Upper Bounds on Broadcasting Time in UDG Radio Networks with Unknown Topology

Yuval Emek[‡] Leszek Gąsieniec [*] Erez Kantor[‡] Andrzej Pelc [†]
David Peleg[‡] Chang Su [*]

Abstract

The paper considers broadcasting in radio networks, modeled as unit disk graphs (UDG). Network stations are modeled as points in the Euclidean plane, where a station is connected to all stations at distance at most 1 from it. A message transmitted by a station reaches all its neighbors, but a station *hears* a message (receives the message correctly) only if exactly one of its neighbors transmits at a given time step. One station of the network, called the *source*, has a message which has to be disseminated to all other stations. Stations are unaware of the network topology. Two broadcasting models are considered. In the *conditional wake up* model, the stations other than the source are initially idle and cannot transmit until they receive a message for the first time. In the *spontaneous wake up* model, all stations are awake (and may transmit messages) from the beginning.

It turns out that broadcasting time depends on two parameters of the UDG, namely, its diameter D and its *granularity* g, which is the inverse of the minimum distance between any two stations. We present a deterministic broadcasting algorithm which works in time $O(Dg)$ under the conditional wake up model. For the spontaneous wake up model, we design two deterministic broadcasting algorithms: the first works in time $O(D + g^2)$ and the second in time $O(D \log g)$. We prove that a combination of these two algorithms accomplishes broadcasting in time $O\left(\min\left\{D + g^2, D \log g\right\}\right)$.

---

[*] Department of Computer Science, The University of Liverpool, Chadwick Building, Peach Street, Liverpool L69 7ZF, UK. E-mail:`leszek@csc.liv.ac.uk`

[†] Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mail: `pelc@uqo.ca`. Research partially supported by NSERC discovery grant and by the Research Chair in Distributed Computing at the Université du Québec en Outaouais.

[‡] Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: {`yuval.emek,erez.kantor,david.peleg`}`@weizmann.ac.il`. Supported in part by a grant from the Israel Ministry of Science.

1 Introduction

The model and the problem: A radio network consists of stations, each of which can act in a given time step either as a *transmitter* or as a *receiver.* The network is modeled as a *unit disk graph* (UDG) whose nodes are the stations. These nodes are represented as points in the Euclidean plane. Two nodes are joined by an edge if their distance is at most 1. Such nodes are called *neighbors.* It is assumed that transmitters of stations have power which enables them to transmit at distance 1. Hence the existence of an edge between two nodes indicates that transmissions of one of them can reach the other, i.e., these nodes can communicate directly. We will refer to radio networks modeled by unit disk graphs as UDG radio networks.

In a radio network, a node acting as a transmitter in a given time step sends a message which is delivered to all of its neighbors in the same time step. An important distinction at the receiving end is between a message being just *delivered* and being *heard*, i.e., received successfully by a node. A node acting as a receiver in a given step *hears* a message if and only if a message from exactly one of its neighbors is delivered in this step. The message heard in this case is the one that was delivered from the unique neighbor. If messages from at least two neighbors v and v' of u are delivered simultaneously in a given step, none of the messages is heard by u in this step. In this case we say that a *collision* occurred at u. It is assumed that the effect at node u of a collision is the same as that of no message being delivered in this step, i.e., a node cannot distinguish a collision from silence.

We assume that each node of the network knows only its own coordinates in the Euclidean plane and a parameter d representing the minimum distance between two stations. This latter parameter is fairly natural and characterizes the physical nature of the set of stations the network consists of. For example, if stations are sensors of the form of disks of radius r and the coordinates of a station are coordinates of the center of the sensor, then it is natural to take d to be about $2r$ to minimize overlapping. In fact, our results remain unchanged if nodes know only a linear lower bound on d. Nodes are unaware of the topology of the network and have no knowledge of other parameters of the network, such as the diameter or the size. They are not even aware of their immediate neighborhood. Such networks are often called *ad hoc* networks.

We consider *broadcasting*, which is the following basic communication task. In the beginning, one distinguished node, called the *source*, has a message which has to be transmitted to all other nodes. Remote nodes get the source message via intermediate nodes, along paths in the network. We distinguish between two broadcasting models. In the *conditional wake* *up* model, the stations other than the source are initially idle and cannot transmit until they receive the source message for the first time (and subsequently wake up). In the *spontaneous wake up* model, all stations are assumed to be awake when the source transmits for the first time, and may contribute to the broadcasting process by transmitting control messages even before they received the source message. All nodes have individual clocks that tick at the same rate, measuring time steps, referred to as *rounds*. In the conditional wake up model, the clock of a node starts in the round when the node first receives the source message. In the spontaneous wake up model, the clocks of all nodes start simultaneously, in the round when the source transmits for the first time.

The task of broadcasting in the conditional wake up model can be interpreted as activating the network from a single source, and is related to the task of waking up the network. In this latter task, some nodes spontaneously wake up and have to wake up other nodes by sending messages. Thus broadcasting in the conditional wake up model, i.e., activating the network from a single source, is equivalent to waking up the network when exactly one node (the source) wakes up spontaneously. The broadcasting models with spontaneous wake up and conditional wake up have also been called broadcasting with and without spontaneous transmissions, respectively [23, 24].

We consider only deterministic broadcasting algorithms and do not assume any central authority monitoring the broadcasting process. Thus the decision made by a node whether to transmit or to receive in a given round, and what message to transmit, if any (some control messages can be transmitted on their own or be appended to the source message) is based solely on the coordinates of the node and on the messages received by it so far. The execution time of a broadcasting algorithm in a given radio network is the number of rounds it takes since the first transmission until all nodes of the network hear the source message.

Our results: The focus of this paper is on the design of fast broadcasting algorithms working in arbitrary UDG radio networks with unknown topology, and the analysis of the execution time of such algorithms. It turns out that the execution time of broadcasting algorithms depends on two parameters of the network. One of them is the *diameter* of the network, denoted by D: this is the maximum length (in hops) of a shortest path in the graph between any two nodes of the network. (The diameter of a UDG network should not be confused with the diameter of the set of points representing its nodes, i.e., the largest Euclidean distance between any two such points: the latter can be much smaller than D.) The other parameter is the *granularity* of the network, denoted by g. This is the inverse of the minimum Euclidean distance d between any two nodes of the network. Hence networks of large granularity are those that have some nodes close to each other. Our upper bounds on the optimal broadcasting time are increasing functions of D and g. It should be noted that, since nodes know (a linear lower bound on) d, they know (a linear upper bound on) g. However, we do not assume any knowledge of D.

