Distrib. Comput. (2009) 21:331–351
DOI 10.1007/s00446-008-0075-z

Broadcasting in UDG radio networks with unknown topology

Yuval Emek · Leszek Gąsieniec · Erez Kantor · Andrzej Pelc · David Peleg · Chang Su

Received: 24 June 2007 / Accepted: 9 October 2008 / Published online: 19 November 2008

Abstract The paper considers broadcasting in radio networks, modeled as unit disk graphs (UDG). Such networks occur in wireless communication between sites (e.g., stations or sensors) situated in a terrain. Network stations are represented by 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 hear 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 network, 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 and prove that broadcasting in this model cannot be accomplished by any deterministic algorithm in time better than $\Omega(D\sqrt{g})$. 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 $O(D \log g)$. While neither of these algorithms alone is optimal for all parameter values, we prove that the algorithm obtained by interleaving their steps, and thus working in time $O\left(\min\left\{D+g^2,\, D\log g\right\}\right)$, turns out to be optimal by establishing a matching lower bound.

Keywords Radio networks · Unit disk graphs · Ad hoc networks · Broadcasting

The research of Leszek Gąsieniec is partially supported by Royal Society International Joint Project 2006/R2. The research of Andrzej Pelc is partially supported by NSERC discovery grant and by the Research Chair in Distributed Computing at the Université du Québec en Outaouais. The research of David Peleg is partially supported by grants from the Minerva Foundation and the Israel Ministry of Science.

Y. Emek (✉)
School of Electrical Engineering, Tel Aviv University,
Tel Aviv 69978, Israel
e-mail: yuvale@eng.tau.ac.il

L. Gąsieniec · C. Su
Department of Computer Science, University of Liverpool,
Liverpool L69 3BX, UK
e-mail: L.A.Gasieniec@liverpool.ac.uk

C. Su
e-mail: suc@csc.liv.ac.uk

E. Kantor · D. Peleg
Department of Computer Science and Applied Mathematics,
The Weizmann Institute of Science, Rehovot 76100, Israel
e-mail: erez.kantor@weizmann.ac.il

D. Peleg
e-mail: david.peleg@weizmann.ac.il

A. Pelc
Département d'informatique, Université du Québec en Outaouais,
Gatineau, QC J8X 3X7, Canada
e-mail: pelc@uqo.ca

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 plane. Two nodes are joined by an edge if their Euclidean 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 Euclidean 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 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.

It is assumed that the network topology is *unknown*, namely, each node knows its own coordinates in the Euclidean plane, but it is unaware of the coordinates of any other node including its neighbors. Such networks are often called *ad hoc* networks. In fact, the lower bounds established in this paper remain valid even if the nodes know some global parameters of the network, such as the size or the diameter.

The paper considers *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 hear a 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 hear the source message. All transmissions proceed in synchronous *rounds* measured by a global clock that shows the round number.

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 [26,27].

It is assumed that the nodes are aware of the network's *density* d, which is the minimum Euclidean distance between any two nodes. However, all our results remain true if this information is replaced by a linear lower bound on d. The physical size of a radio station may provide such a lower bound, as it is reasonable to assume that stations do not overlap. The issue of density of nodes in geometric graphs has been mostly studied in the context of random graphs, where the emphasis is on exploration of threshold probability values controlling certain graph properties, such as *connectivity*, bounded *cover time* of a random walk, and limited *routing stretch*, see, e.g., [2,34]. In our work, however, we focus on worst case instances of geometric graphs, as well as on relations between the parameters of the network and the efficiency of communication algorithms.

We consider only deterministic broadcasting algorithms and do not assume any central authority monitoring the broadcasting process. Thus the decision made by a node on 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 it heard 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.

As in most of the papers on algorithmic aspects of radio broadcasting, we do not impose bounds on the size of messages. In particular we assume that the (exact) position of a node can be encoded in a message. It is worthwhile to mention that the problematic implications of this assumption are addressed (at least to some extent) in the sequel papers [18,19].

Our contributions. The focus of this paper is on the design of fast broadcasting algorithms working in arbitrary UDG radio networks with unknown topology, and on establishing lower bounds on 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 network between any two nodes. (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 other parameter is the *granularity* of the network, denoted by g. This is the inverse of the density parameter d, namely, the inverse of the minimum Euclidean distance between any two nodes of the network. Hence networks of large granularity are those that have some nodes close to each other. The broadcasting times in both models are increasing functions of D and g.

For the conditional wake up model, we present a deterministic algorithm that completes broadcast in time $O(Dg)$ in any UDG radio network of diameter D and granularity g. This is done in Sect. 2.1. On the negative side, in Sect. 2.2 we establish an $\Omega(D\sqrt{g})$ lower bound on the execution time of such algorithms. For the spontaneous wake up model, in Sect. 3.1 we develop two deterministic 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. However, when they are combined together by interleaving transmissions from their consecutive rounds they form an asymptotically optimal broadcasting algorithm with the time complexity $O\big(\min\{D + g^2, D \log g\}\big)$. Indeed, a matching lower bound of $\Omega\big(\min\{D + g^2, D \log g\}\big)$ is established in Sect. 3.2.

Our results give a provable separation between the conditional and the spontaneous wake up models for broadcasting in UDG radio networks: for networks of small diameter (e.g., D polylogarithmic in g) the lower bound for the conditional wake up model is significantly larger than the upper bound for the spontaneous wake up model.

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. [1,5,6,17,20,22,28]), 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 [5]. The authors also formulated the model of radio network subsequently used by many researchers. In [6], an $O(D \log^2 n)$-time broadcasting algorithm was given for all n-node networks of diameter D. In [20], an $O(D + \log^5 n)$-time broadcasting was proposed. This was improved to $D + O(\log^4 n)$ in [17], then to $D + O(\log^3 n)$ in [22], and very recently to $O(D + \log^2 n)$ in [28]. 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 [3]. The authors assumed that nodes know only their own label and labels of their neighbors. Many authors [4,7,8,11–13,26] 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 (ad hoc networks). In [7] the authors gave a broadcasting algorithm working in time $O(n)$ 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 [26] even for the class of networks of constant radius. On the other hand, in [27] a lower bound $\Omega\left(n\frac{\log n}{\log(n/D)}\right)$ was proved for (undirected) n-node networks of diameter D, if spontaneous transmissions are not allowed.

In [7,8,11,13] 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 algorithms for such networks are the $O(n \log^2 D)$-time algorithm from [13] and the $O(n \log n \log\log n)$-time algorithm from [14]. On the other hand, in [12] an $\Omega(n \log D)$ lower bound 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 [3,31]. The authors do not assume that nodes know the topology of the network or that they have distinct labels. In [3] the authors showed a randomized broadcasting algorithm running in expected time $O(D \log n + \log^2 n)$. In [31] it was shown that for any randomized broadcasting algorithm and parameters $D \le n$, 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 $O(D \log(n/D) + \log^2 n)$, and thus matching the above lower bounds, was presented in [27] (cf. also [13]).

The wakeup problem in radio networks was first studied in [21] for single-hop networks (modeled by complete graphs), and then in [9,10] for arbitrary networks. In [23] 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 longer arbitrary. It may be a unit disk graph, or its generalization, where radii of disks representing reachability areas may differ from node to node [15], or reachability areas may be of shapes different than a disk [16,29]. Broadcasting in such geometric radio networks and some of their variations was considered, e.g., in [15,16,29,35,36]. In [36] 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 [35] broadcasting was considered in networks with nodes randomly placed on a line. In [29] 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 [15]. 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 under the spontaneous wake up model which works in time linear in the number of nodes, assuming that nodes are labeled by consecutive integers. The main difference between our model and the one from [15] is that we investigate dependence of broadcasting time on diameter and granularity, while in [15] the authors use diameter and the number of nodes as parameters. It should be noted that the total number of nodes in a network of diameter D and granularity g may be as large as $\Omega(D^2 g^2)$ or as small as $O(D)$, hence the algorithm of [15] is much slower than ours.

Modeling ad hoc radio networks by unit disk graphs and their generalizations has recently attracted growing attention. In [32] this model was used for studying distributed solutions of the maximum independent set problem, in [33] of the coloring problem, and in [24] the convergecast problem was studied in geometric radio networks with varying reachability radii.

2 Conditional wake up

2.1 Broadcasting algorithm

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 [25], and new notions of *a grid of boxes*, *a border* and an *effective border*, to be defined later in this section. We begin in Sect. 2.1.1 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 Sect. 2.1.2.

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.

2.1.1 $O(Dg \log g)$-*time 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 $c \times c$. All boxes are aligned with the coordinate axes and each box includes its left side with both endpoints and its bottom side without the right endpoint and does not include its right and top sides. 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 (see Fig. 1a), plays a central role in our broadcasting algorithm. Note that $\frac{1}{c} = \sqrt{2}$ is the smallest 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, as illustrated in Fig. 1b.

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 transmission range of the box and to select leaders in boxes occupied by the newly informed nodes.

