structure*. A *radio network* is modeled by a (*directed* or *undirected*) network $\mathfrak{N} = (V, E)$, where the set of nodes corresponds to the set of transmitter-receiver stations. The nodes of the network are assigned different identifiers (IDs), and throughout this paper we assume that all IDs are distinct numbers in $\{1, \ldots, |V|\}$. 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.

In accordance with the standard model of unknown (ad-hoc) radio networks (for more elaborate discussion about the model, see, e.g., [1, 2, 6, 10, 11, 14, 20, 22, 25]), we 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. We assume that the only knowledge of each node is its own ID, the *size* of the network n, the *maximum in-degree* of any node Δ, and the *eccentricity* of the network D, which is the maximum distance from the source node to any node in $\mathfrak{N}$.

*Submitted to the editors DATE.

Funding: Research partially supported by the Centre for Discrete Mathematics and its Applications (DIMAP)

†University of Warwick (A.Czumaj.warwick.ac.uk, http://www.dcs.warwick.ac.uk/~czumaj/).

‡University of Warwick (P.W.Davies@warwick.ac.uk, http://www2.warwick.ac.uk/fac/sci/dcs/people/research/csukah/).

1 A preliminary version of the paper appeared in *Proceedings of the 43rd Annual International Colloquium on Automata, Languages and Programming (ICALP)*, Rome, Italy, July 12–15, 2016.

Nodes operate in discrete, synchronous time steps, but we do not need to assume knowledge of a global clock. When we refer to the "running time" of an algorithm, we mean the number of time steps which elapse before completion (i.e., we are not concerned with the number of calculations nodes perform within time steps). In each time step a node can either *transmit* a message to all of its out-neighbors at once or can remain silent and *listen* to the messages from its in-neighbors. Some variants of the model make restrictions upon message size (e.g. that they should be $O(\log n)$ bits in length); our algorithms only forward the source message so comply with any such restriction.

The distinguishing feature of radio networks is the interfering behavior of transmissions. In the most standard radio networks model, the *model without collision detection* (see, e.g., [1, 2, 11, 25]), which is studied in this paper, if a node v listens in a given round and precisely one of its in-neighbors transmits, then v receives the message. In all other cases v receives nothing; in particular, the lack of collision detection means that v is unable to distinguish between zero of its in-neighbors transmitting and more than one.

The model without collision detection describes the most restrictive interfering behavior of transmissions; also considered in the literature is a less restrictive variant, the model with collision detection, where a node listening in a given round can distinguish between zero of its in-neighbors transmitting and more than one (see, e.g., [14, 25]).

1.2. Discussion of assumptions of node knowledge. We consider the model that assumes that all nodes have knowledge of the parameters n, D, and Δ. While these assumption may seem strong, they are standard in previous works when running time dependencies upon the parameters appear. For example, the $O(n \log^2 D)$-time algorithm of [12] requires knowledge of n and D, and the $O(D\Delta \log \frac{n}{\Delta})$-time algorithm of [11] requires knowledge of n and Δ (though they provide methods of removing these knowledge assumptions at the expense of extra running time factors). Similar assumptions also appear in previous related work.

Furthermore, we note that nodes need only know common upper bounds for the parameters, rather than the exact values (these upper bounds will replace the true values in the running time expression). Therefore, even if only some polynomial upper bound for D is known, and no knowledge about Δ is assumed at all, our broadcasting algorithm still runs within $O(n \log D \log\log D)$ time, and remains the fastest known algorithm. Similarly, with only a polynomial upper bound on Δ and no bound on D, our wake-up algorithm still runs in $O(\frac{n \log n \log \Delta}{\log\log \Delta})$-time. In this latter case, the algorithm is also faster than previous algorithms when only n is known.

For both algorithms (as with all broadcasting and wake-up algorithms with at least linear dependency on n) this assumption too can be removed by standard double-and-test techniques, at the cost of never having acknowledgment of completion. The task of achieving acknowledgment in such circumstances is addressed in [26].

Note that to avoid non-well-defined expressions, we will use $\log(x)$ to mean $\min\{1, \log_2(x)\}$ wherever logarithms appear.

1.3. Communications primitives: broadcasting and wake-up. In this paper we consider two fundamental communications primitives, namely *broadcasting* and *wake-up*, and consider *deterministic protocols* for each of these tasks.

1.3.1. Broadcasting. *Broadcasting* is one of the most fundamental problems in communication networks and has been extensively studied for many decades (see, e.g.,

[25] and the references therein).

The premise of the broadcasting task is that one particular node, called the *source*, has a message which must become known to all other nodes. We assume that all other nodes start in a dormant state and do not participate until they are "woken up" by receiving the source message (this is referred to in some works as the "no spontaneous transmissions" rule). As a result, while the model does not assume knowledge of a global clock, we can make this assumption in practice, since the current time can be appended to the source message as it propagates, and therefore will be known be all active nodes. This is important since it allows us to synchronize node behavior into fixed-length *blocks*.

1.3.2. Wake-up. The *wake-up problem* (see, e.g., [17]) is a related fundamental communication problem that arises in networks where there is no designated "source" node, and no synchronized time-step at which all nodes begin communicating. The goal is for all nodes to become "active" by receiving some transmission. Rather than a single source node which begins active, we instead assume that some subset of nodes spontaneously become active at arbitrary time-steps. The task can be seen as broadcast from multiple sources, without the ability to assume a global clock. This last point is important, and results in wake-up protocols being slower than those for broadcast, since nodes cannot co-ordinate their behavior.

1.4. Related work. As a fundamental communications primitive, the task of *broadcasting* has been extensively studied for various network models for many decades.

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 $\mathcal{O}(n^{11/6})$-time broadcasting algorithm. After several small improvements (cf. [7, 24]), Chrobak et al. [10] designed an almost optimal algorithm that completes the task in $\mathcal{O}(n \log^2 n)$ time, the first to be only a poly-logarithmic factor away from linear dependency. Kowalski and Pelc [20] improved this bound to obtain an algorithm of complexity $\mathcal{O}(n \log n \log D)$ and Czumaj and Rytter [12] gave a broadcasting algorithm running in time $O(n \log^2 D)$. Finally, De Marco [23] designed an algorithm that completes broadcasting in $O(n \log n \log \log n)$ time steps. Thus, in summary, the state of the art result for deterministic broadcasting in directed radio networks with unknown structure (without collision detection) is the complexity of $O(n \min\{\log n \log \log n, \log^2 D\})$ [12, 23]. The best known lower bound is $\Omega(n \log D)$ due to Clementi et al. [11].

Broadcasting 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 [19] exists. If spontaneous transmissions are allowed and a global clock available, then deterministic broadcast can be performed in $O(n)$ 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 $\mathcal{O}((D + \log n) \cdot \log n)$. This bound has been later improved by Czumaj and Rytter [12], and independently Kowalski and Pelc [21], 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 [22].

Haeupler and Wajc [15] improved this bound for undirected networks in the model that allows spontaneous transmissions and designed an algorithm that completes broadcasting in $O(D \log n \log \log n / \log D + \log^{O(1)} n)$ time with high probability. In the model with collision detection for undirected networks, an $O(D + \log^6 n)$-time randomized algorithm due to Ghaffari et al. [14] 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 model, see e.g., [25] and the references therein.

The *wake-up problem* (see, e.g., [17]) is a related communication problem that arises in networks where there is no designated "source" node, and no synchronized time-step at which all nodes begin communicating. Before any more complex communication can take place, we must first require all nodes to be "active," i.e., aware that they should be communicating. This is the goal of wake-up, and it is a fundamental starting point for most other tasks in this setting, for example leader election and clock synchronization [9].

The first sub-quadratic deterministic wake-up protocol was given in by Chrobak et al. [9], 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 two improvements in running time, both 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]. Unlike for the problem of broadcast, the fastest known protocol for directed networks is also the fastest for undirected networks. Randomized wake-up has also been studied (see, e.g., [9, 18]). A recent survey of the current state of research on the wake-up problem is given in [17].

1.5. New results. In this paper we present a *new construction of universal radio synchronizers* and introduce and analyze a *new concept of block synchronizers* to improve the deterministic complexity of two fundamental communication primitives in the model of ad-hoc radio networks with unknown topology: broadcasting and wake-up.

By applying the analysis of block synchronizers, we present a new deterministic broadcasting algorithm (**Algorithm 1**) in directed ad-hoc radio networks with unknown structure, without collision detection, that for any directed network $\mathfrak{N}$ with n nodes, with eccentricity D, and maximum in-degree Δ, completes broadcasting in $O(n \log D \log\log \frac{D\Delta}{n})$ time-steps. This result almost matches a lower bound of $\Omega(n \log D)$ due to Clementi et al. [11], and improves upon the previous fastest algorithms due to De Marco [23] and due to Czumaj and Rytter [12], which require $O(n \log n \log\log n)$ and $O(n \log^2 D)$ time-steps, respectively.

Our result reveals that a non-trivial speed-up can be achieved for a broad spectrum of network parameters. Since $\Delta \le n$, our algorithm has the complexity at most $O(n \log D \log\log D)$. Therefore, in particular, it significantly improves the complexity of broadcasting for shallow networks, where $D \ll n^{O(1)}$. Furthermore, the dependency on Δ reduces the complexity even further for networks where the product $D\Delta$ is near linear in n, including sparse networks which can appear in many natural scenarios.