For the conditional wake up model, we show an algorithm that completes broadcast in time $O(Dg)$ in any UDG radio network of diameter D and granularity g. An $\Omega(D\sqrt{g})$ lower bound is presented in the companion paper [15]. For the spontaneous wake up model, we show two broadcasting algorithms, one working in time $O(D+g^2)$ and the other in time $O(D\log g)$. These algorithms are based on completely different ideas and, depending on parameter values, one or the other may be more efficient. The combined algorithm obtained by interleaving these two algorithms completes broadcast in time $O\left(\min\left\{D+g^2, D\log g\right\}\right)$. A matching lower bound of $\Omega\left(\min\left\{D+g^2, D\log g\right\}\right)$ is presented in [15].

Related work: In most of the papers concerning algorithmic aspects of radio communication, the radio network was modeled as an arbitrary (directed or undirected) graph. This literature can be divided into two subareas, one dealing with centralized communication, in which it is assumed that nodes have complete knowledge of the network topology, and hence can simulate a central transmission scheduler (cf. [1, 4, 5, 14, 17, 19, 25]), and the other assuming only limited (usually local) knowledge of topology and studying distributed communication in such networks. The current paper belongs to the latter subarea.

The first paper to study deterministic centralized broadcasting in radio networks, assuming complete knowledge of the network, was [4]. The authors also formulated the model of radio network subsequently used by many researchers. In [5], a $\mathcal{O}(D\log^2 n)$-time broadcasting algorithm was given for all n-node networks of diameter D. In [17], $\mathcal{O}(D+\log^5 n)$-time broadcasting was proposed. This was improved to $\mathcal{O}(D+\log^4 n)$ in [14], then to $\mathcal{O}(D+\log^3 n)$ in [19], and very recently to $\mathcal{O}(D+\log^2 n)$ in [25]. The latter complexity is optimal. On the other hand, in [1] the authors proved the existence of a family of n-node networks of radius 2, for which any broadcast requires time $\Omega(\log^2 n)$.

The study of deterministic distributed broadcasting in radio networks whose nodes have only limited knowledge of the topology was initiated in [2]. The authors assumed that nodes know only their own label and labels of their neighbors. Many authors [3, 6, 7, 10, 11] studied deterministic distributed broadcasting in radio networks under the assumption that nodes know only their own label (but not labels of their neighbors), and that the topology of the network is unknown. Such networks are often called ad hoc. In [6] the authors gave a broadcasting algorithm working in time $\mathcal{O}(n)$ for arbitrary n-node networks, assuming that nodes can transmit spontaneously, before getting the source message. For this model, a matching lower bound $\Omega(n)$ on deterministic broadcasting time was proved in [23] even for the class of networks of constant radius. On the other hand, in [3] a lower bound $\Omega(D \log n)$ was proved for n-node networks of diameter D, if spontaneous transmissions are not allowed.

In [6, 7, 10, 12] the model of directed graphs was used. The aim of these papers was to construct broadcasting algorithms working as fast as possible in arbitrary (directed) radio networks without knowing their topology. The currently fastest deterministic broadcasting algorithm for such networks is the $\mathcal{O}(n \log^2 D)$-time algorithm from [12]. On the other hand, in [11] a lower bound $\Omega(n \log D)$ on broadcasting time was proved for directed n-node networks of diameter D.

The first papers to study randomized broadcasting algorithms in radio networks were [2, 27]. The authors do not assume that nodes know the topology of the network or that they have distinct labels. In [2] the authors showed a randomized broadcasting algorithm running in expected time $\mathcal{O}(D \log n + \log^2 n)$. In [27] it was shown that for any randomized broadcasting algorithm and parameters $D \leq n$, there exists an n-node network of diameter D requiring expected time $\Omega(D \log(n/D))$ to execute this algorithm. It should be noted that the lower bound $\Omega(\log^2 n)$ from [1], for some networks of radius 2, holds for randomized algorithms as well. A randomized algorithm working in expected time $\mathcal{O}(D \log(n/D) + \log^2 n)$, and thus matching the above lower bounds, was presented in [24] (cf. also [12]).

The wakeup problem in radio networks was first studied in [18] for single-hop networks (modeled by complete graphs), and then in [9, 8] for arbitrary networks. In [20] the authors studied randomized wakeup algorithms for radio networks. In all these papers it was assumed that a subset of all nodes wake up spontaneously (possibly at different times) and have to wake up other (dormant) nodes.

Another model of radio networks is based on geometric positions in the plane of the points representing stations. The underlying graph is no more arbitrary. It may be a unit disk graph, or its generalization, where radii of disks representing reachability areas may differ from node to node [13], or reachability areas may be of shapes different than a disk [16, 26]. Broadcasting in such geometric radio networks and some of their variations was considered, e.g., in [13, 16, 26, 30, 31]. In [31] the authors proved that scheduling optimal broadcasting is NP-hard even when restricted to such graphs, and gave an $O(n \log n)$ algorithm to schedule an optimal broadcast when nodes are situated on a line. In [30] broadcasting was considered in networks with nodes randomly placed on a line. In [26] the authors discussed fault-tolerant broadcasting in radio networks arising from regular locations of nodes on the line and in the plane, with reachability regions being squares and hexagons, rather than circles. In

[16] broadcasting with restricted knowledge was considered but the authors studied only the special case of nodes situated on the line. The first paper to study deterministic broadcasting in arbitrary geometric radio networks with restricted knowledge of topology was [13]. The authors studied several models, also assuming a positive knowledge radius, i.e., the knowledge available to a node, concerning other nodes inside some disk. In the case of knowledge radius 0, corresponding to our present scenario, they showed a broadcasting algorithm for the spontaneous wake up model working in time linear in the number of nodes, assuming that nodes are labeled by consecutive integers. It should be noted that in our case the total number of nodes in a network of diameter D and granularity g may be as large as $O(D^2g^2)$, hence the algorithm from [13] is much slower than ours.