Outline of the broadcasting algorithm. 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 Fig. 2a, and its period is 36, i.e., each box has a chance to become active during every 36th 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 only if $i = k \bmod 6$ and $j = \lfloor k/6 \rfloor \bmod 6$. It is evident that the proposed transmission pattern (based on distant boxes whose ranges are at

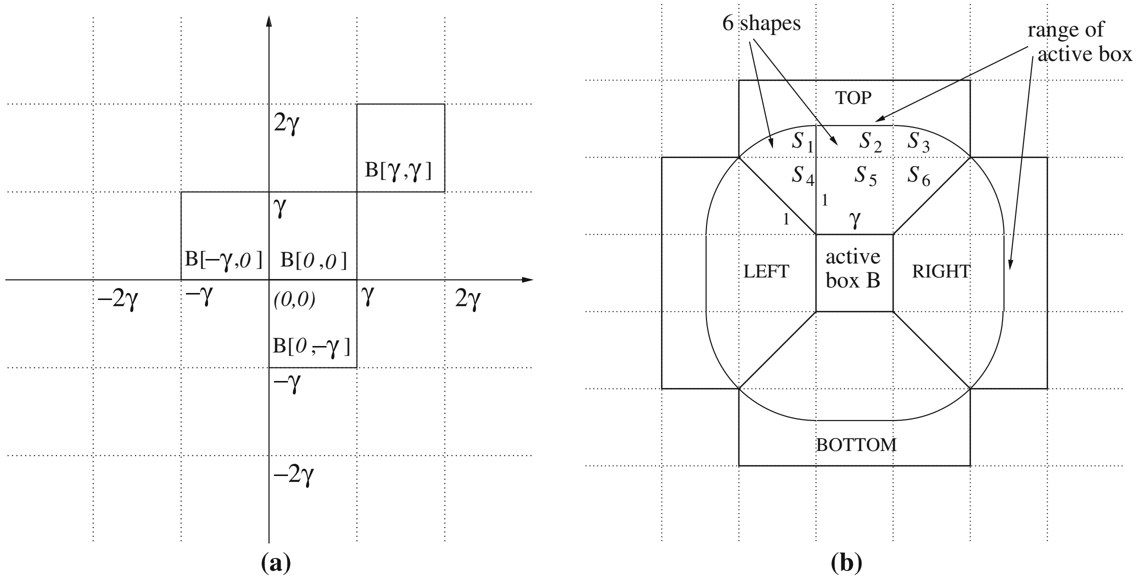

Fig. 1 **a** the pivotal grid, and **b** the transmission range

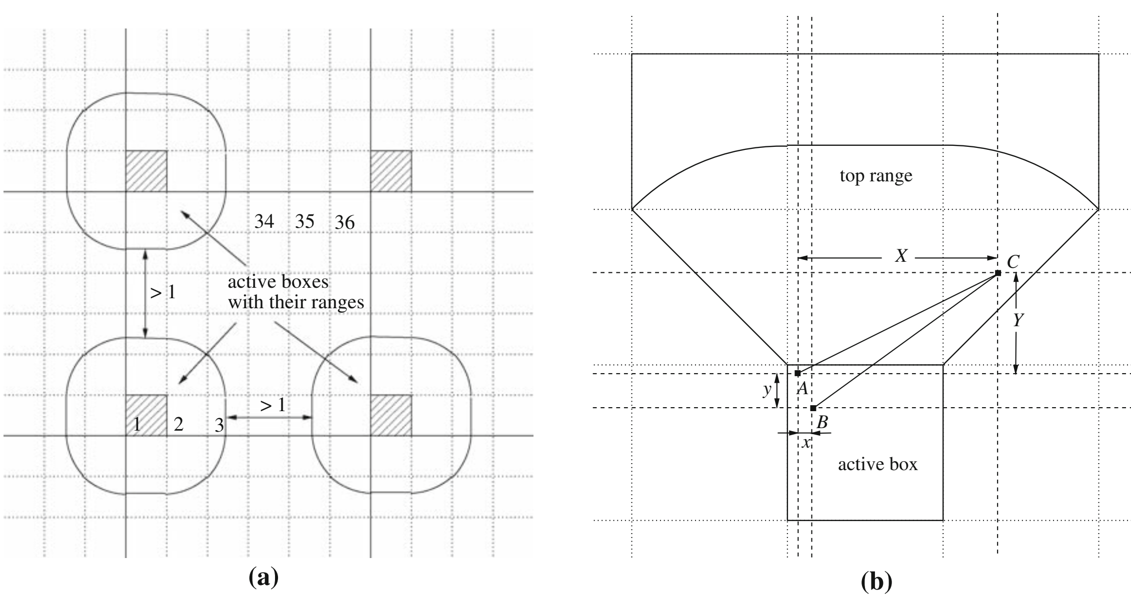

Fig. 2 **a** Active boxes and their ranges, and **b** node *A* dominates node *B* w.r.t. the top range

distance more than 1 apart, see Fig. 2a), prevents the possibility of collisions between transmissions coming from different active boxes.

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 Fig. 1b). Each part of the range overlaps with six boxes of the grid. The overlapping parts form six different *shapes* $S_1, \ldots, S_6$ (see Fig. 1b).

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.

Domination property. 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*. The following lemma can now be established using a straightforward geometric analysis.

Lemma 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 $y = y_A - y_B$ (observe that x is in absolute value while y may be negative). 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 Fig. 2b). If $X = 0$ and $y > 4x \geq 0$, then clearly, under the assumption that C is reachable from B, we conclude that C is also reachable from A. The same holds if $x = 0$ and $y > 4x \geq 0$. Therefore in what follows we assume that both x and X are positive. Suppose, towards contradiction, that $y > 4x$ (as in the premise of the lemma), yet C is reachable from B but not from A, i.e.,

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

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. □

Let $c = \gamma/a$ for some positive integer a 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 (or *cells*) of 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 $1 \leq i, j \leq \gamma/c$. One of the main consequences of Lemma 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 1.

Let $\xi = \min\{k \mid 2^k \geq 5\gamma g\}$ and let $\eta = \gamma/2^\xi$. We say that the set of top nodes 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 top border, we conclude that it suffices to use only the top border nodes while performing communication with the nodes in the top range.

The following phenomenon serves as a fundamental ingredient in our broadcasting algorithm. Consider some region R in the plane, possibly inhabited by some nodes, and a distinguished node $\psi \notin R$ such that the Euclidean distance between any two points in $R \cup \{\psi\}$ is at most 1. Suppose that in some round t all nodes in R transmit simultaneously with ψ (all other relevant nodes listen) and let $\phi \neq \psi$ be some node at Euclidean distance at most 1 from all points in $R \cup \{\psi\}$. We argue that by the outcome of round t, the node ϕ can decide whether R contains any node. Indeed, if $\phi \in R$, then clearly, ϕ knows that R contains at least one node. On the other hand, if $\phi \notin R$, then ϕ was listening in round t. In that case, if ϕ does not hear anything, then ψ's transmission collided with some transmission originated in R, hence R must contain some nodes; if ϕ hears ψ's transmission, then R does not contain any node. We refer to such a round t as a *content test* of the region R.

Procedure echo. Our broadcasting algorithm makes an extensive use of Procedure `Echo` [25]. The procedure operates on a region R in the plane and a distinguished node $\psi \notin R$ such that the Euclidean distance between any two points in $R \cup \{\psi\}$ is at most 1. During execution of procedure `Echo`, the distinguished node ψ assists in the selection of a single node in R, called a *representative*. As the outcome of the selection process, if R contains some nodes, then all participating nodes are aware of the coordinates of the selected representative; if R does not contain any node, then ψ learns about it.

The region R has an abstract representation in the form of a vector $V[1, \ldots, k]$ of k *subregions*, where each subregion contains at most one node. For simplicity, we assume that k is a power of 2 (this can be easily generalized). Unless stated otherwise, we assume that the subregions are boxes of the grid G_c imposed on R for some $c \leq d/\sqrt{2}$ so that a box cannot contain more than one node. The procedure is invoked by ψ which sends a message (heard by all nodes in R assuming that there are no collisions) that encodes the vector V, that is, the exact manner in which R is partitioned to subregions. Therefore every node $v \in R$ knows the index $1 \leq i_v \leq k$ of the subregion in which it is located.

A pseudocode of procedure `Echo` governing the selection process is written from the perspectives of the distinguished node ψ and a regular participant $v \in R$ (see Fig. 3). This procedure is a tuned version of the algorithm originally presented by Kowalski and Pelc in [25]. It is based on content tests of subsets of the subregions performed in a binary search fashion. The result of a typical content test rules out some nodes in R from participating in the selection process. When a regular node is no longer taking part in the selection process, its local variable *participate*, initially set to *true*, converts to *false*.