Our broadcasting result has also direct implications on the fastest *deterministic leader election algorithm* in directed and undirected radio networks. It is known that leader election can be completed in $O(\log n)$ times broadcasting time (see, e.g., [10, 13]) (assuming the broadcast algorithm extends to multiple sources, which is the case here as long as we have a global clock), and so our result improves the bound to achieve a deterministic leader election algorithm running in $O(n \log n \log D \log\log \frac{D\Delta}{n})$ time. For undirected networks the best result is $O(n \log^{3/2} n \sqrt{\log\log n})$ time [8] (we note that the $O(n \log D)$ broadcast protocol of [19] cannot be used at a $\log n$ slowdown for leader election, since it relies on token traversal and does not extend to multiple sources). Our result therefore favorably compares for shallow networks (for small D) even in undirected networks.

We also present a deterministic algorithm (**Algorithm 2**) for the related task of wake-up. We show the existence of universal radio synchronizers of delay $g(k) = O(\frac{n \log n \log k}{\log \log k})$, and demonstrate that this yields a wake-up protocol taking time $O(\frac{\min\{n, D\Delta\} \log n \log \Delta}{\log \log \Delta})$. This improves over the previous best result for both directed and undirected networks, the $O(n \log^2 n)$-time protocol of [3]; the improvement is largest when Δ is small, but even when it is polynomial in n, our algorithm is a $\log \log n$-factor faster.

Our improved result for wake-up has direct applications to communication algorithms in networks that do not have access to a global clock, where wake-up is an essential starting point for most more complex communication tasks. For example, wake-up is used as a subroutine in the fastest known protocols for fundamental tasks of *leader election* and *clock synchronization* (cf. [9]). These are two fundamental tasks in networks without global clocks, since they allow initially unsynchronized networks to be brought to a state in which synchronization can be assumed, and results from the better-understood setting with a global clock can then be applied. Our wake-up protocol yields $O(\frac{\min\{n, D\Delta\} \log^2 n \log \Delta}{\log \log \Delta})$-time leader election and clock synchronization algorithms, which are the fastest known in both directed and undirected networks.

1.6. Previous approaches. Almost all deterministic broadcasting protocols with sub-quadratic complexity (that is, since [6]) have made use of 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 $[n] := \{1, 2, \ldots, n\}$ below a certain size has an intersection of size exactly 1 with some member of the family. 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 an active 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).

One of the drawbacks of selective-family based algorithms is that applying them requires coordination between nodes. For the problem of broadcast, this means that some time may be wasted waiting for the current selective family to finish, and also that nodes cannot alter their behavior based on the time since they were informed, which might be desirable. For the problem of wake-up, this is even more of a difficulty; since we cannot assume a global clock, we cannot synchronize node behavior and hence cannot use selective families at all.

To tackle this issue, Chrobak et al. [9] 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. However, the adaptability to different node start times comes at a cost of increased size, meaning that synchronizer-based wake-up algorithms were slightly slower than selective family-based broadcasting algorithms.

The proofs of existence for selective families and synchronizers 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. The proofs are all non-constructive (and therefore all resulting algorithms non-explicit; cf. [16, 5] for explicits construction of selective families).

Returning to the problem of broadcasting, a breakthrough came in 2010 with a paper by De Marco [23] which took a new approach. Rather than having all nodes synchronize their behavior, it instead had them begin their own unique pattern, starting immediately upon being informed. These behavior patterns were collated into a transmission matrix. The existence of a transition matrix with appropriate selective properties was then proven probabilistically. The ability for a node to transmit with a frequency which decayed over time allowed De Marco's method to inform nodes with a very large in-neighborhood faster, and this in turn reduced total broadcasting time from $O(n \log^2 D)$ [12] to $O(n \log n \log \log n)$.

A downside of this new approach is that having nodes begin immediately, rather than wait until the beginning of the next selector, gives rise to a far greater number of possible starting-time scenarios that have be accounted for during the probabilistic proof. This caused the logarithmic factor in running time to be $\log n$ rather than $\log D$. Furthermore, the method was comparatively slow to inform nodes of low in-degree, compared to a selective family of appropriate size. These are the difficulties that our approach overcomes.

1.7. Overview of our approach. Our wake-up result follows a similar line to the previous works; we prove the existence of smaller universal synchronizers than previously known, using the probabilistic method. Our improvement stems from new techniques in analysis rather than method, which allow us to gain a log-logarithmic factor by choosing what we believe are the optimal probabilities by which to construct a randomized candidate.

Our broadcasting result takes a new direction, some elements of which are new and some of which can be seen as a compromise between selective family-type objects and the transmission schedules of De Marco [23]. We first note that nodes of small in-degree can be quickly dealt with by repeatedly applying $(n, \frac{n}{D})$-selective families "in the background" of the algorithm. This allows us to tailor the more novel part of the approach to nodes of large in-degree. We have nodes performing their own behavior patterns with decaying transmission frequency over time, but they are semi-synchronized to "blocks" of length roughly $\frac{n}{D}$, in order to cut down the number of circumstances we must consider. This idea is formalized by the concept of *block synchronizers,* combinatorial objects which can be seen as an extension of the radio synchronizers used for wake-up.

An important new concept used in our analysis of block synchronizers (and also in our proof of small universal synchronizers) is that of *cores*. Cores reduce a set of nodes and starting times to a (usually smaller) set of nodes which are active during a critical period. In this way we can combine many different circumstances into a single case, and demonstrate that for our purposes they all behave in the same way.

The most technically involved part of both of the proofs is the selection of the probabilities with which we generate a randomized candidate object (universal synchronizer or block synchronizer). Intuitively, when thinking about radio networks, a node in our network is aiming to inform its out-neighbors, and it should assume that as time goes on, only those with large in-neighborhoods will remain uninformed (because these nodes are harder to inform quickly). Therefore a node should transmit with ever-decreasing frequency, roughly inversely proportional to how large it estimates remaining uninformed neighbors' in-neighborhoods must be. However, the size of these in-neighborhoods cannot be estimated precisely, and so we must tweak the probabilities slightly to cover the possible range. In block synchronizers we do this using phases of length $O(\log\log\frac{D\Delta}{n})$ during which nodes halve their transmission probability every step, but since behavior must be synchronized to achieve this we cannot do the same for radio synchronizers. Instead, we allow our estimate to be further from the true value, and require more time-steps around the same value to compensate.

As with previous results based on selective families, synchronizers, or similar combinatorial structures, the proofs of the structures we give are non-constructive, and therefore the algorithms are non-explicit.

2. Combinatorial tools.

Our communications protocols rely upon the existence of objects with certain combinatorial properties, and we will separate these more abstract results from their applications to radio networks. In this section, we will define the combinatorial objects we will need. Next, in Sections 3–4, we will demonstrate in detail how these combinatorial objects can be used to obtain fast algorithms for broadcasting and wake-up.

2.1. Selective families.

We begin with a brief discussion about *selective families*, whose importance in the context of broadcasting was first observed by Chlebus et al. [6]. A selective family is a family of subsets of $[n] := \{1, \ldots, n\}$ such that every subset of $[n]$ below a certain size has intersection of size exactly 1 with a member of the family. For the sake of consistency with successive definitions, rather than defining the family of subsets S_i, we will instead use the equivalent definition of a set of binary sequences S^v (that is, $S_i^v = 1$ if and only if $v \in S_i$).

For some $m \in \mathbb{N}$, let each $v \in [n]$ have its own length-m binary sequence $S^v = S_0^v S_1^v S_2^v \ldots S_{m-1}^v$.

DEFINITION 2.1. *$S = \{S^v\}_{v\in[n]}$ is an (n, k)-**selective family** if for any $X \subseteq [n]$ with $1 \le |X| \le k$, there exists j, $0 \le j < m$, such that $\sum_{v\in X} S_j^v = 1$. (We say that such j* hits *X.)*

2.1.1. Existence of small selective families.

The following standard lemma (see, e.g., [11]) posits the existence of (n, k)-selective families of size $O(k \log \frac{n}{k})$. This has been shown to be asymptotically optimal [11].

LEMMA 2.2 (**Small selective families**). *For some constant c and for any $1 \le k \le n$, there exists an (n, k)-selective family of size at most $m = ck \log \frac{n}{k}$.*

2.1.2. Application to radio networks.

During the course of radio network protocols we can "apply" a selective family S on an n-node network by having each node v transmit in time-step j if and only if v has a message it wishes to transmit and $S_j^v = 1$ (see, e.g., [6, 11]). Some previous protocols involved nodes starting to transmit immediately if they were informed of a message during the application of a selective family (or a variant called a selector designed for such a purpose), but here we will require nodes to wait until the current selective family is completed before they start participating. That is, nodes only attempt to transmit their message if they knew it at the beginning of the current application.

The result of applying an (n, k)-selective family is that any node u which has between 1 and k active neighbors before the application will be informed of a message upon its conclusion. This is because there must be some time-step j which hits the set of u's active neighbors, and therefore exactly one transmits in that time-step, so u receives a message. This method of selective family application in radio networks was first used in [6].

2.2. Radio synchronizers. Radio synchronizers are an extension of selective families designed to account for nodes in a radio network starting their behavior patterns at different times, and without access to a global clock. They were first introduced in [9] and used in an algorithm for performing wake-up, and this is also the purpose for which we will apply them.

To define radio synchronizers, we first define the concept of *activation schedule*.

DEFINITION 2.3. *An n-**activation schedule** is a function $\omega : [n] \to \mathbb{N}$.*

We will extend the definition to subsets $X \subseteq [n]$ by setting $\omega(X) = \min_{v \in X} \omega(v)$.