Modeling ad hoc radio networks by unit disk graphs and their generalizations has recently attracted growing attention. In [28] this model was used for studying distributed solutions of the maximum independent set problem, in [29] of the coloring problem, and in [21] the convergecast problem was studied in geometric radio networks with varying reachability radii.

2 Broadcasting in the conditional wake up model

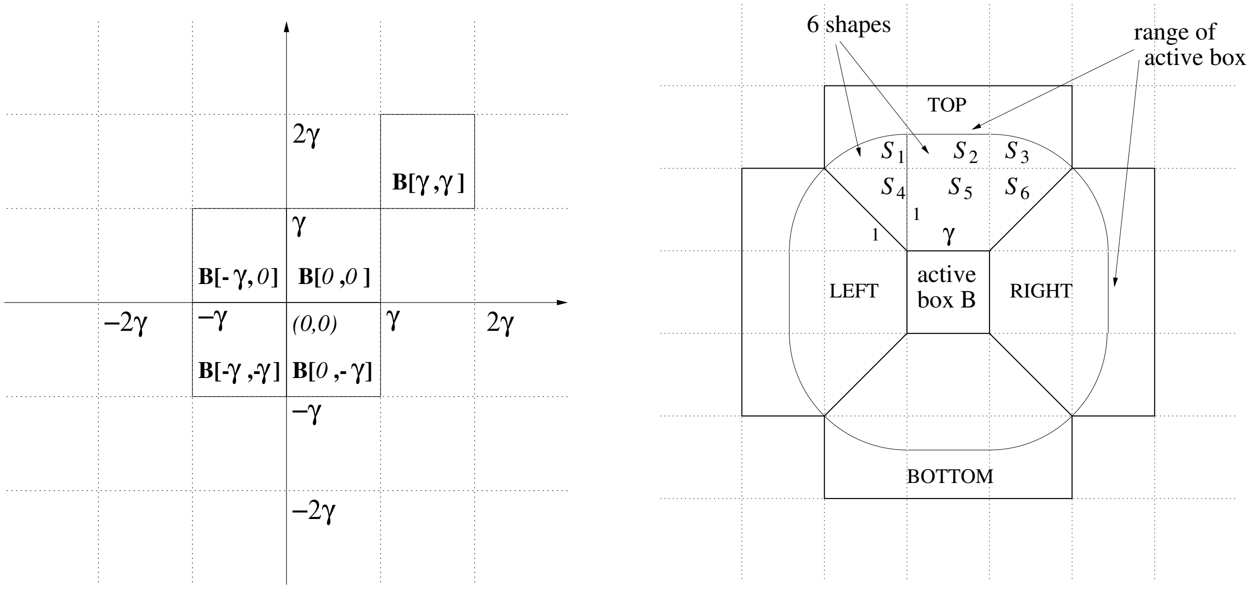
In this section we address the problem of broadcasting in UDG radio networks assuming that stations may transmit only after receiving the source message for the first time. A broadcasting algorithm for UDG radio networks that works in time $O(Dg)$ is presented. Our algorithm relies on the procedure `Echo` proposed in [22], and new notions of *a grid of boxes*, *a border* and *an effective border*, to be defined later in this section. We begin in Section 2.1 by presenting an $O(Dg \log g)$-time broadcasting algorithm. The structure of this algorithm is geared towards facilitating our improved $O(Dg)$-time algorithm, presented in Section 2.2.

Recall that every network node has unique (x, y) coordinates and it is aware of this fact at any time of the communication process. We say that a node becomes *informed* on the first receipt of the broadcast message. Otherwise, the node stays uninformed. Initially, only the source node s is informed.

2.1 $O(Dg \log g)$ broadcasting algorithm

Our solution uses extensively the notion of a *grid of boxes*. The entire 2-dimensional space can be partitioned into square *boxes*, each of size $c \times c$. The left and bottom edges of each box are assumed to be closed (i.e., points on these edges belong to the box), while the top and right edges are open. The boxes form an infinite grid G_c, where each *box* is identified by the location ((x, y) coordinates) of its bottom left corner and $1/c$ is called the *precision* of the grid. In general, for any two integers i and j, the corners of the box $B[i \cdot c, j \cdot c]$ are located in points (ic, jc), $(ic, (j+1)c)$, $((i+1)c, (j+1)c)$ and $((i+1)c, jc)$.

Fix $\gamma = 1/\sqrt{2}$. The grid G_γ, referred to later as the *pivotal* grid, plays a central role in our broadcasting algorithm. Note that $\frac{1}{c} = \sqrt{2}$ is the worst possible precision of any grid G_c with the property that all nodes occupying the same box can communicate directly with each other. Each box on the pivotal grid has a *transmission range*, which is defined as a maximal area around the box that can be reached by transmissions coming from inside the box. In other words, the range is a set of points located at distance at most 1 from the box, see Figure 1(b).



(a)

(b)