In each iteration of the loop "repeat", the number of yet to be considered subregions of R (the subregions whose potential inhabiting nodes are still participating) is halved. In case the upper half of the yet to be considered subregions (those

```
Node v on invocation of Echo(V[1,...,k]) by distinguished node ψ ≠ v:
 1. bot ← 0; top ← k; participate ← true;
 2. repeat for log k rounds:
    (a) mid ← (bot + top)/2;
    (b) case [participate = false]:
          i. listen;
    (c) case [participate = true]:
          i. if mid + 1 ≤ i_v ≤ top, then
             transmit own coordinates;
             bot ← mid;
         ii. if bot + 1 ≤ i_v ≤ mid, then
             listen;
             A. if m_ψ is heard, then
                  top ← mid;
             B. else participate ← false;

Distinguished node ψ on invocation of Echo(V[1,...,k]):
 1. repeat for log k − 1 rounds:
    (a) transmit m_ψ;
 2. listen (for one round);
```

Fig. 3 Pseudocode of Procedure `Echo`

with indices in $\{mid + 1, \ldots, top\}$) is inhabited, the nodes in it know about it (recall that each node knows the index of its subregion) and they survive (to participate in the next iterations). The nodes in the lower half of the yet to be considered subregions (those with indices in $\{bot + 1, \ldots, mid\}$) either hear the message m_ψ and learn that the upper half is empty, in which case they survive, or learn that the upper half is inhabited and withdraw from further selection process.

On the conclusion of procedure `Echo`, all participating nodes learn the coordinates of the node v inhabiting the subregion with the largest index in V. These coordinates are disseminated to all nodes (including ψ) in the last iteration of the loop "repeat". The node v is thus selected to be the representative of R. Note that the time complexity of procedure `Echo` is logarithmic in the number of subregions of R.

The broadcasting algorithm. 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 only if $i = k \bmod 6$ and $j = \lfloor \frac{k}{6} \rfloor \bmod 6$. Recall also that each stage is split into four phases. 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 $f(g)$ of 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, as every box has a chance to become active every 36th stage, the maximum time that passes since a leader in some box is established until the activation of the box is bounded by $36 \times 4 \cdot f(g) = O(f(g))$. We show that $f(g) = O(g \log g)$.

It is important to point out that each (inhabited) box B has one leader although several boxes may attempt to select a leader in B. We assume that once a leader in B is selected, say via communication with some box B' such that B intersects with the top range of B', the nodes in B will not cooperate (i.e., act as if the box is not inhabited) with any subsequent attempt to select a leader in B.

Let λ be the leader in the active box. Our broadcasting algorithm is divided to two parts, namely, border computation and leader selection.

Border computation. The border computation part (see Fig. 4), establishes the content of the top border B_{top}, that is, the top cell occupied by some network node for each column C_j of M_η (a column does not contribute any cell to the top border if it is not inhabited). The top occupied cell is found via a direct application of Procedure `Echo`, where the region R corresponds to the column C_j and the subregions of R correspond to the cells of C_j. Since there are $O(g)$ columns as well as $O(g)$ cells in each column of M_η, the total time complexity of the border computation is $O(g \log g)$.

Leader selection. When the top border B_{top} is found, the broadcasting algorithm turns to select a leader (if possible) in the shape S_i, for $i = 1, \ldots, 6$ (see Fig. 1b). For each node $v \in B_{top} \setminus \{\lambda\}$, the algorithm 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 hear the control message from λ, then at least one node in S_i received the transmission of v. In that case, a leader in S_i is selected via an application of procedure `Echo` with distinguished node v, where the region R is the intersection of S_i and the transmission range of v. Finally, to handle the possibility that $\lambda \in B_{top}$, if a leader in S_i was not selected by some node in $B_{top} - \{\lambda\}$, then λ invokes Procedure `Echo` in attempt to select a leader in S_i by itself. (Refer to Fig. 4 for pseudocode.)

To analyze the running time of the leader selection part, note that there are $O(g)$ border nodes to consider in the active box. Each border node v spends $O(1)$ rounds to decide whether it can establish communication (and select a leader) with some nodes in the shape S_i. If v can communicate with S_i, then it invokes Procedure `Echo` on a region of area smaller than $1/2$ (an upper bound on the area of a shape) and a

Leader λ when computing the top border in the active box:	Node $v \neq \lambda$ when computing the top border in the active box:
1. **foreach** column C_j of the matrix M_η, **do**: (a) invoke Procedure `Echo` to select the top most node in C_j (a selected node becomes part of the top border);	1. participate in invocations of Procedure `Echo` as instructed by λ;

Leader λ when selecting a new leader in shape S_i:	Node $v \neq \lambda$ when selecting a new leader in shape S_i:
1. **foreach** node v in the top border, **do** (a) instruct v to attempt communication with S_i; (b) wait one round and then transmit m_λ; (c) **if** v announces that communication with S_i is established, **then** wait until v announces that a leader in S_i is selected and start over with the next shape; 2. invoke Procedure `Echo` on (reachable part of) S_i to select a leader;	1. **upon** hearing an instruction to attempt communication with S_i, **do** (a) instruct all (reachable) nodes in S_i to transmit their coordinates in the next round; (b) listen; i. **if** m_λ is heard, **then** announce: "cannot establish communication with S_i"; ii. **else** A. announce "communication with S_i is established"; B. invoke Procedure `Echo` on (reachable part of) S_i to select a leader; C. announce "a leader in S_i is selected";

Fig. 4 Pseudocode for broadcasting from within an active box

leader in S_i is selected (there is no need to try the remaining border nodes). As such a region is partitioned into $O(g^2)$ subregions, it follows that Procedure Echo takes $O(\log g)$ rounds. Therefore the time complexity of selecting a leader in S_i is $O(g) + O(\log g) = O(g)$. A similar process is performed to select a leader (and pass the source message) in the remaining five shapes of the top range and in the right, bottom, and left ranges. The theorem follows.

Theorem 1 *There exists a deterministic algorithm that completes broadcast in any unknown UDG radio network of diameter D and granularity g in time $O(Dg \log g)$ under the conditional wake up model.*

Proof Recall that the algorithm works in stages and every box has a chance to become active in every 36th 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.

2.1.2 $O(Dg)$-time broadcasting algorithm

Observe that the main bottleneck in our $O(Dg \log g)$-time broadcast algorithm is the computation of the top 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 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. If the top occupied cell in a column is the ith cell from the bottom for some $i < 5$, then the border area consists of i cells only. 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 1 is that we sometimes do not need to extract all the top border nodes since some top border nodes can be dominated by 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 $0 \le i \le \xi$.

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 (or $2i \times 2$ submatrix if A consists of $i < 5$ cells) of $M_{\alpha(i)}$ (see Figure 5). Procedure `Effective` computes the border areas A_1 and A_2 of this submatrix via $2(\lceil\log(10)\rceil + 1) = 10$ applications of the content test (in a binary search fashion). Upon completion of these 10 content tests, each node in A knows whether it is in A_i for some $i = 1, 2$.

The (sub)regions A_1 and A_2 are considered as border areas in W_i. The above process is repeated for every column in $M_{\alpha(i-1)}$ (recall that A is a region in some column) in an iterative manner. Since each node knows its row and column in $M_{\alpha(i-1)}$, it also knows when (if at all) to participate in the content tests. The next lemma can now be established by induction on i.

Lemma 2 *The set W_i is an effective subset of the border field of the matrix $M_{\alpha(i)}$ for every $0 \le i \le \xi$.*

Proof For every $0 \le i \le \xi$, 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 by definition 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 corresponding 10×2 submatrix (or $2i \times 2$ submatrix if A consists of $i < 5$ cells) 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 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. □

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

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

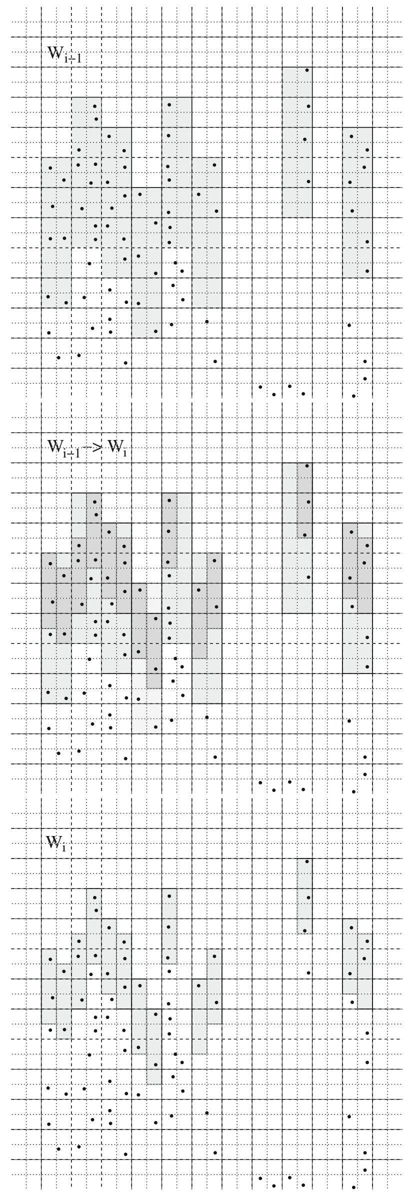

Fig. 5 Transition from the border field W_{i-1} to W_i

Theorem 2 *There exists a deterministic algorithm that completes broadcasting in any unknown UDG radio network of diameter D and granularity g in time $O(Dg)$ under the conditional wake up model.*