As for selective families, let each $v \in [n]$ have its own length-m binary sequence $S^v = S^v_0 S^v_1 S^v_2 \dots S^v_{m-1}$. We then define radio synchronizers as follows:

DEFINITION 2.4. *$S = \{S^v\}_{v \in [n]}$ is called an (n, k, m)-**radio synchronizer** if for any activation schedule ω and for any $X \subseteq [n]$ with $1 \le |X| \le k$, there exists j, $\omega(X) \le j < \omega(X) + m$, such that $\sum_{v \in X} S^v_{j-\omega(v)} = 1$.*

One can see that the definition is very similar to that of selective families (Definition 2.1), except that now each v's sequence is offset by the value $\omega(v)$. To keep track of this shift in expressions such as the sum in the definition, we will call such values j *columns*. As with selective families, we say that any column j satisfying the condition in Definition 2.4 *hits* X.

In [4], the concept of radio synchronizers was extended to universal radio synchronizers which cover the whole range of k from 1 to n. Let $g : [n] \to \mathbb{N}$ be a non-decreasing function, which we will call the *delay* function.

DEFINITION 2.5. *$S = \{S^v\}_{v \in [n]}$ is called an (n, g)-**universal radio synchronizer** if for any activation schedule ω, and for any $X \subseteq [n]$, there exists column j, $\omega(X) \le j < \omega(X) + g(|X|)$, such that $\sum_{v \in X} S^v_{j-\omega(v)} = 1$.*

2.2.1. New result: Existence of small universal radio synchronizers. We obtain a new, improved construction of universal radio synchronizers, which improves over the previous best result of Chlebus et al. [3] of universal synchronizers with $g(q) = O(q \log q \log n)$.

THEOREM 2.6. *For any $n \in \mathbb{N}$, there exists an (n, g)-**universal radio synchronizer** with $g(q) = O(\frac{q \log q \log n}{\log \log q})$.*

Our approach 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. We will prove Theorem 2.6 in Section 5.

2.2.2. Application of universal radio synchronizers to radio networks. One can apply universal radio synchronizers to the problem of wake-up in radio networks by having $\omega(v)$ represent the time-step in which node v becomes active during the course of a protocol (either spontaneously or by receiving a transmission). Subsequently, v interprets S^v as the pattern in which it should transmit, starting immediately from time-step $\omega(v)$. That is, in each time-step j after activation, v checks the next value in S^v (i.e., $S^v_{j-\omega(v)}$), transmits if it is **1** and stays silent otherwise. Then, the selective property specified by the definition guarantees that any node u with an in-neighborhood of size q hears a transmission within at most $g(q)$ steps of its first in-neighbor becoming active.

We will present this approach in details in Section 3.2, where we will obtain a new, improved algorithm for the wake-up problem.

2.3. Block synchronizers. Next, we introduce *block synchronizers*, which are a new type of combinatorial object designed for use in a fast broadcasting algorithm. They can be seen as an extension of both radio synchronizers and the transmission matrix formulation of De Marco [23].

Let ω be an n-activation schedule (cf. Definition 2.3). Let each $v \in [n]$ have its own length-m binary sequence $S^v = S_0^v S_1^v S_2^v \dots S_{m-1}^v$. For any fixed B, define a function $\mu_B : \mathbb{N} \to \mathbb{N}$ which rounds its input up to the next multiple of B, that is, $\mu_B(x) = \min\{pB : p \geq \frac{x}{B}, p \in \mathbb{N}\}$; we will call $s(v) := \mu_B(\omega(v))$ the *start column* of v. We extend s to subsets of $[n]$ in the obvious way, $s(X) = \mu_B(\omega(X))$.

DEFINITION 2.7. *$S = \{S^v\}_{v\in[n]}$ is an (n, Δ, r, B)-**block synchronizer** if for any activation schedule ω and any set $X \subseteq [n]$ with $|X| \leq \Delta$, there exists a column j, $s(X) \leq j < s(X) + B \cdot \lceil \frac{|X|}{r} \rceil$, such that $\sum_{v\in X} S^v_{j-s(v)} = 1$.*

Block synchronizers differ from radio synchronizers in two ways: Firstly, on top of the offsetting effect of the activation schedule, there is also the function μ_B that effectively "snaps" behavior patterns to blocks of size B, hence the name block synchronizer. Secondly, the size of the range in which we must hit X is linearly dependent on $|X|$. This could be generalized to a generic non-decreasing function $g(|X|)$ as with universal radio synchronizers, but here for simplicity we choose to use the specific function which works best for our broadcasting application. The parameter r is the increment by which each block increases the size of sets we can hit.

2.3.1. New result: Existence of small block synchronizers. We will show the existence of small block synchronizers in the following theorem.

THEOREM 2.8. *For any $n, D, \Delta \in \mathbb{N}$ with $D,\ \Delta \leq n < D\Delta$, there exists an $(n, \Delta, \frac{n}{D}, O(\frac{n}{D} \log D \log\log \frac{D\Delta}{n}))$-**block synchronizer**.*

We will prove the existence of a small block synchronizer by randomly generating a candidate S, and proving that it indeed has the required properties with positive probability, in a similar fashion to the proof of small radio synchronizers. We will prove Theorem 2.8 in Section 6.

2.3.2. Application of block synchronizers to radio networks. The idea of our broadcasting algorithm will be that any node v waits until the start of the first block after its activation time $\omega(v)$, and then begins its transmission pattern S^v. The definition of block synchronizer aims to model this scenario. The hitting condition ensures that any node with an in-neighborhood of size $q \leq \Delta$ will be informed within $B\lceil \frac{q}{r} \rceil$ time-steps of the start of the block in which its first in-neighbor begins transmitting.

We will present this approach in details in Section 3.1, where we will obtain a new, improved algorithm for the broadcasting problem.

3. Algorithms for broadcasting and wake-up.

In this section we use the machinery developed in the previous section to design our algorithms for broadcasting and wake-up in radio networks.

3.1. Broadcasting. We will assume that $D\Delta > n$, otherwise an earlier $O(D\Delta \log \frac{n}{\Delta})$-time protocol from [11] can be used to achieve $O(D\Delta \log \frac{n}{\Delta}) = O(n \log D)$ time.

Let $\mathcal{S}$ be an $(n, \Delta, \frac{n}{D}, \mathcal{B})$-block synchronizer, with $\mathcal{B} = c\frac{n}{D} \log D \log\log \frac{D\Delta}{n}$ (cf. Theorem 2.8), and recall that $\mu_{\mathcal{B}}(x) = \min\{p\mathcal{B} : p \geq \frac{x}{\mathcal{B}}, p \in \mathbb{N}\}$, i.e. the start of the first block after x. We will say that the source node becomes active at time-step 0, and any other node v becomes active in a time-step i if it received its first transmission at time-step $i-1$. Our broadcasting algorithm is the following (Algorithm 1):

Algorithm 1 Broadcast at a node v

```
Let i be the time-step in which v becomes active
for j from 0 to DB − 1, in time-step μ_B(i) + j do
    v transmits source message iff S_j^v = 1
end for
```

3.2. Wake-up. Let S be an (n,g)-universal radio synchronizer with $g(q) = \frac{cq \log q \log n}{\log\log q}$ (cf. Theorem 2.6). We will say that a node v becomes active in a time-step i if it either spontaneous wakes up at i, or received its first transmission at time-step $i-1$. Our wake-up algorithm is the following (Algorithm 2):

Algorithm 2 Wake-up at a node v

```
Let i be the time-step in which v becomes active
for j from 0 to g(n) − 1, in time-step i + j do
    v transmits source message iff S_j^v = 1
end for
```

---

4. Analysis of broadcasting and wake-up algorithms. In this section we show that our algorithms for broadcasting and wake-up have the claimed running times. Our analysis critically relies on the constructions of small block synchronizers and small universal radio synchronizers, as presented in Theorems 2.8 and 2.6.

We begin with the analysis of the broadcasting algorithm.

THEOREM 4.1. *Algorithm 1 performs broadcast in $O(n \log D \log\log \frac{D\Delta}{n})$ time-steps.*

To begin the analysis, fix some arbitrary node v and let P be a shortest path from the source (or first informed node) x to v. Number the nodes in this path consecutively, e.g., $P_0 = x$ and $P_{dist(x,v)} = v$. Classify all other nodes into *layers* dependent upon the furthest node along the path P to which they are an in-neighbor (some nodes may not be an in-neighbor to any node in P; these can be discounted from the analysis). That is, layer $L_\ell = \{u \in V : \max_{u \text{ in-neighbour to } P_i} i = \ell\}$ for $\ell \le dist(x,v)$. We separately define layer $L_{dist(x,v)+1}$ to be $\{v\}$.

(For a depiction of layer numbering, see Figure 1.)

At any time step, we call a layer *leading* if it is the foremost layer containing an active node, and our goal is to progress through the network until the final layer is leading, i.e., v is active. The use of layers allows us to restrict to the set of nodes of our main interest: if we focus on the path node whose in-neighborhood contains the leading layer, we cannot have interference from earlier layers since they contain no in-neighbors of this path node, and we cannot have interference from later layers since they are not yet active.

LEMMA 4.2. *Let $h : [\Delta] \to \mathbb{N}$ be a non-decreasing function, and define $T(n,D,\Delta,h)$ to be the supremum of the function $\sum_{i=1}^{D} h(q_i)$, where integers $1 \le q_i \le \Delta$ satisfy the additional constraint $\sum_{i=1}^{D} q_i \le n$. If a broadcast or wake-up protocol ensures that any layer (under any choice of v) of size q remains leading for no more than $h(q)$ time-steps, then all nodes become active within $T(n,D,\Delta,h)$ time-steps.*

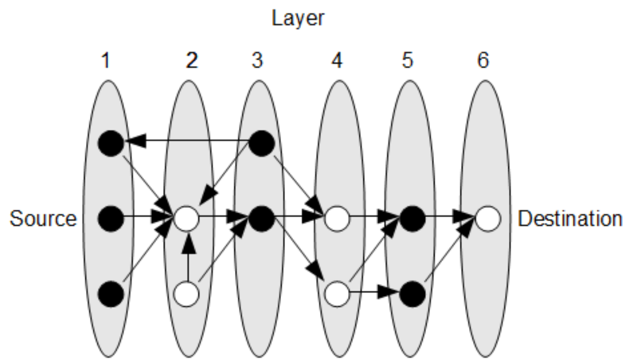

FIG. 1. *An example of layer numbering.*

Proof. Let $q_i = |L_i|$. Layer $L_{dist(x,v)+1}$ must be leading (and thus node v active) once no other layers are leading, and so this occurs within $\sum_{i=1}^{dist(x,v)} h(q_i)$ time-steps after layer L_1 becomes leading. Since $\sum_{i=1}^{dist(x,v)} h(q_i) \leq \sum_{i=1}^{D} h(q_i)$ and $\sum_{i=1}^{D} q_i \leq n$, this is no more than $T(n, D, \Delta, h)$ time-steps.