Figure 1: (a) the pivotal grid, and (b) the transmission range

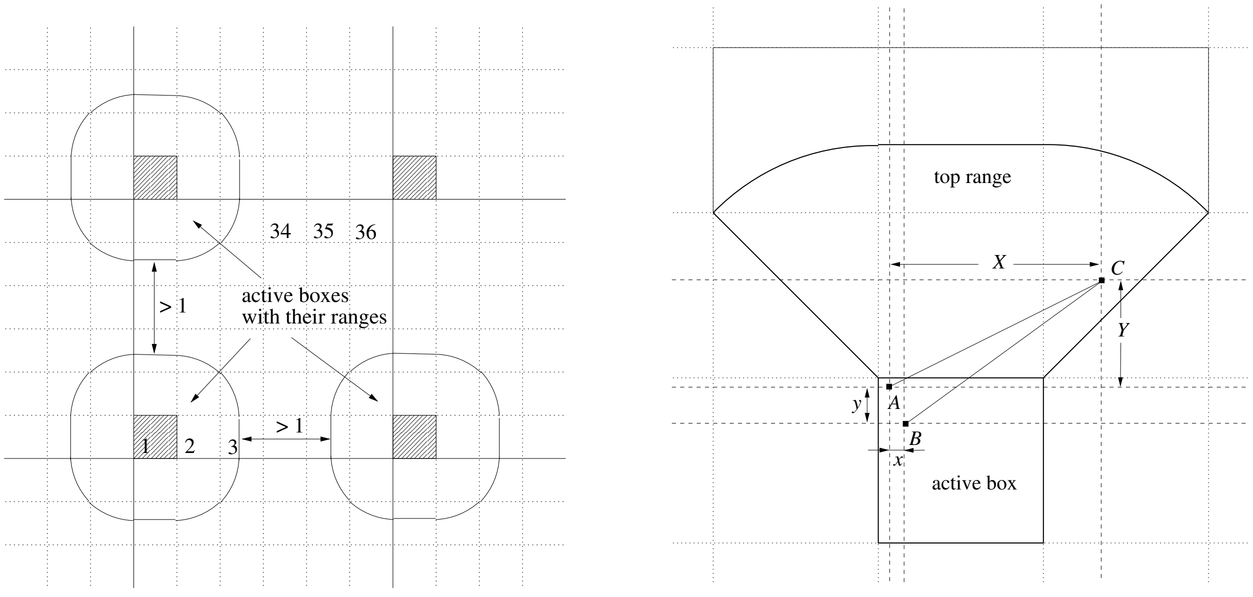
We say that a box in the pivotal grid has *a leader* if all nodes in the box are informed and they are aware of the choice of some distinguished node as the leader. Further, any box in the pivotal grid is allowed to become *active* according to a *transmission pattern* to be defined shortly. Each box becomes active only once, and this happens on the first occasion when the following two conditions are satisfied:

(1) the box is allowed to become active according to the transmission pattern, and

(2) the box has a leader.

The main purpose of the activation process is to inform all nodes that reside within the range of the box and to select leaders in boxes occupied by the newly informed nodes.

The transmission pattern used in our broadcasting procedure is defined as a periodic sequence of *stages*. During each stage, only a certain collection of boxes is allowed to become active, to avoid collisions that may be caused by transmissions occurring in different active boxes and their ranges. The transmission pattern is based on distant active boxes, see Figure 2(a), and its period is 36, i.e., each box has a chance to become active during every 36^{th} stage. More precisely, the box $B[i \cdot \gamma, j \cdot \gamma]$ is allowed (depending on whether the leader is already selected) to become active in stage k iff $(i = k) \bmod 6$ and $(j = \lfloor k/6 \rfloor) \bmod 6$. It is evident that the proposed transmission pattern (based on distant boxes whose ranges are at distance more than 1 apart, see Figure 2(a)) prevents the possibility of collisions between transmissions coming from different active boxes.



(a) (b)

Figure 2: (a) Active boxes and their ranges, and (b) node A dominates node B w.r.t. the top range

Each *stage* is divided into four *phases*. During each phase, the nodes of the active box attempt to communicate (pass on the broadcast message and help to find a leader) with the nodes from one of the four parts of its range, referred to as its *top, right, bottom* and *left* range, see Figure 1(b). Each part of the range overlaps with six boxes of the grid. The overlapping parts form six different *shapes* $S_1, .., S_6$, see Figure 1(b). We focus in this section on the communication performed between the nodes in the active box and its top range. The communication with nodes from the right, bottom and left ranges is analogous.

We need the following definition. For any two nodes A and B in the active box, we say that A *dominates* B with respect to the top range if any node in the top range reachable from B is also reachable from A. We have the following lemma.

Lemma 2.1. *Let $A = (x_A, y_A)$ and $B = (x_B, y_B)$ be two nodes in the active box and define $x = |x_A - x_B|$ and $y = y_A - y_B$. If $y > 4x$, then A dominates B.*

Proof: Assume that $C = (x_C, y_C)$ is a node belonging to the top range of the active box and let $X = |x_A - x_C|$ and $Y = y_C - y_A$, see Figure 2(b). Suppose, towards contradiction, that $x/y < 1/4$ (as in the premise of the lemma), yet C is reachable from B but not from A, i.e.,

$$X^2 + Y^2 \; > \; 1 \; > \; (X - x)^2 + (Y + y)^2 \; . \tag{1}$$

We argue that since C is outside of the transmissions range of A, necessarily $X < 4Y$. To see this, note that the ratio Y/X is minimized when A is positioned in the top-left corner of the active box, C is positioned in the right border of the top range and the distance between A and C is exactly 1. One can show that in this border case $\frac{Y}{X} = \frac{\sqrt{3}-1}{\sqrt{3}+1} > \frac{1}{4}$.

Note that Inequality (1) implies that $2Xx - 2Yy > x^2 + y^2$. Since the right hand side of the inequality is nonnegative, we also conclude that $2Xx > 2Yy$, which implies that $\frac{x}{y} > \frac{Y}{X}$. Contradiction now follows by the lemma's premise and by the fact that $Y/X > 1/4$ as argued above. $\square$

Let c be a divisor of γ and consider the infinite grid G_c and its intersection with an active box on the pivotal grid G_γ. The boxes of G_c located in the active box form entries of (or *cells*) a *matrix* M_c with *precision* $1/c$. The matrix M_c has γ/c rows and columns with indices starting at the top-left corner of M_c.

Observation Let A be a node that occupies the cell $M_c[i, j]$, for some $1 < i, j < \gamma/c$. One of the main consequences of Lemma 2.1 is that A dominates all nodes belonging to matrix cells $M_c[i + 5, j], M_c[i + 6, j], \ldots, M_c[\gamma/c, j]$ in the same column j.

We use this observation later in the main algorithm. Note also, that if we enhance matrix precision so that $1/c \geq 5g$, then the top node in each column of M_c dominates all other nodes in the same column. This follows from the fact that the minimum distance between any two nodes in the network is $1/g$ and from Lemma 2.1. Let $\xi = \min\left\{k \mid 2^k \geq 5\gamma g\right\}$ and let $\eta = \gamma/2^\xi$. We say that the set of top elements in each column of M_η forms the *top border* B_{top} of the active box. Since all nodes in the active box are dominated by the border nodes, we conclude that it suffices to use only the border nodes while performing communication with the nodes in the top range.