Table 1 A table of notations for Sect. 2.2

Notation	Definition
τ	$\lfloor\sqrt{\lfloor g\rfloor+1}\rfloor \approx \sqrt{g}$
Γ	A set of τ^2 auxiliary cells
$B_0, \ldots, B_{\tau-1}$	Blocks of auxiliary cells
$z_0, \ldots, z_{\tau-1}$	Target cells
T^t	Cells scheduled to transmit in round t
T_j^t	$T^t \cap B_j$
U^t	Cells uncommitted at the beginning of round t
U_j^t	$U^t \cap B_j$
Q_j^t	$T_j^t \cap U_j^t$
X^t	$\{j \mid Q_j^t \neq \emptyset\}$
$\mathcal{O}^t$	Cells occupied at the beginning of round t
$\mathcal{E}^t$	Cells empty at the beginning of round t
$\mathcal{O}^\infty$	Cells occupied at the end of the execution
$\mathcal{E}^\infty$	Cells empty at the end of the execution
h^t	The delivery status in round t
$\mathcal{C}_j$	The constraints collection of block B_j
$\mathcal{C}_j^t$	The constraints collection $\mathcal{C}_j$ at the beginning of round t
j^*	An index $j \in X^t$ that maximizes $\lvert Q_j^t\rvert$

2.2 Lower bound

In this section we establish an $\Omega(D\sqrt{g})$ lower bound on the running time of any deterministic broadcasting algorithm under the conditional wake up model. We start with presenting a class $\mathcal{N}$ of UDG radio networks of radius 2 (i.e., the graph distance from the source to any node is at most 2), and prove that for every deterministic broadcasting algorithm $\mathcal{A}$, there exists a network $N \in \mathcal{N}$ such that the running time of $\mathcal{A}$ on N is $\Omega(\sqrt{g})$.

2.2.1 The networks

In the forthcoming constructions we make an extensive use of the notion of *cells*. A cell is a point in the Euclidean plane that may contain a node, in which case we say that the cell is *occupied*, or it may be *empty*. We assume that all the nodes in the networks we construct are placed in cells.

Let $\tau = \lfloor\sqrt{\lfloor g\rfloor+1}\rfloor$. A UDG radio network $N \in \mathcal{N}$ consists of a *source cell* s, a set Γ of τ^2 *auxiliary cells* and a set of τ *target cells*. The source cell is always occupied with a *source node*. The auxiliary cells and the target cells may be occupied with *auxiliary nodes* and *target nodes*, respectively. The auxiliary cells are arranged in pairwise disjoint subsets $B_0, \ldots, B_{\tau-1}$, referred to as *blocks*, each of size τ. The auxiliary cells and the source cell are all within the transmission range of each other, hence they form a clique in N. The target cells are denoted by $z_0, \ldots, z_{\tau-1}$, where z_j is in the transmission range of an auxiliary cell c if and only if $c \in B_j$. The source cell is not in the transmission range of any target cell, hence any communication from the source node to a target node has to be delivered via some auxiliary node. For ease of reference, the reader may use Table 1 which summarizes the notations used throughout this section.

The above structure is guaranteed by positioning the source cell s in $(0, 1/2)$, the auxiliary cells of B_j in $\left\{\left(\frac{1}{2}, (j\tau+i)d\right) \mid 0 \le i < \tau\right\}$ and the target cell z_j in $\left(\frac{1}{2} + \sqrt{1-\left(\frac{\tau-1}{2}d\right)^2}, \left(j\tau+\frac{\tau-1}{2}\right)d\right)$. The embedding of a network of the class $\mathcal{N}$ in the Euclidean plane is illustrated in Fig. 6. The network is designed so that exactly one target cell is occupied by a target node, while the rest of the target cells are empty. Moreover, if z_j is the occupied target cell, then at least one cell $c \in B_j$ is occupied. Therefore any network in $\mathcal{N}$ is connected with radius 2.

2.2.2 Overview

Our lower bound can be viewed as a game played between a deterministic algorithm, whose goal is to deliver the source message to the (single) target, and an adversary which attempts to slow down the algorithm, forcing it to executions of length at least τ. We treat the algorithm as if it enjoys a centralized authority which monitors its behavior (thus increasing its strength). Initially, the algorithm does not have any information on the network, except that it is an instance of $\mathcal{N}$. As the execution progresses, the algorithm gains additional information on which cells are empty and which cells are occupied.

From the perspective of the adversary, initially all auxiliary cells and all target cells are *uncommitted* and as the execution progresses, the adversary *commits* some of the cells to be either empty or occupied (recall that the source cell is

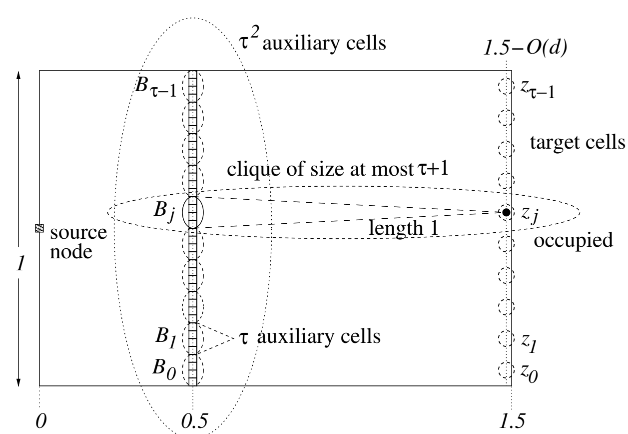

Fig. 6 A network in $\mathcal{N}$. The y-coordinate of the target cells is set to ensure that the occupied target cell z_j is not in the transmission range of B_i for every $j \neq i$

committed to be occupied from the beginning). We assume that the algorithm knows for each cell whether it is uncommitted, empty, or occupied. Eventually (in round τ), all cells become committed and the network $N \in \mathcal{N}$ is revealed. The adversarial policy guarantees that the source message was not heard at the target (there is exactly one occupied target cell) prior to round τ.

We ignore the first round in which the source transmits alone and consider the state in which all auxiliary nodes (but not the target) have already heard the message. In each round, the algorithm schedules a subset T of the cells to transmit. Some of the cells in T may be committed to contain a node (occupied cells) and they certainly transmit. Other cells in T may be committed to be empty and they certainly do not transmit. (In fact, there is no point on behalf of the algorithm to schedule some cell c to transmit if the algorithm already knows that c does not contain a node.) The rest of the nodes in T are still uncommitted and the adversary may have to commit some of them (to be either empty or occupied) in order to support its policy. The adversary makes sure that a target cell z_j does not hear any message unless it is already committed to be empty. Therefore under the conditional wake up model, we may assume that $T \subseteq \Gamma \cup \{s\}$.

At the end of the round, the adversary reports to the algorithm on the *delivery status* of the round which may take one of the following three values:

(1) silence, if all cells in T are (or eventually will be) empty;
(2) hearing, if exactly one cell in T is (or eventually will be) occupied; or
(3) collision, if at least two cells in T are (or eventually will be) occupied.

This report may be employed by the algorithm to design its next moves. The adversary is designed to guarantee that the delivery status reports it makes throughout the execution are consistent with the network N revealed in round τ. Note that under the first or last delivery status reports, neither the auxiliary nodes nor the source heard anything in that particular round.

Recall that the auxiliary cells are arranged in τ blocks, each containing τ cells. A block is said to be *active* as long as it contains at least one uncommitted cell. The adversarial policy maintains the following invariant for every $1 \le t < \tau$: at the end of round t, at least $\tau - t$ blocks are still active, each one of them contains at least $\tau - t$ uncommitted cells. Moreover, the target cells corresponding to active blocks are always uncommitted; whenever a block becomes inactive, the corresponding target cell is committed to be empty. It follows that in round τ we are left with at least one active block B_k, $0 \le k < \tau$, whose corresponding target cell is uncommitted. The network N is then determined by committing the yet uncommitted auxiliary cells in B_k and the target cell z_k to be occupied.

2.2.3 The adversarial policy

We will need the following notation (refer to Table 1 for a brief summary). Let T^t be the set of cells that were scheduled to transmit on round t of the execution (recall that $T^t \subseteq \Gamma \cup \{s\}$). Let $T_j^t = T^t \cap B_j$ for every $0 \le j < \tau$. Let U^t be the subset of cells that are uncommitted at the beginning of round t and let $U_j^t = U^t \cap B_j$ (note that U^t may contain auxiliary cells and target cells, while U_j^t contains auxiliary cells only). Let $Q_j^t = T_j^t \cap U_j^t$ be the set of uncommitted cells in the block B_j that were scheduled to transmit in round t. Let $X^t = \{j \mid Q_j^t \neq \emptyset\}$. Recall that the block B_j is *active* at the beginning of round t if $U_j^t \neq \emptyset$. Otherwise, the block is said to be *inactive*. Clearly, the set X^t contains indices of active blocks only.