Since v was chosen arbitrarily, all nodes must be active within $T(n, D, \Delta, h)$ time-steps of x becoming active. ◻

We make use of Lemma 4.2 to give bounds on the running times of our algorithms:

LEMMA 4.3. *Algorithm 1 ensures that any layer of size q remains leading for fewer than $\mathcal{B}\lceil\frac{q+r}{r}\rceil$ time-steps.*

Proof. For all nodes w, let $\omega(w)$ be the time-step that w becomes active during the course of the algorithm. By definition of a block selector, for any layer L_i of size q_i there is a time-step $j < s(L_i) + B\lceil\frac{q_i}{r}\rceil$ in which exactly one element of L_i transmits. Then, either path node P_i hears the transmission (and so layer L_i is no longer leading in time-step $j+1$), or P_i has active in-neighbors not in L_i, in which case these must be in a later layer so L_i is not leading. Thus, L_i can remain leading for no more than $s(L_i) + \mathcal{B}\lceil\frac{q_i}{r}\rceil - \omega(L_i) < \mathcal{B}\lceil\frac{q_i+r}{r}\rceil$ time-steps. ◻

With these tools, we are now ready to complete the proof of Theorem 4.1.

Proof of Theorem 4.1. By Lemma 4.2, Algorithm 1 ensures that all nodes are active (and have therefore heard the source message) within $T(n, D, \Delta, h)$ time-steps, where $h(q) = \mathcal{B}\lceil\frac{q+r}{r}\rceil$. We will use an upper bound $T(n, D, \Delta, h')$, where $h'(q) = \mathcal{B}\frac{q+2r}{r}$. Since h' is linear and increasing, $\sum_{i=1}^{D} h'(q_i)$ subject to $\sum_{i=1}^{D} q_i \leq n$ is maximized whenever $\sum_{i=1}^{D} q_i = n$, for example at $q_i = \frac{n}{D}$ for all $i \in [D]$. So, the algorithm completes broadcast within

$$\sum_{i=1}^{D} h'(\frac{n}{D}) = \sum_{i=1}^{D} \mathcal{B}\frac{\frac{n}{D} + 2r}{r} = 3\mathcal{B}D = 3c'n \log D \log\log \frac{D\Delta}{n}$$

time-steps. ◻

In a similar way, we can analyze Algorithm 2:

THEOREM 4.4. *Algorithm 2 performs wake-up in $O(\frac{\min(n, D\Delta)\log n \log\Delta}{\log\log\Delta})$ time-steps.*

Proof. By Lemma 4.2, and the selective property of the universal synchronizers proven in Theorem 2.6, Algorithm 2 ensures that all nodes are active within $T(n, D, \Delta, g)$ time-steps, where $g(q) = \frac{cq \log q \log n}{\log \log q}$. Since g is convex and increasing, $\sum_{i=1}^{D} g(q_i)$ subject to $\sum_{i=1}^{D} q_i \leq n$ and $q_i \leq \Delta$ is maximized at $q_i = \Delta$ if $i \leq \frac{n}{\Delta}$, and $q_i = 0$ otherwise. Hence, the algorithm completes wake-up within

$$\sum_{i=1}^{\min(D, \frac{n}{\Delta})} g(\Delta) = \sum_{i=1}^{\min(D, \frac{n}{\Delta})} \frac{c\Delta \log \Delta \log n}{\log \log \Delta} = \frac{c \min(n, D\Delta) \log n \log \Delta}{\log \log \Delta}$$

time-steps. ∎

5. Small universal radio synchronizers: Proof of Theorem 2.6. In this section we will prove our main result about the existence of small universal radio synchronizers, Theorem 2.6. We first restate the theorem:

THEOREM 2.6. *For any $n \in \mathbb{N}$, there exists an (n, g)-**universal radio synchronizer** with $g(q) = O(\frac{q \log q \log n}{\log \log q})$.*

Our approach 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. We note that, since we are only concerned with asymptotic behavior, we can assume that n is at least a sufficiently large constant.

Let c be a constant to be chosen later. Our candidate $S = \{S^v\}_{v \in [n]}$ will be generated by independently choosing each S^v_j (for $j < g(n)$) to be **1** with probability $\frac{c \log n}{6(j + c \log n)}$ and **0** otherwise.

In analyzing whether S hits all sets $X \subseteq [n]$ under any activation schedule, we must first define the concept of a *core* to reduce the number of possibilities we must consider.

DEFINITION 5.1. *Fix any $X \subseteq [n]$ and any activation schedule ω. Let X_j be the elements of X which are active by column j, i.e., $X_j = \{v \in X : \omega(v) \leq j\}$. Let j' be the smallest j such that $j - \omega(X) \geq g(|X_j|)$. For every v, define $\psi(v) = \omega(v) - \omega(X)$, i.e., ψ is ω shifted so that $\psi(X) = 0$.*

*The **core** $C_{X,\omega}$ of a subset $X \subseteq [n]$ with respect to activation schedule ω is defined to be*

$$\{(v, \psi(v)) : \omega(v) < j'\}$$

This definition aims to narrow our focus to only the important elements in a particular subset X. Cores cut down the number of possibilities by removing redundant elements which only become active after the set must already have been hit, and by shifting activation times to begin at zero (which, as we show, can be done without loss of generality). We do not want cores to be subject to an overriding activation schedule, so we include the activation times of elements of a core within its definition. When we talk about "hitting" a core, we mean using these incorporated activation times rather than an activation schedule, and we assume that column numberings start at 0 at the beginning of the core.

We note that if S hits a core $C_{X,\omega}$ within $g(|C_{X,\omega}|)$ columns under ψ, then it hits the set X within $g(|X|)$ columns under ω.This result allows us to 'shift' the activation times, and analyze a core independently of the many activation schedules from which it could be derived. We now need only prove that our candidate synchronizer hits all possible cores, since this will imply that it hits all subsets of $[n]$ under all activation schedules.

We make one further definition which will simplify our analysis:

DEFINITION 5.2. *For a core C and column j, let $C(j)$ denote $\{(v,\psi(v)) \in C : \psi(v) \leq j\}$. The **load** of column j of core C, denoted $f_C(j)$, is defined to be $f_C(j) = \sum_{(v,\psi(v))\in C(j)} \frac{c\log n}{6(j-\psi(v)+c\log n)}$.*

Note that load of a column j of core C is the expected number of $\mathbf{1}$s in a column, under the probabilities used for our candidate S, that is, $f_C(j) = \sum_{(v,\psi(v))\in C(j)} \mathbf{Pr}[S^v_{j-B\phi(v)} = \mathbf{1}]$.

If $f_C(j)$ is close to constant, then the probability of S hitting C in column j will also be almost constant. We therefore wish to bound $f_C(j)$, both from above and below.

LEMMA 5.3. *For all $j < g(|C|)$, $f_C(j) > \frac{\log\log|C|}{12\log|C|}$.*

Proof. The minimum contribution each $v \in C(j)$ can add to $f_C(j)$ is $\frac{c\log n}{6(j+c\log n)}$. Hence, $f_C(j) \geq \frac{c\log n}{6(j+c\log n)} \cdot |C(j)|$. To bound this quantity, we separate into two cases:

Case 1: $j < c\log n$. In this case we can obtain an adequate bound simply using that $|C| \geq 1$:

$$\frac{c\log n}{6(j+c\log n)} \cdot |C(j)| \geq \frac{c\log n}{6(j+c\log n)} > \frac{1}{12} \geq \frac{\log\log|C|}{12\log|C|}$$

Case 2: $j \geq c\log n$. If $j < g(|C|)$, then we also have $j < g(|C(j)|)$. This can be seen by examining any set X and activation schedule ω from which C can be derived, and noting that

$$j + \omega(X) < g(|C|) + \omega(X) = g(|X_{j'}|) + \omega(X) \leq j'$$

by Definition 5.1, and so

$$j = (j+\omega(X)) - \omega(X) < g(|X_{j+\omega(X)}|) = g(|C(j)|)$$

also by Definition 5.1.

Recalling (cf. Theorem 2.6) that $g(q) = \frac{cq\log q\log n}{\log\log q}$, rearranging gives $|C(j)| > \frac{j\log\log|C(j)|}{c\log n\log|C(j)|}$. Therefore total load is bounded by

$$f_C(j) \geq \frac{c\log n}{6(j+c\log n)} \cdot |C(j)| > \frac{j\log\log|C(j)|}{6(j+c\log n)\log|C(j)|} \geq \frac{\log\log|C|}{12\log|C|} \qquad \square$$

This lemma provides a lower bound on $f_C(j)$. We also need an upper bound, but we cannot obtain a good one for all j, since transmission load in a particular column can be as large as $|C|$. We instead prove that the set of columns with load within our desired range is sufficiently large.