Before we present a pseudocode of our broadcasting algorithm we recall that it runs in

36 periodic stages, where a box $B[i \cdot \gamma, j \cdot \gamma]$ has a chance to become active in stage k iff $(i = k) \bmod 6$ and $(j = \lfloor \frac{k}{6} \rfloor) \bmod 6$. Recall also that each stage is split into four rounds. In what follows we focus on transmissions performed by nodes of some active box during a single stage, and in particular on the time complexity $f(g)$ related to communication with the nodes in the top range of the box. Since the communication mechanism with other parts of the range is analogous the total time complexity of a stage is bounded by $4 \cdot f(g)$. Now since every box has a chance to become active in every 36^{th} stage, the maximum time between a distinguished node within some box is established and the activation of the box is bounded by $36 \cdot 4 \cdot f(g) = O(f(g))$. We show that $f(g) = O(g \log g)$.

Our broadcasting algorithm makes an extensive use of Procedure `Echo`(R, v) [22]. The arguments for this procedure are a region R in the plane and a point $v \notin R$ occupied with a network node (to which we also refer as v). The role of the procedure is to find out whether or not there exists a node in R. This is done as follows. Every node in R transmits its location and v transmits a control message simultaneously. Let $u \neq v$ be any node in the transmitting range of all points in $R \cup \{v\}$. The node u now knows whether there exists an occupied point in R as such a point exists if and only if u did not receive the control message (or if u is positioned in R).

Let λ be the leader in the active box. Our broadcasting algorithm is based on two procedures. The first procedure, named Procedure `TopMost`(M_η), computes the top border B_{top}, namely, the top cell occupied by some network nodes for each column C_j of M_η. The top occupied cells are found via direct applications of Procedure `Echo`(R, λ), where the (region) parameter R corresponds to consecutive cells in M_η and the area of R decreases exponentially from one application to the next (in a binary search fashion). In each column of the matrix, the top occupied cell can be found within a logarithmic (in the number of cells in a column) number of applications of Procedure `Echo`. Since there are $O(g)$ columns as well as $O(g)$ cells in each column of M_η, the total time complexity of a single call to Procedure `TopMost`(M_η) is $O(g \log g)$.

When the top border B_{top} is found, the second procedure, named Procedure `Select`(B_{top}, S_i), selects a leader (if possible) in the shape S_i, for $i = 1, \ldots, 6$, forming a part of the top range of the active box (see Figure 1(b)). For each node $v \in B_{top} \setminus \{\lambda\}$, the selection procedure tries to transmit the broadcast message from v to S_i. All newly informed nodes in S_i send confirmation messages and λ sends a control message simultaneously. If v does not receive the control message from λ, then at least one S_i node received the transmission of v. In that case, a leader in S_i is selected via applications of `Echo`(R, v), where $R \subseteq S_i$ and the area of R decreases exponentially from one application to the next (in a binary search fashion) until

R can no longer occupy more than a single node (due to the granularity of the network). The procedure then halts. To handle the possibility that λ is the only node in the active box with a neighbor in S_i, if all nodes $v \in B_{top} \setminus \{\lambda\}$ fail to communicate with S_i, then λ tries to select a leader in S_i (once again, via applications of Procedure `Echo` in a binary search fashion). To analyze the running time of Procedure `Select`, note that the area of S_i is smaller than 1/4 and that the last application of Procedure `Echo` can be applied to a region of size $\Omega(1/g^2)$ (such a region can not occupy more than one node), thus $O(\log g)$ applications of Procedure `Echo` suffices. As $|B_{top}| = O(g)$, it follows that the running time of the selection procedure is $O(g + \log g) = O(g)$.

A similar process is performed to select a leader (and pass the source message) in the right, bottom and left range. The next theorem follows.

Theorem 2.2. *The algorithm presented above performs radio broadcasting in unknown radio networks of granularity g and diameter D in time $O(Dg \log g)$.*

Proof: Recall that the algorithm works in stages and every box has a chance to become active in every 36^{th} stage. Thus the maximum time between selection of the leader in the box and the box activation is bounded by 36 times the length of each stage. The time complexity of each stage is dominated by executions of the border computation procedure, which is executed four times (once for each part of the range). As a single execution of this procedure requires time $O(g \log g)$, the total waiting time is also $O(g \log g)$.

Let P be a shortest path connecting any node w with the source node s (in the UDG). Since a move along each edge of P takes time $O(g \log g)$, and since P consists of at most D edges, it follows that w receives the source message in time $O(Dg \log g)$. □

Information about expansion of the broadcasting tree and ultimately the lack of it can be sent continuously towards the root of the tree using one extra time slot during each stage. The root figures out that the expansion is terminated if it does not receive any expansion messages during consecutive 36 stages. This information can be later distributed to all other nodes in the network to acknowledge termination of the broadcasting process.

2.2 $O(Dg)$−time broadcasting algorithm

Observe that the main bottleneck in our broadcast algorithm is the computation of the top (right, bottom and left) border of the active box, which requires time $\Omega(g \log g)$ in the worst case. Here we show how to reduce this time to $O(g)$.

Recall that in the matrix M_η the (single) node in the top occupied cell of each column dominates every other node in the column (this is actually how we defined the top border B_{top}). However, if the precision $1/c$ of M_c is not so fine, then in order to dominate the whole column, we have to consider the top five cells in the column starting from the highest cell occupied by network nodes. This is a consequence of Lemma 2.1. We call the collection of the top five occupied cells (that is, the top most occupied cell and the four cells immediately below it) in each column C_j the *border area* of C_j. Note that (depending on the matrix precision) the border area of C_j may contain nodes that do not belong to B_{top} (determined with respect to the matrix precision $1/\eta$), while on the other hand, there may exist B_{top} nodes in C_j that do not belong to the border area. We refer to the union of the border areas in all columns of M_c as the *border field* of the matrix.

Another interesting consequence of Lemma 2.1 is that we sometimes do not need to extract all the top border nodes since some top border nodes can be dominated by the others. A subset of the top border nodes is called *effective* if it dominates all other nodes in the top border and in consequence also all other nodes in the active box. Analogously, a subset of the border field of M_c is called *effective* if its nodes dominate all other nodes in the active box.