Let $\mathcal{O}^t$ (respectively, $\mathcal{E}^t$) be the set of cells that are occupied (resp., empty) at the beginning of round t. Let $\mathcal{O}^\infty$ (respectively, $\mathcal{E}^\infty$) be the set of cells that are occupied (resp., empty) at the end of the execution. (In fact, since the execution lasts for τ rounds, we have $\mathcal{O}^\infty = \mathcal{O}^{\tau+1}$ and $\mathcal{E}^\infty = \mathcal{E}^{\tau+1}$.) Define h^t to be a variable indicating the delivery status decided by the adversary in round t. The adversary will ensure that

$$h^t = \begin{cases} \text{Silence}, & \text{if } T^t \subseteq \mathcal{E}^\infty, \\ \text{Hearing}, & \text{if } |T^t \cap \mathcal{O}^\infty| = 1, \\ \text{Collision}, & \text{otherwise.} \end{cases}$$

Throughout the execution, the adversary maintains a collection $\mathcal{C}_j \subseteq 2^{B_j}$ of *constraints* for each block B_j. At any given time, each constraint consists of uncommitted cells. The existence of the constraint $\chi \subseteq B_j$ in $\mathcal{C}_j$ indicates that the adversary must eventually commit the cells in χ so that $|\chi \cap \mathcal{O}^\infty| \neq 1$. If this is the case, then we say that χ is *satisfied*. Intuitively, if T_j^t is a constraint in $\mathcal{C}_j$, then the target cell z_j could not hear the message in round t, as the round yielded either silence or collision (in z_j). Let $\mathcal{C}_j^t$ denote the constraints collection $\mathcal{C}_j$ at the beginning of round t. We say that $\mathcal{C}_j^t$ is in *canonical form* if $|\chi| > 1$ for every $\chi \in \mathcal{C}_j^t$, i.e., $\mathcal{C}_j^t$ does not contain singleton constraints. Note that a constraint consisting of a single (uncommitted) cell c implies that c must eventually be empty, i.e., $c \in \mathcal{E}^\infty$. On the other hand, if $\mathcal{C}_j^t$ is in canonical form, then the adversary can satisfy all constraints in $\mathcal{C}_j^t$ by either (a) committing all cells in U_j^t to be empty (this is trivial); or (b) committing all cells in U_j^t to be occupied (this is because there are no singleton constraints).

The adversary maintains the constraints collections in canonical form as follows. Given a collection of constraints $\mathcal{C}_j$ in canonical form, and a new constraint χ that should be added to it, if χ consists of a single cell c, then the resulting constraints collection $\mathcal{C}_j \cup \{\chi\}$ is not in canonical form. In that case we can *canonize* it by committing the cell c to be empty and erasing it from all the constraints in which it appears. Consequently, some other constraints may now become singleton and this process is repeated with each of them. The above process is referred to as Procedure `Canonize`. This procedure is invoked at the end of each round so that the constraints collections are always in canonical form (at least at the beginning of each round). Note that Procedure `Canonize` does not contradict any existing constraint.

Intuitively, we would like to maintain as many active blocks as possible, each with as many uncommitted cells as possible, as this will allow the adversary to force the algorithm to long executions. In every round the adversary reveals at most one active block, i.e., at most one active block becomes inactive. Moreover, at the end of round t, each active block contains at least $\tau - t$ uncommitted auxiliary cells, so that in an amortized manner, in every round at most one uncommitted cell in each active block is committed.

In attempt to prevent the algorithm from gaining information, the adversary ensures that $h^t = silence$ or $h^t = collision$ whenever possible. Clearly, the decisions of the adversary must depend on the current action of the algorithm, that is, on the set of cells T^t that are scheduled to transmit. An adversarial decision is not required when $X^t = \emptyset$, as all cells scheduled to transmit are already committed (to be either occupied or empty), so in the following discussion we assume that $X^t \neq \emptyset$.

If a cell c was committed to be occupied, then the algorithm may learn about it in a logarithmic number of rounds (actually, under our assumptions, the algorithm learns about it right away) and schedule c to transmit alone so that the message will be heard in the corresponding target cell. Therefore, whenever the adversary is forced to commit some cells in the block B_j to be occupied, it commits all other (uncommitted) cells in B_j to be occupied as well, and at the same time it commits the corresponding target z_j to be empty. Hence the block B_j becomes inactive. Consequently the adversary maintains the following property.

Property 1 If the block B_j is active at the beginning of round t, then $B_j \cap \mathcal{O}^t = \emptyset$.

We now turn to describe the adversarial policy in round t for every $1 \leq t < \tau$ such that $X^t \neq \emptyset$. The choice of the network $N \in \mathcal{N}$, performed in round τ, will be discussed afterwards. In what follows, we define j^* to be an index in X^t that maximizes $|Q^t_{j^*}|$ (so that $|Q^t_{j^*}| \geq |Q^t_j|$ for every $0 \leq j < \tau$). We first consider the case where T^t contains two (or more) cells that are already committed to be occupied, namely, $|T^t \cap \mathcal{O}^t| \geq 2$. Collision is ensured in $\Gamma \cup \{s\}$ (that is, $h^t = collision$) and the adversary need not commit on any (new) occupied cells (at least not immediately). In that case, the adversary adds the constraint Q^t_j to $\mathcal{C}_j$ for every $j \in X^t$. By Property 1, the cells in $T^t_j \setminus Q^t_j$ are all empty at the beginning of round t, thus the message was not heard in z_j during that round (recall that such a constraint implies that there was either silence or collision in the target cell).

Next, if T^t contains exactly one cell that is already committed to be occupied, then the adversary guarantees that $h^t = collision$ by occupying the (still uncommitted) cells of one block B_{j^*}, thus making this block inactive. The target cell z_{j^*} is committed to be empty. For all other indices $j \in X^t$, the adversary ensures that no message was heard in z_j by adding the constraint Q^t_j to $\mathcal{C}_j$. Note that in principle, occupying the cells in $Q^t_{j^*}$ is sufficient to ensure collision, but if the cells in $U^t_{j^*} \setminus Q^t_{j^*}$ remain uncommitted (and hence the block B_{j^*} remains active), then Property 1 will be violated in round $t + 1$.

The last case is where T^t does not contain any cell that is committed to be occupied. We distinguish between two subcases. First, if Q^t_j is a singleton for every $j \in X^t$, then the adversary guarantees that $h^t = silence$ by adding the constraint Q^t_j to $\mathcal{C}_j$ for every $j \in X^t$, thus committing all cells in $T^t \cap U^t$ to be empty (at most one cell in each block). Otherwise, $|Q^t_{j^*}| \geq 2$ (as $j^* = \operatorname{argmax}\{|Q^t_j| : j \in X^t\}$) and the adversary guarantees that $h^t = collision$ by occupying the (still uncommitted) cells of B_{j^*}, which turns this block inactive. The target cell z_{j^*} is committed to be empty. For all other indices $j \in X^t$, the adversary ensures that no message was heard in z_j by adding the constraint Q^t_j to $\mathcal{C}_j$. The complete adversarial policy is described formally in Fig. 7.

We would like to show that the adversary forces the algorithm to long executions. In particular, we prove that there exists a network $N \in \mathcal{N}$ such that (1) all cells in $\mathcal{O}^\tau$ are occupied in N; (2) all cells in $\mathcal{E}^\tau$ are empty in N; and (3) when the algorithm is invoked on N, it agrees with the delivery status h^t for every round $1 \leq t < \tau$ (see Lemma 5). This will be used to prove that the algorithm requires at least τ rounds in order to complete a successful broadcasting on N.

Before we can describe the choice of the network N, we have to prove the following two lemmas.

Lemma 3 *Consider some $0 \leq j < \tau$ and $1 \leq t \leq \tau$ such that block B_j is active at the beginning of round t. Then $|U^t_j| \geq \tau - t + 1$.*

Proof Consider round t' for some $1 \leq t' \leq t$. Clearly, since B_j is active at the beginning of round t, B_j is also active at the beginning of round t'. Let $q^{t'}_j = |U^{t'}_j|$ and let $r^{t'}_j = |\mathcal{C}^{t'}_j|$. We would like to show that

$$q^{t'}_j \ \geq \ \tau + 1 + r^{t'}_j - t' \tag{2}$$

as this validates the assertion by taking $t' = t$.

On round $1 \leq t < \tau$ of the execution:

1. If $X^t = \emptyset$, then report h^t in accordance with $|T^t \cap \mathcal{O}^t|$ and continue to round $t+1$.
2. Let j^* be an index in X^t such that $|Q^t_{j^*}| \geq |Q^t_j|$ for any $0 \leq j < \tau$.
3. Case $[|T^t \cap \mathcal{O}^t| \geq 2]$:
   (a) Add the constraint Q^t_j to $\mathcal{C}_j$ for every $j \in X^t$.
   (b) Report $h^t = \mathit{collision}$.
4. Case $[|T^t \cap \mathcal{O}^t| \leq 1]$:
   (a) If $|T^t \cap \mathcal{O}^t| = 0$ and $|Q^t_{j^*}| = 1$, then do:
      i. Add the constraint Q^t_j to $\mathcal{C}_j$ for every $j \in X^t$.
      ii. Report $h^t = \mathit{silence}$.
   (b) Else, do:
      i. Commit the (still uncommitted) cells of B_{j^*} to be occupied.
      ii. Commit z_{j^*} to be empty.
      iii. Add the constraint Q^t_j to $\mathcal{C}_j$ for every $j \in X^t \setminus \{j^*\}$.
      iv. Report $h^t = \mathit{collision}$.
5. Invoke Procedure `Canonize` on $\mathcal{C}_j$ for every $0 \leq j < \tau$.