Let $\mathcal{F}_C = \{j < g(|C|) : \frac{\log\log|C|}{12\log|C|} < f_C(j) < \frac{1}{2}\log\log|C|\}$. We prove the following bound:

LEMMA 5.4. $|\mathcal{F}_C| \geq \frac{c|C|\log n\log|C|}{10\log\log|C|}$.

Proof. Let us first upper-bound the total load over all columns $j < g(|C|)$:

$$\sum_{j<g(|C|)} f_C(j) = \sum_{j<g(|C|)} \sum_{(v,\psi(v))\in C(j)} \frac{c\log n}{6(j-\psi(v)+c\log n)}$$

$$= \sum_{(v,\psi(v))\in C} \sum_{j<g(|C|)} \frac{c\log n}{6(j-\psi(v)+c\log n)}$$

(by standard integral bound)

$$\le \sum_{(v,\psi(v))\in C} \int_{\psi(v)-1}^{g(|C|)-1} \frac{c\log n}{6(j-\psi(v)+c\log n)}dj$$

(evaluating integral)

$$= \frac{c\log n}{6} \sum_{(v,\psi(v))\in C} \ln\left(\frac{g(|C|)-1-\psi(v)+c\log n}{c\log n-1}\right)$$

$$\le \frac{c\log n\cdot|C|}{6}\cdot\ln\left(\frac{g(|C|)+c\log n-1}{c\log n-1}\right)$$

(substituting g's definition)

$$= \frac{c|C|\log n}{6}\cdot\ln\left(\frac{\frac{c|C|\log n\log|C|}{\log\log|C|}+c\log n-1}{c\log n-1}\right)$$

$$\le \frac{c|C|\log n}{6}\cdot\ln\left(\frac{\frac{c|C|\log n\log|C|}{\log\log|C|}}{\frac{1}{2}c\log n}+1\right)$$

$$\le \frac{c|C|\log n}{6}\cdot\ln(4|C|^{1.1})$$

$$= \frac{1.1\ln 2\log|C|+\ln 4}{6}c|C|\log n$$

$$\le 0.45c|C|\log n\log|C|$$

In the penultimate inequality we use that $\frac{2|C|\log|C|}{\log\log|C|}+1\le 4|C|^{1.1}$, which is obvious for sufficiently large $|C|$ and can be checked manually for small $|C|$ (remembering that we consider $\log(x)$ to mean $\min\{\log_2(x),1\}$). The final inequality can be checked similarly.

Since $f_C(j)\ge 0$ for any $j<g(|C|)$, the inequality above implies that the number of columns $j<g(|C|)$ with $f_C(j)\ge\frac{1}{2}\log\log|C|$ must be fewer than $\frac{0.9c|C|\log n\log|C|}{\log\log|C|}$. Therefore, since by Lemma 5.3 all elements $j\notin\mathcal{F}_C$ must have $f_C(j)\ge\frac{1}{2}\log\log|C|$, and since $g(|C|)=\frac{c|C|\log n\log|C|}{\log\log|C|}$, we obtain:

$$|\mathcal{F}_C| \ge g(|C|)-\frac{0.9c|C|\log n\log|C|}{\log\log|C|} = \frac{c|C|\log n\log|C|}{10\log\log|C|}$$ ∎

Next, we will give a lower bound for the probability that j hits C, which will later be shown to imply that columns in the set $\mathcal{F}_C$ (and hence the candidate synchronizer as a whole) have a good probability of hitting C. The following lemma, or variants thereof, has been used in several previous works such as [23], but we prove it here for completeness.

Lemma 5.5. *Let x_i, $i\in[n]$ be independent $\{0,1\}$-valued random variables with $\mathbf{Pr}[x_i=1]\le\frac{1}{2}\forall i$, and let $f=\sum_{i\in[n]}\mathbf{Pr}[x_i=1]$. Then $\mathbf{Pr}[\sum_{i\in[n]}x_i=1]\ge f4^{-f}$.*

Proof.

$$
\begin{aligned}
\mathbf{Pr}[\sum_{i\in[n]} x_i = 1] &= \sum_{j\in[n]} \mathbf{Pr}[x_j = 1 \wedge x_i = 0 \forall i \neq j] \\
&\geq \sum_{j\in[n]} \mathbf{Pr}[x_j = 1] \cdot \mathbf{Pr}[x_i = 0 \forall i] \\
&\geq f \cdot \mathbf{Pr}[x_i = 0 \forall i] \\
&= f \cdot \prod_{i\in[n]} (1 - \mathbf{Pr}[x_i = 1]) \\
&\geq f \cdot \prod_{i\in[n]} 4^{-\mathbf{Pr}[x_i=1]} \\
&= f \cdot 4^{-\sum_{i\in[n]} \mathbf{Pr}[x_i=1]} \\
&= f 4^{-f}
\end{aligned}
$$ ∎

For any j, applying this lemma with $x_v = S^v_{j-\psi(v)}$, we get that the probability that j hits C is at least $f_C(j) \cdot 4^{-f_C(j)}$.

Lemma 5.6. *For any core C, the probability that there is no column $j < g(|C|)$ that hits C is at most $1 - n^{\frac{-c|C|}{140 \ln 2}}$.*

Proof. By Lemma 5.5, each column j independently hits C with probability at least $f_C(j) \cdot 4^{-f_C(j)}$. To proceed with the analysis we will focus on the columns in $\mathcal{F}_C$, that is, columns $j < g(|C|)$ with $\frac{\log\log|C|}{12\log|C|} < f_C(j) < \frac{1}{2}\log\log|C|$.

Let us consider the function $1 - x4^{-x}$ for $x > 0$, and notice that this function has a global minimum at $\mu = 1/\ln 4$, is decreasing for $x < \mu$, and is increasing for $x > \mu$. For simplicity of notation, let h denote the number of columns $j \in \mathcal{F}_C$ with $\mu < f_C(j) < \frac{1}{2}\log\log|C|$. Then, the probability that no columns hit is upper bounded as follows:

$$
\begin{aligned}
\mathbf{Pr}[\text{no column hits}] &\leq \prod_{j<g(|C|)} (1 - f_C(j) \cdot 4^{-f_C(j)}) \\
&\leq \prod_{j\in\mathcal{F}_C} (1 - f_C(j) \cdot 4^{-f_C(j)}) \\
&= \prod_{\substack{j\in\mathcal{F}_C,\\ \mu<f_C(j)\leq\frac{1}{2}\log\log|C|}} (1 - f_C(j)4^{-f_C(j)}) \prod_{\substack{j\in\mathcal{F}_C,\\ \frac{\log\log|C|}{12\log|C|}<f_C(j)\leq\mu}} (1 - f_C(j) \cdot 4^{-f_C(j)})
\end{aligned}
$$

(since products are maximised by setting $f_C(j) = \frac{1}{2}\log\log|C|$ and $f_C(j) = \frac{\log\log|C|}{12\log|C|}$, respectively)

$$
\begin{aligned}
&\leq \prod_{\substack{j\in\mathcal{F}_C,\\ \mu<f_C(j)\leq\frac{1}{2}\log\log|C|}} \left(1 - \frac{\log\log|C|}{2\log|C|}\right) \prod_{\substack{j\in\mathcal{F}_C,\\ \frac{\log\log|C|}{12\log|C|}<f_C(j)\leq\mu}} \left(1 - \frac{\log\log|C|}{14\log|C|}\right) \\
&\leq \left(1 - \frac{\log\log|C|}{2\log|C|}\right)^h \cdot \left(1 - \frac{\log\log|C|}{14\log|C|}\right)^{|\mathcal{F}_C|-h}
\end{aligned}
$$

$$\leq \left(1 - \frac{\log\log|C|}{14\log|C|}\right)^{|\mathcal{F}_C|}$$

(by Lemma 5.4)

$$\leq \left(1 - \frac{\log\log|C|}{14\log|C|}\right)^{\frac{c|C|\log n \log|C|}{10\log\log|C|}}$$

(using $1 - x \leq e^{-x}$ for $x \in (0,1)$)

$$\leq e^{\frac{-c|C|\log n}{140}}$$

$$= n^{\frac{-c|C|}{140\ln 2}} \qquad \square$$

We now have a lower bound on the probability that S hits a particular core, but it remains to bound the number of possible cores we must hit.

Let C_q be the set of possible cores of size q.

LEMMA 5.7. $|C_q| \leq n^{3q}$.

Proof. There are at most $n \cdot g(n)$ possible pairs of $(v, \psi(v))$, and thus at most $\binom{n\cdot g(n)}{q}$ ways of choosing a size-q subset. So, $|C_q|$ is at most $\binom{n\cdot g(n)}{q} \leq (n \cdot g(n))^q = (\frac{cn^2\log^2 n}{\log\log n})^q \leq n^{3q}$ (for sufficiently large n). $\square$

We are now ready to prove our existence result:

LEMMA 5.8. *With positive probability, $\mathcal{S}$ is an (n, g)-universal synchronizer.*

Proof. We will set c to be $700\ln 2$. By union bound, using Lemmas 5.6 and 5.7,

$$\mathbf{Pr}[\mathcal{S} \text{ is an } (n,g)\text{-universal synchronizer}] \leq \sum_{q=1}^{n}\sum_{C\in C_q}\mathbf{Pr}[\text{C is not hit}]$$

$$\leq \sum_{q=1}^{n}\sum_{C\in C_q} n^{\frac{-c|C|}{140\ln 2}} \leq \sum_{q=1}^{n} n^{3q}\cdot n^{\frac{-cq}{140\ln 2}} = \sum_{q=1}^{n} n^{(3-\frac{c}{140\ln 2})q}$$

$$\leq \sum_{q=1}^{n} n^{-2q} < 1$$

$\square$

We are now ready to prove Theorem 2.6:

Proof. Since our candidate S satisfies the properties of an (n, g)-universal radio synchronizer with positive probability, such an object must exist. This completes the proof of Theorem 2.6. $\square$

6. Small block synchronizers: Proof of Theorem 2.8. In this section we will prove our main result about the existence of small block synchronizers, Theorem 2.8. We first restate the theorem:

THEOREM 2.8. *For any $n, D, \Delta \in \mathbb{N}$ with D, $\Delta \leq n < D\Delta$, there exists an $(n, \Delta, \frac{n}{D}, O(\frac{n}{D}\log D \log\log\frac{D\Delta}{n}))$-**block synchronizer**.*

As in our proof of the existence of small radio synchronizers (see Section 5), we only consider the case where n is at least a sufficiently large constant, since we are only concerned with asymptotic behavior. We will again need to define the *core* of a subset of $[n]$ (with respect to an activation schedule ω) in order to reduce the amount of possible circumstances we will consider. The main difference to our definition of cores in Section 5 is that we need only retain the relative values of ω to the nearest *block*, rather than keeping the exact (shifted) values. This is the reason for us introducing the concept of blocks (and block synchronizers), and it allows the range of possible cores to be cut down substantially.