At the heart of our improved algorithm is a procedure that computes an effective subset of the top border nodes in time $O(g)$. The algorithm is based on computation of a sequence of effective subsets of border fields in matrices M_c with exponentially increasing precision $1/c$. This is done by Procedure `Effective`, which works as follows. Define $\alpha(i) = \gamma/2^i$. Let W_0 be the set that consists of the single cell of $M_{\alpha(0)}$. For $i = 1, \ldots, \xi$, Procedure `Effective` computes a subset W_i of the border field of $M_{\alpha(i)}$. We will prove soon that W_i is effective for every $0 \leq i \leq \xi$. The computation of W_i is performed based on the set W_{i-1} in the following manner. Let A be a border area in W_{i-1}, i.e., the region A consists of the top 5 occupied cells in some column of the matrix $M_{\alpha(i-1)}$. As each cell of $M_{\alpha(i-1)}$ is split into 2×2 cells of $M_{\alpha(i)}$, the region A is split into a 10×2 submatrix of $M_{\alpha(i)}$. Procedure `Effective` computes the border areas A_1 and A_2 of this submatrix via $2(\lceil \log(10) \rceil + 1) = 10$ applications of Procedure `Echo` and stores them in W_i, and this is done for every region A in W_{i-1}.

Lemma 2.3. *The set W_i is an effective subset of the border field of the matrix $M_{\alpha(i)}$ for every $0 \leq i \leq \xi$.*

Proof: For every $0 \leq i \leq \xi$, we have to prove that W_i is a subset of the border field of $M_{\alpha(i)}$ and that it is effective. We prove these two properties by induction on i. The assertion holds trivially for $i = 0$ as the matrix $M_{\alpha(0)}$ admits a single cell contained in W_0. Every region A in W_{i-1} is replaced in W_i by the border areas A_1 and A_2 of the 10×2 submatrix of $M_{\alpha(i)}$. By the inductive hypothesis, A is a border area in $M_{\alpha(i-1)}$, hence A_1 and A_2 are border areas in $M_{\alpha(i)}$. To see that W_i is effective, note that by the inductive hypothesis, the nodes in W_{i-1} dominate the whole active box. By Lemma 2.1, the nodes in W_i dominate the nodes in W_{i-1}, therefore, since dominance is a transitive relation, the nodes in W_i also dominate the whole active box and W_i is indeed an effective subset of the border field. $\square$

Recall that $2^\xi = O(g)$, and that the last matrix $M_{\alpha(\xi)}$ is exactly M_η. It is left to bound the running time of Procedure `Effective`. For every $1 \le i \le \xi$, the computation of W_i requires (at most) a constant number of rounds for each column in $M_{\alpha(i-1)}$. As $M_{\alpha(i)}$ has 2^i columns, the running time of Procedure `Effective` is

$$\sum_{i=0}^{\xi-1} O\left(2^i\right) \;=\; O(g)\ .$$

Theorem 2.4. *There exists a deterministic algorithm that broadcasts a source message in radio networks of granularity g and diameter D in time $O(Dg)$.*

3 Broadcasting in the spontaneous wake up model

In this section we address the problem of broadcasting in UDG radio networks assuming that stations may transmit even before they received the source message for the first time. We present two broadcasting algorithms: one working in time $O(D+g^2)$ and the other in time $O(D\log g)$. While these algorithms are based on completely different ideas and, depending on parameter values, one or the other of them may be more efficient, we show that as a pair of algorithms working together (implemented as an algorithm interleaving their steps), they enable broadcasting in time $O\left(\min\left\{D+g^2, D\log g\right\}\right)$.

3.1 An $O(D+g^2)$ algorithm

We first define a hierarchy of three increasingly coarser partitions of the plane. These partitions use squares of different sizes, named *tiles* (of size $d/\sqrt{2}$), *blocks* (of size $1/\sqrt{2}$) and *5-blocks* (of size $5/\sqrt{2}$), respectively. All squares are aligned with the coordinate axes and each square includes its left side with both endpoints and its bottom side without the right endpoint and does not include its right and top sides. In each partition, one square has a corner at the point $(0,0)$, and all other squares of the partition are horizontal and vertical shifts of it. Hence every 5-block consists of 25 blocks and every block consists of g^2 tiles. Enumerate all tiles in each block by integers 1,...,g^2, row by row from left to right, and all blocks in each 5-block by integers 1,..., 25, also row by row from left to right. All nodes know these partitions, and since every node knows its own coordinates, it also knows its *hierarchical address* $\langle i, j, k \rangle$, where k is the 5-block where it resides, $1 \leq j \leq 25$ is the block it belongs to in this 5-block, and $1 \leq i \leq g^2$ is the tile it belongs to in this block. Without loss of generality, the source resides in the first tile of the first block in its 5-block, i.e., its hierarchical address is $\langle 1, 1, k \rangle$ for some k. Notice that by the granularity assumption, at most one node resides in each tile, and the diameter of each block is 1, i.e., all nodes in a block are in each other's range. Also, if nodes u and v are each in the j^{th} block of different 5-blocks, then the distance between them is more than 2 and hence they do not create collisions while transmitting simultaneously.

Algorithm `Elect&Transmit`

The algorithm consists of two parts: preprocessing and source message transmission.

Preprocessing: The preprocessing part consists of two phases, each lasting $25g^2$ rounds. The rounds in a phase are enumerated by integer pairs (i, j), where $1 \leq i \leq g^2$, $1 \leq j \leq 25$, ordered lexicographically. In round (i, j) of the first phase, all nodes of address $\langle i, j, k \rangle$ for some k transmit their coordinates. In round (i, j) of the second phase, all nodes of address $\langle i, j, k \rangle$ for some k transmit all the information received in the first phase. Due to the properties of the square partitions, no collisions occur in those phases. Upon completion of the second phase, for any two blocks B_1 and B_2, all nodes in B_1 and B_2 know all pairs of nodes u, v such that $u \in B_1$, $v \in B_2$ and u and v are adjacent in the UDG (i.e., they are in each other's range). Preprocessing terminates by electing one such adjacent pair for any pair of blocks in which adjacent pairs exist (e.g., the lexicographically first pair).