Fig. 7 The policy of the adversary in the conditional wake up model

In order to establish inequality (2), we first argue that if $j \in X^{t'-1}$ and the adversary adds the constraint $Q^{t'-1}_j$ to $\mathcal{C}_j$ in round $t'-1$, then $q^{t'}_j - r^{t'}_j \geq q^{t'-1}_j - r^{t'-1}_j - 1$. Indeed, by adding the constraint $Q^{t'-1}_j$ to $\mathcal{C}_j$, the cardinality of $\mathcal{C}_j$ increases by 1. At the end of round $t'-1$, Procedure `Canonize` is invoked which may cause some constraints to vanish and some uncommitted cells to become committed, but each uncommitted cell that becomes committed accounts for at least one vanishing constraint. The argument follows.

Inequality (2) can now be proved by induction on t'. The inequality trivially holds at the beginning of round 1 as $q^1_j = \tau$ and $r^1_j = 0$. Assume that the inequality holds at the beginning of round $t'-1$. If $j \notin X^{t'-1}$, then $q^{t'}_j = q^{t'-1}_j$ and $r^{t'}_j = r^{t'-1}_j$, hence, by the inductive assumption, inequality (2) holds at the beginning of round t'. In what follows we assume that $j \in X^{t'-1}$.

The adversary could not have occupied the cells in B_j in round $t'-1$ (see line 4(b)i of Fig. 7) as this implies that B_j is inactive at the beginning of round t'. Therefore the adversary added the constraint $Q^{t'-1}_j$ to $\mathcal{C}_j$ in round $t'-1$. By the above argument, we have $q^{t'}_j \geq q^{t'-1}_j - r^{t'-1}_j - 1 + r^{t'}_j$. By plugging the inductive assumption into the last inequality, we get $q^{t'}_j \geq \tau + r^{t'}_j - t' + 1$, hence inequality (2) holds at the beginning of round t'. □

Lemma 3 is employed in order to establish the following lemma.

Lemma 4 *At the beginning of round τ there exists at least one active block.*

Proof It is sufficient to prove that at most one (active) block becomes inactive in every round $1 \leq t < \tau$. It is obvious from the adversarial policy that at most one block B_{j^*} is explicitly inactivated (see line 4(b)i of Fig. 7). Thus we have to show that every block B_j, $j \neq j^*$, which was active at the beginning of round t remains active at the beginning of round $t+1$ and this is guaranteed by Lemma 3 since $t+1 \leq \tau$. □

Recall that if the target cell z_j is committed in round $1 \leq t < \tau$, then the block B_j is inactive at the beginning of round $t+1$. Therefore Lemma 4 guarantees that at the beginning of round τ there exists some $0 \leq k < \tau$ such that the block B_k is active and the target cell z_k is uncommitted. The network $N \in \mathcal{N}$ is constructed as follows. The source cell s is occupied. If the adversary committed some cell c to be occupied (respectively, empty) in some round $t < \tau$, then c is occupied (resp., empty) in N. The target cell z_k is occupied and so are the cells in U^τ_k. All other cells in U^τ are empty. Now that the sets $\mathcal{O}^\infty$ and $\mathcal{E}^\infty$ are determined (to contain the occupied and empty cells in N, respectively), we can establish the consistency of the adversary.

Lemma 5 *Consider the execution of the algorithm on N. For every round $1 \leq t < \tau$, we have*

(i) *if $h^t = \mathit{silence}$, then none of the nodes in N transmit*;
(ii) *if $h^t = \mathit{hearing}$, then exactly one node in N transmits*; *and*
(iii) *if $h^t = \mathit{collision}$, then at least two of the nodes in N transmit.*

Proof The assertion clearly holds if $X^t = \emptyset$ as the delivery status in such rounds t is determined by cells which are already committed (known to the algorithm). Therefore we can assume that $X^t \neq \emptyset$. The proof is by case analysis. The first case is when $|T^t \cap \mathcal{O}^t| \geq 2$. This case is trivial as $\mathcal{O}^t \subseteq \mathcal{O}^\infty$ and since the adversary reports *collision*. Another case is when $|T^t \cap \mathcal{O}^t| = 1$. In this case the adversary occupies at least one cell in $T^t \cap U^t$, thus $|T^t \cap \mathcal{O}^{t+1}| \geq 2$. The assertion holds as $\mathcal{O}^{t+1} \subseteq \mathcal{O}^\infty$ and since the adversary reports *collision*.

The last case is when $|T^t \cap \mathcal{O}^t| = 0$. Now we have to consider two subcases. If $|Q^t_{j*}| = 1$, then Q^t_j is a singleton for every $j \in X^t$. Therefore, by adding the constraint Q^t_j to $\mathcal{C}_j$ for every $j \in X^t$, the adversary ensures that $T^t \subseteq \mathcal{E}^{t+1} \subseteq \mathcal{E}^\infty$ (recall that Procedure `Canonize` commits a cell that appears in a singleton constraint to be empty) and indeed the adversary reports *silence*. Else ($|Q^t_{j*}| > 1$), the adversary occupies the cells in Q^t_{j*} (there are at least two of them), thus $|T^t \cap \mathcal{O}^{t+1}| \geq 2$. Once again, the assertion holds as $\mathcal{O}^{t+1} \subseteq \mathcal{O}^\infty$ and since the adversary reports *collision*. □

Since the adversary maintains either collision or silence on every round t such that $X^t \neq \emptyset$, it follows that the source node and the auxiliary nodes do not hear any message on such rounds. For the target node, we prove the following lemma.

Lemma 6 *Let j be some index in $[0, \tau)$ and consider some $1 \leq t \leq \tau$ such that B_j is active at the beginning of round t. If all the constraints in $\mathcal{C}^t_j$ are satisfied, namely, if $|\chi \cap \mathcal{O}^\infty| \neq 1$ for every $\chi \in \mathcal{C}^t_j$, then no message was heard at the target cell z_j in any round $t' < t$.*

Proof We first observe that by Property 1, no message is heard at z_j in any round $t' < t$ such that $j \notin X^{t'}$. Consider round t' for some $t' < t$ such that $j \in X^{t'}$. The adversary could not have occupied the cells in B_j in round t' (see line 4(b)i of Fig. 7) as this implies that B_j is no longer active after round t' and in particular it is not active at the beginning of round t. It follows that the adversary must have added the constraint $Q^{t'}_j$ to the constraints collection $\mathcal{C}_j$ in round t'. If this constraint is still in $\mathcal{C}_j$ at the beginning of round t, then it is satisfied and no message was heard at z_j in round t', so the assertion holds. Otherwise, the constraint $Q^{t'}_j$ must have been reduced to some $Q^{t'}_j \supset \chi \in \mathcal{C}^t_j$ or it was erased entirely. The latter case implies that $Q^{t'}_j \subseteq \mathcal{E}^t$, thus the assertion holds. In the former case, the cells in $Q^{t'}_j \setminus \chi$ were all committed to be empty, therefore (the constraint) $Q^{t'}_j$ is satisfied if and only if χ is satisfied. □

We complement Lemma 6 with the observation that for every $0 \leq j < \tau$, the constraints $\mathcal{C}_j$ are not contradicted in any round $t < \tau$. This fact is established simply by noticing that neither occupying all uncommitted cells in a block (see line 4(b)i of Fig. 7) nor invoking Procedure `Canonize` (see line 5 of Fig. 7) contradict the constraints in $\mathcal{C}_j$, and since the adversary manipulates $\mathcal{C}_j$ only via one of these two actions.

The network N is a valid network in $\mathcal{N}$ and Lemma 5 implies that when the algorithm is invoked on N, the outcome of every round $1 \leq t < \tau$ agrees with the delivery status h^t reported by the adversary. Recall that z_k is the (sole) occupied target cell in N. Since $\mathcal{C}^\tau_k$ is in canonical form, it does not contain any singleton constraint, hence by committing all cells in U^τ_k to be occupied, the constraints in $\mathcal{C}^\tau_k$ are all satisfied. Therefore, by Lemma 6, the message was never heard at z_k. It follows that when the algorithm is invoked on N, it requires at least τ rounds to ensure that the message was heard by the target node z_k.

The networks of $\mathcal{N}$ can be concatenated, identifying the target of one segment with the source of the next, to form a concatenation network of arbitrary diameter D. A deterministic broadcasting algorithm $\mathcal{A}$ requires τ rounds to deliver the message through any segment in this network. Since the concatenation network contains $D/2$ segments altogether, the broadcasting process cannot be accomplished in less than $\tau D/2$ rounds. As τ is proportional to $\sqrt{g}$, the following theorem is established.