DEFINITION 6.1. *Fix any $X \subseteq [n]$ and activation schedule ω. Let X_j be the elements of X which are active by the start of the block containing column j, i.e., $X_j = \{v \in X : s(v) \leq j\}$. Let j' be the smallest j such that $j - s(X) \geq \frac{B \cdot |X_j|}{r}$.*

For every v, define $\phi(v) = \frac{s(v)-s(X)}{B}$, i.e., $\phi(v)$ is the number of blocks that pass between the start column of X and the start column of v. Note that $\phi(v) \in \mathbb{N}$.

*The **core** $\mathcal{C}_{X,\omega}$ of a subset $X \subseteq [n]$ with respect to activation schedule ω is defined to be*

$$\{(v, \phi(v)) : v \in X, s(v) < j'\}$$

We see, as we did in Section 5, that if some object S "hits" all cores, then it hits all subsets of $[n]$ under any activation schedule. By hitting a core $\mathcal{C}$ at column j, we mean that $\sum_{(v,\phi(v))\in\mathcal{C}} S^v_{j-B\phi(v)} = 1$, and we assume column numberings start at the beginning of the core. So, if S hits a core $\mathcal{C}_{X,\omega}$ within $\frac{B\cdot|\mathcal{C}_{X,\omega}|}{r}$ columns, then it hits the set X within $\frac{B\cdot|X|}{r}$ columns of $s(X)$ under activation schedule ω.

We wish to prove the existence of a small block synchronizer by randomly generating a candidate S, and proving that it indeed has the required properties with positive probability, in a similar fashion to the proof of small radio synchronizers. While this could be achieved directly, we can in fact get a better result by proving existence of a slightly weaker object using this method, and then bridging the gap with selective families.

DEFINITION 6.2. *$S = \{S^v\}_{v\in[n]}$ is an (n, k, Δ, r, B)-**upper block synchronizer** if, for any core $\mathcal{C}$ with $k \leq |\mathcal{C}| \leq \Delta$, there exists column $j < \frac{B\cdot|\mathcal{C}|}{r}$ such that $\sum_{(v,\phi(v))\in\mathcal{C}} S^v_{j-B\phi(v)} = 1$.*

An upper block synchronizer has a lower bound k on the size of the cores it must hit. To obtain our full block synchronizer result, we will first show the existence of small upper block synchronizers, and then show that these can be extended to block synchronizers by adding selective families to hit cores of size less than k.

THEOREM 6.3. *For some constant c and for any n, D, Δ with $D, \Delta \leq n < D\Delta$, there exists an $(n, \frac{n}{D}, \Delta, \frac{n}{D}, c\frac{n}{D}\log D \log\log\frac{D\Delta}{n})$-upper block synchronizer.*

Proof. Let c be a constant to be chosen later. For simplicity of notation we now set $k = \frac{n}{D}$, $r = \frac{n}{D}$, and $B = c\frac{n}{D}\log D\log\log\frac{D\Delta}{n}$.

Define $\rho(j) = j \bmod 2\log\log\frac{D\Delta}{n}$. Our candidate upper block synchronizer $S = \{S^v\}_{v\in[n]}$ will be generated by independently choosing each S^v_j (for $j < \frac{nB}{r}$) to be $\mathbf{1}$ with probability $\frac{c\log D\log\log\frac{D\Delta}{n}}{(B+j)2^{\rho(j)+1}}$ and $\mathbf{0}$ otherwise.

We will analyze our candidate upper block synchronizer by fixing some particular core and bounding the probability that the candidate hits it. We begin by defining the *load* of a column (with respect to some fixed core $\mathcal{C}$), and bounding it both above and below on a subset of columns. As before, load represents expected number of $\mathbf{1}$s in a column, and we want it to be constant in order to maximize hitting probability. Recall that we now consider column numbering to begin at the start of the core, i.e. $\min_{(v,\phi(v))\in\mathcal{C}} \phi(v) = 0$.

Definition 6.4. *Let $\mathcal{C}(j)$ denote $\{(v,\phi(v)) \in \mathcal{C} : B\phi(v) \leq j\}$. The **load** of a column j of core $\mathcal{C}$, denoted $f_{\mathcal{C}}(j)$, is defined to be $\sum_{(v,\phi(v))\in\mathcal{C}(j)} \mathbf{Pr}[S^v_{j-B\phi(v)} = \mathbf{1}] = \sum_{(v,\phi(v))\in\mathcal{C}(j)} \frac{c\log D \log\log\frac{D\Delta}{n}}{(j-B\phi(v)+B)2^{\rho(j)+1}}$.*

Since load varies across a wide range during each $2\log\log\frac{D\Delta}{n}$-length "phase," we first consider only the columns at the start of each phase (i.e., those j with $\rho(j) = 0$), which we will call *0-columns*.

Lemma 6.5. *For all $\frac{B}{2} \leq j < \frac{B\cdot|\mathcal{C}|}{r}$ with $\rho(j) = 0$, $f_{\mathcal{C}}(j) > \frac{1}{6}$.*

Proof. Recall that, when deriving a core from a set X, we ended the core at the first column j' with $j' - s(X) \geq \frac{B\cdot|X_j|}{r}$, i.e. for all $j \leq j'-1$, $j - s(X) < \frac{B\cdot|X_j|}{r}$. Having shifted column numberings, this implies that for $j < \frac{B\cdot|\mathcal{C}|}{r}$, $j < \frac{B\cdot|\mathcal{C}(j)|}{r}$. The minimum contribution any$(v,\phi(v)) \in \mathcal{C}(j)$ can add to $f_{\mathcal{C}}(j)$ is $\frac{c\log D\log\log\frac{D\Delta}{n}}{2(j+B)}$. Therefore total load is upper bounded by

$$f_{\mathcal{C}}(j) \geq |\mathcal{C}(j)| \cdot \frac{c\log D\log\log\frac{D\Delta}{n}}{2(j+B)} > \frac{cj}{2c(j+B)} \geq \frac{1}{6}$$ □

This lemma provides a lower bound on $f_{\mathcal{C}}(j)$. We also need an upper bound, but we cannot obtain a good one for all j, since load in a particular column can be very large. We circumvent this issue by only bounding the load on a smaller set of columns.

Let $\mathcal{F}_{\mathcal{C}} = \{j < \frac{B\cdot|\mathcal{C}|}{r} : \rho(j) = 0, \frac{1}{6} < f_{\mathcal{C}}(j) < 3\log\frac{|\mathcal{C}|D}{n}\}$. We prove a lower bound on $|\mathcal{F}_{\mathcal{C}}|$.

Lemma 6.6. *If $\frac{n}{D} \leq |\mathcal{C}| \leq \Delta$, then $|\mathcal{F}_C| \geq \frac{c}{6}|\mathcal{C}|\log D$.*

Proof. We first upper bound the total load of all 0-columns j with $j < \frac{B\cdot|\mathcal{C}|}{r}$ and then show that not too many of these columns can have $f_{\mathcal{C}}(j) \geq 3\log\frac{|\mathcal{C}|D}{n}$, giving a lower bound for the number of 0-columns in $\mathcal{F}_{\mathcal{C}}$.