Source message transmission: The source message transmission part is similar to that in Algorithm Elect-and-Broadcast in [13]. It is divided into identical phases repeated indefinitely, each consisting of 600 two-round steps. The steps in each phase are enumerated by integer pairs $(j, \hat{j})$, where $1 \leq j \neq \hat{j} \leq 25$. Consider step $(j, \hat{j})$. If some node v (1) resides in the j^{th} block of a 5-block, (2) was elected during preprocessing, (3) has received the source message, and (4) has not yet transmitted it, then v transmits the source message in the first round of this step. If some node w (1) resides in the $\hat{j}^{th}$ block of a 5-block, (2) was elected during preprocessing, (3) has received the source message, and (4) has not yet transmitted it, then w transmits the source message in the second round of this step. Notice that no collisions occur in this part either. Indeed, at most one node from a given block transmits in each round, and nodes from different 5-blocks with the same block number j are at distance larger than 2.

Theorem 3.1. *Algorithm* `Elect&Transmit` *completes broadcasting in any spontaneous wake up UDG radio network of diameter D and granularity g in time $O(D+g^2)$.*

Proof: The preprocessing part is completed in time $O(g^2)$. It suffices to show that if a node v gets the source message for the first time in round t and w is in the range of v, then w gets the source message in round $t+3600$ at the latest. This will prove that the source message transmission part is completed in time $O(D)$, and all nodes of the network get the source message.

To prove the above claim, consider such nodes v and w. Suppose that v resides in block B_2 and it got the source message for the first time from a node u in block B_1 in round t, which is in phase p. If $B_1 = B_2$, then all nodes in B_2 know the source message by the end of phase p. Otherwise, all nodes of B_2 know the source message by the end of phase $p+1$.

Suppose that w is in block B_3. If $B_2 = B_3$ then w too knows the source message by the end of phase $p+1$. Otherwise, by phase $p+2$, the node in B_2 which was elected to transmit to B_3 will do it, and consequently, all nodes in B_3 know the source message by the end of phase $p+2$. Since each phase lasts 1200 rounds, the claim follows. □

3.2 An $O(D \log g)$ algorithm

For every block B, fix a binary partition of the set of its tiles. First, partition the set of tiles in B into two halves (or almost halves, if g is odd), then partition each of these halves into halves again, and so on, down to individual tiles. For each member of the partition (except individual tiles), we define the first and the second half, arbitrarily. Every node in B knows in which members it resides on each level of the partition. The main subroutine of the algorithm is the following procedure, based on some ideas of Procedure Elect Couple from [13].

Procedure `Conquer`

The input of the procedure consists of two blocks B_1 and B_2 and a node $v \in B_1$. Block B_1 is called *conquered*. If the two blocks do not have any pair of adjacent nodes $b_1 \in B_1$ and $b_2 \in B_2$, then the procedure has no output. Otherwise, the procedure outputs a pair of adjacent nodes $b_1 \in B_1$ and $b_2 \in B_2$. In this case, block B_2 becomes conquered as well.

The procedure works in two phases.

Phase 1.

Let $x = \lceil 2 \log g \rceil$. Phase 1 takes $2x+2$ rounds divided into $x+1$ two-round steps. With each step we associate some set A of nodes, referred to as the *index* of the step. The nodes in A transmit a bit in the first round of the step, and the nodes in $A \cup \{v\}$ transmit a bit in the second round of the step (cf. Procedure `Echo` from [22]).

The index of the first step is the set of all nodes in B_2. Let C be the set of nodes in B_1 that either heard a message in the first round or heard nothing in both rounds. Nodes in C are exactly those nodes in B_1 that are at distance at most 1 from at least one node in B_2. Suppose that C is nonempty. After the first step, every node in B_1 knows whether or not it is in C. The remaining x steps of Phase 1 are devoted to electing a single node in C. This is done using the binary partition of block B_1. The index of the second step is the set of nodes in C that are in the first half of block B_1. If a node in block B_1 heard a message in the first round of step 2 or heard nothing in both rounds of this step, then it knows that there are some nodes in C that are in the first half of block B_1. In this case, we say that the first half of the block is *promoted*. Otherwise, C contains no nodes from the first half of block B_1 ,and consequently there must be such nodes in the second half of block B_1. In this case, we say that the second half of the block is promoted. The index of the third step consists of nodes in C that are in the promoted half, and so on. After $x+1$ steps, a single tile containing a node is promoted, and the single node in it is elected. Call this node b_1.

Phase 2.

The aim of Phase 2 is to elect in B_2 a single node at distance at most 1 from b_1. Phase 2 takes $2x+1$ rounds. In the first round, node b_1 transmits a bit. Let D be the (nonempty) set of nodes in B_2 that heard this message. The remaining $2x$ rounds are divided into x two-round steps, similar to the last x steps of Phase 1, with the exception that set C is replaced by set D, node v by node b_1, and the binary partition of block b_1 by the (fixed) binary partition of block B_2. The rest is identical. Phase 2 results in electing a node $b_2 \in B_2$ at distance at most 1 from b_1.

We now present the main algorithm using Procedure `Conquer`.

Algorithm `Log_Wave`

In the first round, the source transmits the source message and the block containing the source is *conquered*. The rest of the algorithm is divided into identical phases repeated indefinitely, each consisting of 600 subphases, organized similarly as the Source message transmission part of Algorithm `Elect&Transmit`. The subphases in each phase are enumerated by integer pairs $(j, \hat{j})$, where $1 \leq j \neq \hat{j} \leq 25$. Consider subphase $(j, \hat{j})$. This subphase involves pairs of blocks B_1 and B_2, where B_1 is a j^{th} block in some 5-block, that was first conquered in the preceding phase, and B_2 is a $\hat{j}^{th}$ block in some 5-block. The node

$b_1 \in B_1$ needed as input to Procedure `Conquer` is the one that was elected when block B_1 was first conquered. Subphase $(j, \hat{j})$ consists of the parallel execution of Procedure `Conquer` on all such inputs. Notice that for every j^{th} block B_1 in some 5-block there is at most one $\hat{j}^{th}$ block B_2 in some 5-block such that B_2 can be conquered from B_1. As in Algorithm `Elect&Transmit`, there are no collisions between transmissions from different blocks during a subphase, i.e., no collisions between parallel executions of Procedure `Conquer`. As soon as a new block B_2 is first conquered, the newly elected node in this block transmits the source message. This is the last round of the subphase.