Theorem 3 *For every deterministic broadcasting algorithm $\mathcal{A}$, there exists a UDG radio network N such that $\mathcal{A}$ requires $\Omega(D\sqrt{g})$ rounds to broadcast in N under the conditional wake up model.*

3 Spontaneous wake up

3.1 Broadcasting algorithm

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, it turns out that as a pair of algorithms working together (implemented as an algorithm interleaving their steps, and thus working in time $O\left(\min\left\{D + g^2, D \log g\right\}\right)$), they achieve asymptotically optimal broadcasting time.

3.1.1 An $O(D + g^2)$-time algorithm

Our algorithm relies on three types of grids (refer to Sect. 2.1.1 for the definition of grid): the first grid, whose boxes are called *tiles*, is of precision $g\sqrt{2}$; the second grid, whose boxes are called *blocks*, is of precision $\sqrt{2}$; and the third grid, whose boxes are called *5-blocks*, is of precision $\sqrt{2}/5$. We assume without loss of generality that g is inte-

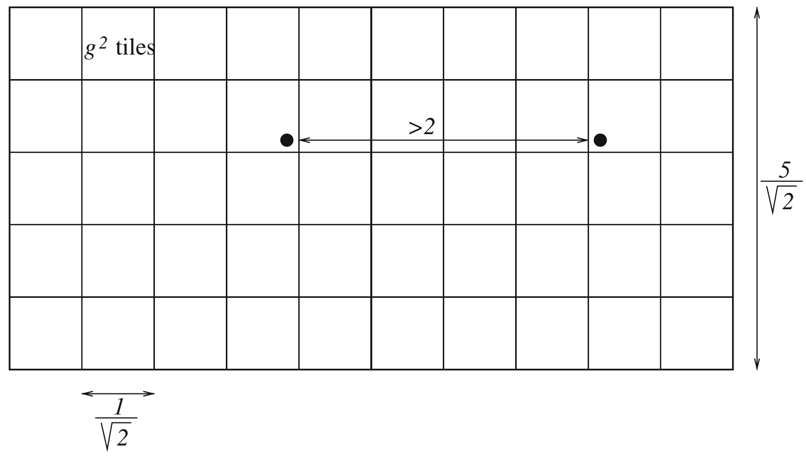

Fig. 8 Two adjacent 5-blocks

gral, hence every block consists of g^2 tiles and every 5-block consists of 25 blocks (see Fig. 8).

Enumerate all tiles in each block by integers $1, \ldots, g^2$, row by row from left to right, and all blocks in each 5-block by integers $1, \ldots, 25$, also row by row from left to right. Fix some enumeration of the 5-blocks in the infinite plane (the origin may serve as a starting point). 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 index of the 5-block where it resides, $1 \leq j \leq 25$ is the index of the block it belongs to in this 5-block, and $1 \leq i \leq g^2$ is the index of the tile it belongs to in this block. Notice that by the granularity assumption, at most one node resides in each tile, hence the hierarchical address is unique. Moreover, the Euclidean diameter of each block is 1, thus all nodes in a block are in each other's range. Also, if nodes u and v are each in the jth block of different 5-blocks, then the distance between them is greater than 2 and hence they do not create collisions while transmitting simultaneously (see Fig. 8).

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) ordered lexicographically, where $1 \leq i \leq g^2$ and $1 \leq j \leq 25$. In round (i, j) of the first phase, all nodes of hierarchical address $\langle i, j, k\rangle$ for some k transmit their coordinates. In round (i, j) of the second phase, all nodes of hierarchical address $\langle i, j, k\rangle$ for some k transmit all the information received in the first phase. Due to the properties of the three grids, 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, v) is an edge in the UDG (i.e., u and v are in each other's transmission range). Preprocessing terminates by selecting one such adjacent pair for any pair of blocks in which adjacent pairs exist (e.g., the lexicographically first pair). In that case, we say that the nodes u and v are *paired*.

Source message transmission: The source message transmission part is similar to that in Algorithm Elect-and-Broadcast in [15]. 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 jth block of a 5-block; (2) was paired with some node residing in a $\hat{j}$th block of a 5-block 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 paired with some node residing in a jth block of a 5-block 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 index j are at distance greater than 2.

Lemma 7 *Algorithm* `Elect&Transmit` *completes broadcasting in any unknown UDG radio network of diameter D and granularity g in time $O(D + g^2)$.*

Proof The preprocessing part is completed in time $O(g^2)$. We argue that if a node v hears the source message for the first time in round t and w is in the transmission range of v, then w hears 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 hear the source message.

To prove the above argument, consider such nodes v and w. Suppose that v resides in block B_2 and it heard 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 hears the source message by the end of phase p. Otherwise, all nodes of B_2 hears the source message by the end of phase $p+1$.

Suppose that w is in block B_3. If $B_2 = B_3$, then we are done. Otherwise, by phase $p+2$, the node in B_2 which was elected to transmit to B_3 will do so, and consequently, all nodes in B_3 hear the source message by the end of phase $p+2$. The assertion follows as each phase lasts 1200 rounds. □

3.1.2 An $O(D \log g)$-time algorithm

The algorithm, named algorithm `Log_Wave`, relies on the following procedure.

Procedure `Conquer`**.**

The input of the procedure consists of two blocks B_1 and B_2 and a node $b_1 \in B_1$. Block B_1 is called *conquered*. If the two blocks do not admit any pair of adjacent nodes, then the procedure does nothing. Otherwise, the procedure elects a node $b_2 \in B_2$ and B_2 becomes conquered as well.

The procedure works in two steps. The first step begins with a round in which the nodes in $\{b_1\} \cup B_2$ transmit a control message. Let C be the set of nodes in B_1 that did not hear anything on that round. Observe that every node in C must have a neighbor in B_2 (as otherwise it would have heard the control message transmitted by b_1). Next, we employ Procedure `Echo` with b_1 serving as the distinguished node λ (see Sect. 2.1.1) to appoint a single node $u \in C$ within $O(\log g)$ rounds. In the last round of the first step the newly appointed node $u \in C$ transmits its coordinates so that every node in B_1 knows it. If C is empty, then the nodes in B_1 learns about it by not hearing any message on that last round, in that case, b_1 itself is appointed.

The second step of Procedure `Conquer` is dedicated to electing a node b_2 in B_2 such that b_2 is a neighbor of the appointed node (either $u \in C$ or b_1). (Note that the appointed node may have many neighbors in B_2. On the other hand, if b_1 was appointed, then it may be the case that it does not have any neighbor in B_2. This implies that the two blocks do not admit any pair of adjacent nodes and the procedure is not assumed to do anything.) Once again, this is done in a logarithmic number of rounds via an application of Procedure `Echo` with the appointed node serving as the distinguished node λ. Altogether, Procedure `Conquer` requires $O(\log g)$ rounds.

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 $1 \le j \ne \hat{j} \le 25$. 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 jth 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`. On the last round of the subphase the newly elected node in B_2 transmits the source message.

Lemma 8 *Algorithm* `Log_Wave` *completes broadcasting in any unknown 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.

Theorem 4 *There exists a deterministic algorithm that completes broadcasting in any unknown UDG radio network of diameter D and granularity g in time $O\left(\min\left\{D+g^2, D\log g\right\}\right)$ under the spontaneous wake up model.*

3.2 Lower bound

We prove that for any choice of parameters D and g, and for every deterministic algorithm $\mathcal{A}$, there exists a UDG radio network N of diameter D and granularity g such that $\mathcal{A}$ requires $\Omega\left(\min\left\{D+g^2, D\log g\right\}\right)$ rounds to broadcast in N under the spontaneous wake up model. The UDG radio networks that serve us to prove this lower bound are of a specific type that we call *chain networks*.