We bound the total load of all 0-columns j with $j < \frac{B\cdot|\mathcal{C}|}{r}$ as follows:

$$\sum_{\substack{j<\frac{B\cdot|\mathcal{C}|}{r}\\ \rho(j)=0}} f_{\mathcal{C}}(j) = \sum_{\substack{j<\frac{B\cdot|\mathcal{C}|}{r}\\ \rho(j)=0}} \sum_{(v,\phi(v))\in\mathcal{C}(j)} \frac{c\log D\log\log\frac{D\Delta}{n}}{2(j-B\phi(v)+B)}$$

$$= \sum_{(v,\phi(v))\in\mathcal{C}} \sum_{\substack{B\phi(v)\leq j<\frac{B\cdot|\mathcal{C}|}{r}\\ \rho(j)=0}} \frac{c\log D\log\log\frac{D\Delta}{n}}{2(j-B\phi(v)+B)}$$

(substitution of sum index variable)

$$= \sum_{(v,\phi(v))\in\mathcal{C}} \sum_{i=\frac{B\phi(v)}{2\log\log\frac{D\Delta}{n}}}^{\frac{B\cdot|\mathcal{C}|}{2r\log\log\frac{D\Delta}{n}}-1} \frac{c\log D\log\log\frac{D\Delta}{n}}{2(2i\log\log\frac{D\Delta}{n} - B\phi(v)+B)}$$

(using standard integral bound)

$$\leq \sum_{(v,\phi(v))\in\mathcal{C}} \int_{\frac{B\phi(v)}{2\log\log\frac{D\Delta}{n}}-1}^{\frac{B\cdot|\mathcal{C}|}{2r\log\log\frac{D\Delta}{n}}-1} \frac{c\log D\log\log\frac{D\Delta}{n}}{2(2i\log\log\frac{D\Delta}{n} - B\phi(v)+B)} di$$

(evaluating integral)

$$= \frac{c\log D}{4} \sum_{(v,\phi(v))\in\mathcal{C}} \ln\left(\frac{\frac{B\cdot|\mathcal{C}|}{r} - 2\log\log\frac{D\Delta}{n} - B\phi(v)+B}{B - 2\log\log\frac{D\Delta}{n}}\right)$$

$$\begin{aligned}
&\le \frac{c|\mathcal{C}|\log D}{4}\ln\left(\frac{\frac{B\cdot|\mathcal{C}|}{r}-2\log\log\frac{D\Delta}{n}+B}{B-2\log\log\frac{D\Delta}{n}}\right)\\
&= \frac{c|\mathcal{C}|\log D}{4}\ln\left(\frac{|\mathcal{C}|c\log D\log\log\frac{D\Delta}{n}-2\log\log\frac{D\Delta}{n}+B}{B-2\log\log\frac{D\Delta}{n}}\right)\\
&\le \frac{c|\mathcal{C}|\log D}{4}\ln\left(\frac{2(c|\mathcal{C}|\log D\log\log\frac{D\Delta}{n}+B)}{B}\right)\\
&= \frac{c|\mathcal{C}|\log D}{4}\ln\left(\frac{2(|\mathcal{C}|+\frac{n}{D})}{\frac{n}{D}}\right)
\end{aligned}$$

(using the assumption $\frac{n}{D}\le|\mathcal{C}|$)

$$\begin{aligned}
&\le \frac{1}{4}c|\mathcal{C}|\log D\ln\frac{4|\mathcal{C}|D}{n}\\
&\le \frac{1}{4}c|\mathcal{C}|\log D\log\frac{|\mathcal{C}|D}{n}
\end{aligned}$$

$\blacksquare$

Since for any $j<\frac{B\cdot|\mathcal{C}|}{r}$ we have $f_{\mathcal{C}}(j)>0$, the inequality above implies that there must be not more than $\frac{1}{12}c|\mathcal{C}|\log D$ 0-columns with $f_{\mathcal{C}}(j)\ge 3\log\frac{|\mathcal{C}|D}{n}$. By Lemma 6.5, the number of columns j with $j<\frac{B\cdot|\mathcal{C}|}{r}$ for which $f_{\mathcal{C}}(j)\le\frac{1}{6}$ is at most $\frac{B}{2}$, and hence the number of such 0-columns is at most $\frac{B}{4\log\log\frac{D\Delta}{n}}$. Therefore, $|\mathcal{F}_{\mathcal{C}}|$, which is the number of 0-columns j with $j<\frac{B\cdot|\mathcal{C}|}{r}$ for which $\frac{1}{6}<f_{\mathcal{C}}(j)<3\log\frac{|\mathcal{C}|D}{n}$, is upper bounded as follows:

$$\begin{aligned}
|\mathcal{F}_{\mathcal{C}}| &\ge \frac{B\cdot|\mathcal{C}|}{2r\log\log\frac{D\Delta}{n}}-\frac{B}{4\log\log\frac{D\Delta}{n}}-\frac{1}{12}c|\mathcal{C}|\log D\\
&= \frac{c}{2}\log D\left(|\mathcal{C}|-\frac{n}{2D}-\frac{|\mathcal{C}|}{6}\right)\ge\frac{c}{6}|\mathcal{C}|\log D
\end{aligned}$$

where the last inequality follows from our assumption that $\frac{n}{D}\le|\mathcal{C}|$. $\square$

With the bound of the load of 0-columns in Lemma 6.6, we can obtain a significantly tighter bound on a subset of all columns.

Let $\mathbb{F}_{\mathcal{C}}=\{j<\frac{B\cdot|\mathcal{C}|}{r}:\frac{1}{6}<f_{\mathcal{C}}(j)\le 2\}$.

LEMMA 6.7. *For any $\mathcal{C}$ with $\frac{n}{D}\le|\mathcal{C}|\le\Delta$, $|\mathbb{F}_{\mathcal{C}}|\ge\frac{c}{12}|\mathcal{C}|\log D$.*

Proof. We show that, whenever we have a 0-column with load in the range $(\frac{1}{6},3\log\frac{|\mathcal{C}|D}{n})$, there must be some column within the same phase for which load is in the range $(\frac{1}{6},2)$.

For any $j\in\mathcal{F}_{\mathcal{C}}$, let $j'=j+\log f_{\mathcal{C}}(j)-1$. Then,

$$j'<j+\log(3\log\frac{|\mathcal{C}|D}{n})-1<j+2\log\log\frac{D\Delta}{n}$$

so j' is in the same phase as j (i.e., $j-\rho(j)=j'-\rho(j')$). Hence,

$$
\begin{aligned}
f_{\mathcal{C}}(j') &= \sum_{(v,\phi(v))\in\mathcal{C}(j')} \frac{c\log D \log\log\frac{D\Delta}{n}}{(j'-B\phi(v)+B)2^{\rho(j')+1}}\\
&= \sum_{(v,\phi(v))\in\mathcal{C}(j)} \frac{c\log D \log\log\frac{D\Delta}{n}}{(j'-B\phi(v)+B)2^{\rho(j)+\log f_{\mathcal{C}}(j)}}\\
&= \sum_{(v,\phi(v))\in\mathcal{C}(j)} \frac{c\log D \log\log\frac{D\Delta}{n}}{(j'-B\phi(v)+B)f_{\mathcal{C}}(j)}\\
&= \frac{2}{f_{\mathcal{C}}(j)}\sum_{(v,\phi(v))\in\mathcal{C}(j)} \frac{c\log D \log\log\frac{D\Delta}{n}}{2(j-B\phi(v)+B)}\cdot\frac{(j-B\phi(v)+B)}{(j'-B\phi(v)+B)}
\end{aligned}
$$

Since, for any $(v,\phi(v)) \in \mathcal{C}(j)$, $\frac{1}{3} < \frac{1}{1+\frac{2\log\log\frac{D\Delta}{n}}{B}} \leq \frac{(j-B\phi(v)+B)}{(j'-B\phi(v)+B)} \leq 1$, we can bound $f_{\mathcal{C}}(j')$ from above:

$$f_{\mathcal{C}}(j') \leq \frac{2}{f_{\mathcal{C}}(j)}\sum_{(v,\phi(v))\in\mathcal{C}(j)} \frac{c\log D \log\log\frac{D\Delta}{n}}{2(j-B\phi(v)+B)}\cdot 1 = 2$$

and below:

$$f_{\mathcal{C}}(j') > \frac{2}{f_{\mathcal{C}}(j)}\sum_{(v,\phi(v))\in\mathcal{C}(j)} \frac{c\log D \log\log\frac{D\Delta}{n}}{2(j-B\phi(v)+B)}\cdot \frac{1}{3} = \frac{2}{3}$$

(The reason we allow loads to be as low as $\frac{1}{6}$ in the definition of $\mathbb{F}_{\mathcal{C}}$ is to account for cases where $f_{\mathcal{C}}(j) \leq 2$ and so $j' = j$.)

Therefore $j' \in \mathbb{F}_{\mathcal{C}}$. This mapping of j to j' is an injection from $\mathcal{F}_{\mathcal{C}}$ to $\mathbb{F}_{\mathcal{C}}$, and so $|\mathbb{F}_{\mathcal{C}}| \geq |\mathcal{F}_{\mathcal{C}}| \geq \frac{c}{12}|\mathcal{C}|\log D$. ∎

Now that we have proven that sufficiently many columns have loads within a constant-size range, we want to show that S has a good probability of hitting $\mathcal{C}$ on these columns. To do so, we again apply Lemma 5.5, setting $x_v = \mathcal{S}^v_{j-B\phi(v)}$, and see that the probability of S hitting $\mathcal{C}$ on column j is at least $f_{\mathcal{C}}(j)\cdot 4^{-f_{\mathcal{C}}(j)}$

Lemma 6.8. *For any core $\mathcal{C}$ with $\frac{n}{D} \leq |\mathcal{C}| \leq \Delta$, with probability at least $1 - D^{-\frac{c|\mathcal{C}|}{63}}$ there is a column $j < \frac{B\cdot|\mathcal{C}|}{r}$ on which S hits $\mathcal{C}$.*

Proof. Let us first recall that $\mathbb{F}_{\mathcal{C}} = \{j < \frac{B\cdot|\mathcal{C}|}{r} : \frac{1}{6} < f_{\mathcal{C}}(j) \leq 2\}$, and note that function $h(x) = 1 - x4^{-x}$ for $\frac{1}{6} \leq x \leq 2$ is maximized at $x = 2$, with $h(2) = \frac{7}{8}$.