Theorem 3.2. *Algorithm* `Log_Wave` *completes broadcasting in any spontaneous wake up UDG radio network of diameter D and granularity g in time $O(D \log g)$.*

Proof: Procedure `Conquer` is executed in time $O(\log g)$. Hence one phase of Algorithm `Log_Wave` takes time $O(\log g)$ as well. After t phases, any node at (hop) distance t from the source gets the source message. Consequently, after time $O(D \log g)$, all nodes get the source message. □

Finally, consider the algorithm that results from interleaving the steps of the two algorithms described in this section: in even rounds it executes the steps of the $O(D + g^2)$ algorithm and in odd rounds - the steps of the $O(D \log g)$ algorithm. Call the resulting algorithm `Fast_Broadcast`. We have the following theorem.

Theorem 3.3. *Algorithm* `Fast_Broadcast` *completes broadcasting in any spontaneous wake up UDG network of diameter D and granularity g in time $O\left(\min\left\{D + g^2, D \log g\right\}\right)$.*

References

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

[2] Bar-Yehuda, R., Goldreich, O., Itai, A.: On the time complexity of broadcast in radio networks: an exponential gap between determinism and randomization. *J. of Computer and System Sciences* **45**, (1992), 104–126.

[3] Bruschi, D., Del Pinto, M.: Lower bounds for the broadcast problem in mobile radio networks. *Distributed Computing* **10**, (1997), 129–135.

[4] Chlamtac, I., Kutten, S.: On broadcasting in radio networks - problem analysis and protocol design. *IEEE Trans. on Communications* **33**, (1985), 1240–1246.

[5] Chlamtac, I., Weinstein, O.: The wave expansion approach to broadcasting in multihop radio networks. *IEEE Trans. on Communications* **39**, (1991), 426–433.

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

[7] Chlebus, B., Gąsieniec, L., Östlin, A., Robson, J.M.: Deterministic radio broadcasting. *Proc. 27th Int. Colloq. on Automata, Languages and Programming* (ICALP'2000), LNCS 1853, 717–728.

[8] Chlebus, B., Kowalski, D.: A better wake-up in radio networks, *Proc. 23rd Symp. on Principles of Distributed Computing* (PODC 2004).

[9] Chrobak, M., Gąsieniec, L., Kowalski, D.: The wake-up problem in multi-hop radio networks. *Proc. 15th ACM-SIAM Symp. on Discrete Algorithms* (SODA 2004), 985 – 993.

[10] Chrobak, M., Gąsieniec, L., Rytter, W.: Fast broadcasting and gossiping in radio networks. *Proc. 41st Symp. on Foundations of Computer Science* (FOCS'2000), 575–581.

[11] Clementi, A.E.F., Monti, A., Silvestri, R.: Selective families, superimposed codes, and broadcasting on unknown radio networks. *Proc. 12th Ann. ACM-SIAM Symp. on Discrete Algorithms* (SODA'2001), 709–718.

[12] Czumaj, A., Rytter, W.: Broadcasting algorithms in radio networks with unknown topology. *Proc. 44th Symp. on Foundations of Computer Science* (FOCS'2003) 492–501.

[13] Dessmark, A., Pelc, A.: Broadcasting in geometric radio networks. *J. of Discrete Algorithms*, to appear.

[14] Elkin, M., Kortsarz, G.: Improved broadcast schedule for radio networks. *Proc. 16th ACM-SIAM Symp. on Discrete Algorithms* (SODA 2005).

[15] Y. Emek, L. Gąsieniec, E. Kantor, D. Peleg, A. Pelc and C. Su. Lower Bounds on Broadcasting Time in UDG Radio Networks with Unknown Topology. Unpublished Manuscript, Nov. 2006.

[16] Diks, K., Kranakis, E., Krizanc, D., Pelc, A.: The impact of knowledge on broadcasting time in linear radio networks. *Theoretical Computer Science* **287**, (2002), 449-471.

[17] Gaber, I., Mansour, Y.: Centralized broadcast in multihop radio networks. *J. of Algorithms* **46**, (2003), 1–20.

[18] Gąsieniec, L., Pelc, A., Peleg, D.: The wakeup problem in synchronous broadcast systems. *SIAM J. on Discrete Mathematics* **14**, (2001), 207-222.

[19] Gąsieniec, L., Peleg, D., Xin, Q.: Faster communication in known topology radio networks. *Proc. 24th ACM Symp. on Principles Of Distributed Computing* (PODC 2005), 129–137.

[20] Jurdzinski, T., Stachowiak, G.: Probabilistic algorithms for the wakeup problem in single-hop radio networks. *Proc. 13th Int. Symp. on Algorithms and Computation* (ISAAC 2002), LNCS 2518, 535 – 549.

[21] Kesselman, A. , Kowalski, D.,: Fast Distributed Algorithm for Convergecast in Ad Hoc Geometric Radio Networks. *Proc. 2nd Int. Conf. on Wireless on Demand Network Systems and Service* (WONS 2005), 119-124.

[22] Kowalski, D., Pelc, A.: Time of deterministic broadcasting in radio networks with local knowledge, *SIAM J. on Computing* **33**, (2004), 870-891.

[23] Kowalski, D., Pelc, A.: Time complexity of radio broadcasting: adaptiveness vs. obliviousness and randomization vs. determinism. *Theoretical Computer Science* **333**, (2005), 355–371.

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

[25] Kowalski, D., Pelc, A.: Optimal deterministic broadcasting in known topology radio networks. *Distributed Computing*, to appear.

[26] Kranakis, E., Krizanc, D., Pelc, A.: Fault-tolerant broadcasting in radio networks, *J. of Algorithms* **39**, (2001), 47–67.

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

[28] Moscibroda, T., Wattenhofer, R.: Maximal independent sets in radio networks. *Proc. 24th ACM Symp. on Principles of Distributed Computing* (PODC 2005), 148–157.

[29] Moscibroda, T., Wattenhofer, R.: Coloring unstructured radio networks. *Proc. 17th ACM Symp. on Parallel Algorithms* (SPAA 2005), 39–48.

[30] Ravishankar, K., Singh, S.: Broadcasting on $[0, L]$. *Discrete Applied Mathematics* **53**, (1994), 299–319.

[31] Sen, A., Huson, M. L.: A New Model for Scheduling Packet Radio Networks, *Proc. 15th Joint Conf. of the IEEE Computer and Communication Societies* (IEEE INFOCOM'96) (1996), 1116 - 1124.