Each column j independently hits C with probability at least $f_{\mathcal{C}}(j)\cdot 4^{-f_{\mathcal{C}}(j)}$, so the probability that none hit is bounded by:

$$
\begin{aligned}
\mathbf{Pr}[\text{no column hits}] &\leq \prod_{j<\frac{B\cdot|\mathcal{C}|}{r}}(1 - f_{\mathcal{C}}(j)\cdot 4^{-f_{\mathcal{C}}(j)}) \leq \prod_{j\in\mathbb{F}_{\mathcal{C}}}(1 - f_{\mathcal{C}}(j)\cdot 4^{-f_{\mathcal{C}}(j)})\\
&\leq \prod_{j\in\mathbb{F}_{\mathcal{C}}}\frac{7}{8} \leq \left(\frac{7}{8}\right)^{\frac{c}{12}|\mathcal{C}|\log D} = D^{-\frac{c}{12}|\mathcal{C}|\log\frac{7}{8}} \leq D^{-\frac{c\cdot|\mathcal{C}|}{63}}
\end{aligned}
$$

where the penultimate inequality follows from Lemma 6.7. ∎

We have a bound on the probability of hitting a particular core, but before we can show that we can hit all of them, we must count the number of possible cores.

Let $\mathcal{C}_q$ be the set of possible cores of size q.

LEMMA 6.9. $|\mathcal{C}_q| \leq D^{2q}$.

Proof. For any $(v, \phi(v)) \in \mathcal{C}$, $B\phi(v) < \frac{B|C|}{r}$, i.e., for a core of size q, $\phi(v) < \frac{q}{r}$. Therefore there are at most $n \cdot \frac{q}{r}$ possible pairs of $(v, \phi(v))$, and thus at most $\binom{n \cdot \frac{q}{r}}{q}$ ways of choosing a size-q subset. So, $|\mathcal{C}_q|$ is at most $\binom{nq/r}{q} = \binom{Dq}{q} \leq (eD)^q \leq D^{2q}$. ∎

We are now ready to prove the existence of a small upper block synchronizer:

LEMMA 6.10. *With positive probability, S is an $(n, \frac{n}{D}, \Delta, \frac{n}{D}, c\frac{n}{D}\log D \log\log \frac{D\Delta}{n})$-upper block synchronizer.*

Proof. We will set c to be 189. By union bound,

$$\begin{aligned}
\mathbf{Pr}[S \text{ is not an upper block synchronizer}] &\leq \sum_{q=\frac{n}{D}}^{\Delta} \sum_{C \in C_q} \mathbf{Pr}[\text{C is not hit}] \\
&\leq \sum_{q=\frac{n}{D}}^{\Delta} \sum_{C \in C_q} D^{-cq/63} \leq \sum_{q=\frac{n}{D}}^{\Delta} D^{2q} D^{-cq/63} = \sum_{q=\frac{n}{D}}^{\Delta} D^{2q} D^{-3q} \\
&= \sum_{q=\frac{n}{D}}^{\Delta} D^{-q} < \frac{2}{D} < 1
\end{aligned}$$
∎

Since, with positive probability, our candidate S is an $(n, \frac{n}{D}, \Delta, \frac{n}{D}, c\frac{n}{D}\log D \log\log \frac{D\Delta}{n})$-upper block synchronizer, at least one such object must exist, and so we have completed our proof of Theorem 6.3. ∎

We can now prove Theorem 2.8:

Proof. We construct block synchronizer $\mathcal{S}$ by taking an $(n, \frac{n}{D}, \Delta, \frac{n}{D}, c\frac{n}{D}\log D \log\log \frac{D\Delta}{n})$-upper block synchronizer S and inserting an $(n, \frac{n}{D})$-selective family R of size $\tilde{c}\frac{n}{D}\log D \log\log \frac{D\Delta}{n}$ at the beginning of each block (we know by Lemma 2.2 that a selective family of size $\tilde{c}\frac{n}{D}\log D$ exists, and we can pad it arbitrarily to this larger size). That is, our block size will now be $\mathcal{B} := |R| + B = (c + \tilde{c})\frac{n}{D}\log D \log\log \frac{D\Delta}{n}$, and our block synchronizer $\mathcal{S}$ will be formally defined by:

$$\mathcal{S} = \{\mathcal{S}^v\}_{v \in [n]} \text{ is defined by } \mathcal{S}^v_j = \begin{cases} R^v_{j \bmod \mathcal{B}} & \text{if } (j \bmod \mathcal{B}) < |R|, \\ S^v_{j - \lceil \frac{j}{\mathcal{B}} \rceil R} & \text{otherwise.} \end{cases}$$

Setting $\hat{c} = c + \tilde{c}$, we show that $\mathcal{S}$ satisfies the conditions of an $(n, \Delta, \frac{n}{D}, \hat{c}\frac{n}{D}\log D \log\log \frac{D\Delta}{n})$-block synchronizer.

Let $\mathcal{C}$ be a core of size at most Δ.

Case 1: $|\mathcal{C}| \leq \frac{n}{D}$**.** $\forall (v, \phi(v)) \in C$ we have $\phi(v) = 0$, since the core ends before column $\mathcal{B}$ by Definition 6.1, and so C will be hit by the $(n, \frac{n}{D})$-selective family R. It will therefore be hit by $\mathcal{S}$ on some column $j < |R| < \mathcal{B} = \mathcal{B}\lceil \frac{|C|}{r} \rceil$. Note that this case is the reason we require the ceiling function in the definition of a block synchronizer, but not in an upper block synchronizer.

Case 2: $|\mathcal{C}| > \frac{n}{D}$**.** If $|\mathcal{C}| > \frac{n}{D}$, then it will be hit by a column $j < \frac{B \cdot |C|}{r}$ in the upper block synchronizer S, which corresponds to the column $j + \lceil \frac{j}{B} \rceil |R|$ in $\mathcal{S}$. Since

$j + \lceil \frac{j}{B} \rceil |R| < \frac{B \cdot |C|}{r} + \lceil \frac{|C|}{r} \rceil |R| \leq (B+R) \lceil \frac{|C|}{r} \rceil = \mathcal{B} \lceil \frac{|C|}{r} \rceil$, this satisfies the block synchronizer property.

So, $\mathcal{S}$ hits all cores $\mathcal{C}$ with $|\mathcal{C}| < \Delta$ within $\mathcal{B} \lceil \frac{|C|}{r} \rceil$ columns, and therefore hits all sets X within $\mathcal{B} \lceil \frac{|X|}{r} \rceil$ under any activation schedule, fulfilling the criteria of an $(n, \Delta, \frac{n}{D}, \hat{c} \frac{n}{D} \log D \log \log \frac{D\Delta}{n})$-block synchronizer. ∎

7. Conclusions. The task of broadcasting in radio networks is a longstanding, fundamental problem in communication networks. Our result for deterministic broadcasting in directed networks combines elements from several of the previous works with some new techniques, and, in doing so, makes a significant improvement to the fastest known running time. Our algorithm for wake-up also improves over the previous best running time, in both directed and undirected networks, and relies on a proof of smaller universal synchronizers, a combinatorial object first defined in [4].

Neither of these algorithms are known to be optimal. The best known lower bound for both broadcasting and wake-up is $\Omega(\min(n \log D, D\Delta \log \frac{n}{\Delta}))$ [11]; our broadcasting algorithm therefore comes within a log-logarithmic factor, but our wake-up algorithm remains a logarithmic factor away.

As well as the obvious problems of closing these gaps, there are several other open questions regarding deterministic broadcasting in radio networks. Firstly, the lower bound for undirected networks is weaker than that for directed networks [21], and so one avenue of research would be to find an $\Omega(n \log D)$ lower bound in undirected networks, matching the broadcasting time of [19]. Secondly, the algorithms given here, along with almost all previous work, are non-explicit, and therefore it remains an important challenge to develop explicit algorithms that can come close to the existential upper bound. The best constructive algorithm known to date is by [16], but it is a long way from optimality.

Some variants of the model also merit interest, in particular the model with collision detection. It is unknown whether the capacity for collision detection improves deterministic broadcast time, as it does for randomized algorithms [14]. Collision detection does remove the requirement of spontaneous transmissions for the use of the $O(n)$ algorithm of [6], but a synchronized global clock would still be required. It should be noted that collision detection renders the wake-up problem trivial, since if every active node transmits in every time-step, collisions will wake up the entire network within D time-steps.

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) (1991), pp. 290–298.

[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) (1992), pp. 104–126.

[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), 2005, pp. 347–359.

[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), 2004, pp. 266–274.

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

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

[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), 2000, pp. 717–728.

[8] B. S. Chlebus, D. R. Kowalski, and A. Pelc, *Electing a leader in multi-hop radio networks*, in Proceedings of the 16th International Conference on Principles of Distributed Systems (OPODIS), 2012, pp. 106–120.

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

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

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

[12] 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), 2003, pp. 492–501.

[13] M. Ghaffari and B. Haeupler, *Near optimal leader election in multi-hop radio networks*, in Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2013, pp. 748–766.

[14] 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), 2013, pp. 325–334.

[15] 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), 2016, pp. 361–370, https://doi.org/10.1145/2933057.2933121.

[16] 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), 2002, pp. 697–704.

[17] T. Jurdziński and D. R. Kowalski, *The wake-up problem in multi-hop radio networks*, in Encyclopedia of Algorithms, M. Y. Kao, ed., Springer-Verlag, Berlin, 2015.

[18] T. Jurdziński and G. Stachowiak, *Probabilistic algorithms for the wakeup problem in single-hop radio networks*, in Proceedings of the International Symposium on Algorithms and Computation (ISAAC 2002), 2002, pp. 535–549.

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

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

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

[22] E. Kushilevitz and Y. Mansour, *An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks*, SIAM Journal on Computing, 27(3) (1998), pp. 702–712.

[23] G. D. Marco, *Distributed broadcast in unknown radio networks*, SIAM Journal on Computing, 39(6) (2010), pp. 2162–2175.

[24] G. D. Marco and A. Pelc, *Faster broadcasting in unknown radio networks*, Information Processing Letters, 79 (2001), pp. 53–56.

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

[26] S. Vaya, *Information dissemination in unknown radio networks with large labels*, Theoretical Computer Science, 520 (2014), pp. 11–26.