For ease of reference, the reader may use Table 2 which summarizes the notations used throughout this section. Fix $\delta = \sqrt{1/17}$ and let $\rho = \lfloor g\delta \rfloor + 1$. A chain network of diameter D and granularity g consists of a *source cell*, which is always occupied by a *source node*, and $D\rho^2$ *chain cells* that may be occupied with *chain nodes* (refer to Sect. 2.2 for the definition of cells). The chain cells are arranged in pairwise

Table 2 A table of notations for Sect. 3.2

Notation	Definition
δ	$\sqrt{1/17}$
ρ	$\lfloor g\delta \rfloor + 1$
$\Gamma_1, \ldots, \Gamma_D$	Subsets of chain cells
$C_{k,j}$	A chain cell
$L_1, \ldots, L_k$	Clusters of chain nodes
L_0	The designated source cluster
λ_k	The first round in which exactly one node in L_k transmitted
T^t	The subset of cells that were scheduled to transmit on round t
T_k^t	$T^t \cap \Gamma_k$
$\#\mathcal{O}(A)$	Number of occupied cells in A
$X_{k,j}$	The variable determined by whether $C_{k,j}$ is occupied or not
$\mathcal{S}_k$	A system of linear equations over $GF(2)$ associated with L_k
$M_k\mathbf{X}_k = \mathbf{h}_k$	The linear system $\mathcal{S}_k$ in vector notation
$\mathcal{S}_k^t$	$\mathcal{S}_k$ at the beginning of round t
$\mathbf{V}(A)^{\mathrm{T}}$	A binary row vector with 1's in the entries corresponding to A
$\mathbf{V}(C)^{\mathrm{T}}$	Short for $\mathbf{V}(\{C\})^{\mathrm{T}}$
μ	$\lfloor 2\log\rho \rfloor - \lceil \log\log\rho \rceil - 2$
φ_k	$\min\left\{\mu\left\lfloor\frac{k-1}{2}\right\rfloor, \left\lfloor\frac{\rho^2}{2}\right\rfloor\right\}$
B_k	A basis for the row space of M_k
$\mathcal{R}_k$	The linear system that corresponds to B_k

disjoint subsets $\Gamma_1, \ldots, \Gamma_D$, each of size ρ^2. We index the chain cells so that $\Gamma_k = \{C_{k,j} \mid 0 \le j < \rho^2\}$ for every $1 \le k \le D$. The chain nodes occupying the chain cells in Γ_k constitute the *cluster* L_k. For simplicity, we assume that the source node is the unique node in a designated cluster L_0. The clusters are arranged in a chain, such that a node in L_k neighbors precisely the nodes in the clusters L_{k-1}, L_k, L_{k+1} (if exist). This is guaranteed by positioning the source cell in $(0,0)$ and the chain cell $C_{k,j}$ in $\left(3k\delta + \left\lfloor\frac{j}{\rho}\right\rfloor d, (j \bmod \rho)\, d\right)$ for every $1 \le k \le D$ and $0 \le j < \rho^2$. The embedding of a chain network in the Euclidean plane is illustrated in Fig. 9. We leave it to the reader to verify that the Euclidean distance between any two cells $C_{k,j} \in \Gamma_k$ and $C_{l,j'} \in \Gamma_l$ is at most 1 if and only if $|k-l| \le 1$. Each cluster contains at least one node, namely, there exists at least one occupied cell in Γ_k for every $1 \le k \le D$. Consequently, the chain network is connected and has diameter D.

We present an adversary whose goal is to slow down any deterministic algorithm broadcasting a message from the source node in chain networks. For every $1 \le k \le D$, let λ_k be the first round in which exactly one node in L_k transmitted. The adversary is designed to guarantee that λ_k is (at least) proportional to $\min\left\{\rho^2, k\log\rho\right\}$ for every $1 \le k \le D$. This implies the desired lower bound on the time required for broadcasting in chain networks as the source message cannot be heard at cluster L_D prior to time λ_{D-1}. (Clearly, broadcasting cannot be completed in less than D rounds.)

In each round, the algorithm schedules a subset T of the cells in $\bigcup_{k=0}^{D}\Gamma_k$ to transmit. The adversary then decides, for every $0 \le k \le D$, whether the nodes in the cluster L_k heard any message. This information is reported to the algorithm that can employ it to design its next moves. Let T^t be the subset of cells that were scheduled to transmit in round t and let $T_k^t = T^t \cap \Gamma_k$ for every $0 \le k \le D$. Clearly, the nodes in L_k heard a message in round t if and only if there is exactly one occupied cell in $T_{k-1}^t \cup T_k^t \cup T_{k+1}^t$.

We say that the algorithm can *distinguish* between two scenarios if there may exist some node in receiving mode under the two scenarios that hears some message in one scenario and does not hear anything (silence or collision) in the other. Given a subset $A \subseteq \bigcup_{k=0}^{D}\Gamma_k$, we denote the number of occupied cells in A by $\#\mathcal{O}(A)$. Consider round t of the execution and suppose that $\#\mathcal{O}(T_k^t) = 1$ for some k. For every $k' \in \{k-2, k-1, k+1, k+2\}$, there exists a scenario in which the algorithm distinguishes between the case $\#\mathcal{O}(T_{k'}^t) = 0$ and the case $\#\mathcal{O}(T_{k'}^t) \ge 2$. For example, assume that $\#\mathcal{O}(T_{k+1}^t) = 0$. Then a node in L_{k+1} hears the message transmitted by the unique occupied cell in T_k^t if $\#\mathcal{O}(T_{k+2}^t) = 0$ and does not hear anything if $\#\mathcal{O}(T_{k+2}^t) > 0$. On the other hand, we have the following proposition.

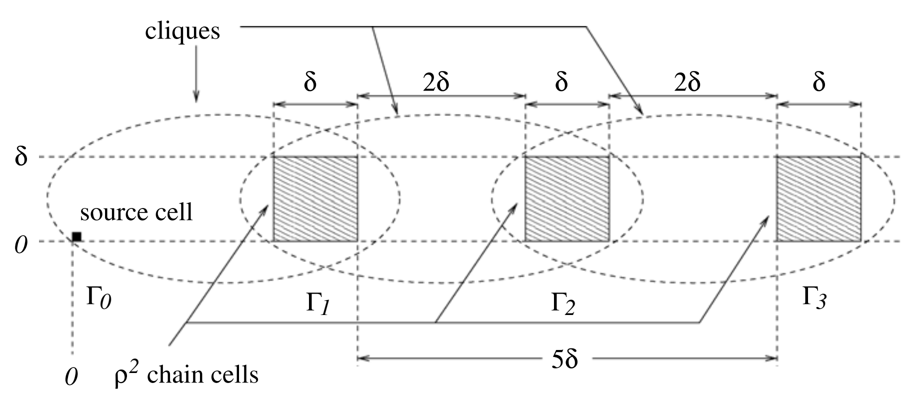

Fig. 9 The first four clusters of a chain network

Proposition 1 *If $t < \min\{\lambda_{k+i} \mid -2 \le i \le 2\}$, then the algorithm does not distinguish between the case $\#\mathcal{O}(T_k^t) = 0$ and the case $\#\mathcal{O}(T_k^t) \ge 2$.*

Given some $1 \le k \le D$ and $0 \le j < \rho^2$, let $X_{k,j}$ be the variable determined by whether the cell $C_{k,j}$ is occupied or not, namely, $X_{k,j} = 0$ if $C_{k,j}$ is empty, and $X_{k,j} = 1$ if $C_{k,j}$ is occupied. Throughout the execution, the adversary maintains a system $\mathcal{S}_k$ of linear equations over $GF(2)$ associated with the cluster L_k, for every $1 \le k \le D$. Each linear equation in $\mathcal{S}_k$ is of the form $X_{k,j_1} + \cdots + X_{k,j_r} = b$, where $0 \le j_1 < \cdots < j_r < \rho^2$ and $b \in \{0, 1\}$. (Recall that summation over $GF(2)$ is equivalent to binary exclusive or.)

The linear equation $X_{k,j_1} + \cdots + X_{k,j_r} = 0$ implies that the number of occupied cells in $\{C_{k,j_1}, \ldots, C_{k,j_r}\}$ is even and in particular $\neq 1$. Therefore, if $T_k^t = \{C_{k,j_1}, \ldots, C_{k,j_r}\}$, then a message that was scheduled to be transmitted in round t by some cell in Γ_k could not have been heard anywhere, as either this cell is empty or the message collided with some message transmitted from another occupied cell in Γ_k. Let $\mathcal{S}_k^t$ denote the linear system $\mathcal{S}_k$ at the beginning of round t.

In vector notation, we write $M_k\mathbf{X}_k = \mathbf{h}_k$, where M_k is the matrix of coefficients, $\mathbf{X}_k$ is the column vector of variables and $\mathbf{h}_k$ is the column vector of non-homogeneous terms, all corresponding to the linear system $\mathcal{S}_k$. Note that $\mathbf{X}_k$ uniquely determines which cells in Γ_k are occupied and which cells are empty. We define the *rank* of the linear system $\mathcal{S}_k$ to be $\text{rank}\,(\mathcal{S}_k) = \text{rank}\,(M_k)$, i.e., the dimension of the row space of M_k. Generally speaking, we will make sure that $\text{rank}\,(\mathcal{S}_k) < \rho^2$ which enables us to compute a non-trivial solution, thus ensuring that each cluster contains at least one occupied cell.

Consider a subset $A \subseteq \Gamma_k$. Let $\mathbf{V}(A)^{\mathrm{T}}$ be a binary ρ^2-dimensional row vector with 1's in the entries corresponding to the cells in A, that is, $\mathbf{V}(A)^{\mathrm{T}}[j] = 1$ if and only if $C_{k,j} \in A$ for every $0 \le j < \rho^2$. When A is a singleton, say $A = \{C\}$, we may write $\mathbf{V}(C)^{\mathrm{T}}$ instead of $\mathbf{V}(\{C\})^{\mathrm{T}}$. Initially, the system $\mathcal{S}_k$ does not contain any equation, i.e., $\mathcal{S}_k^1 = \emptyset$. On round t, the adversary may add to $\mathcal{S}_k$ some linear equations on the variables corresponding to the cells in T_k^t. Therefore the system $\mathcal{S}_k^{t+1}$ contains the linear equations of $\mathcal{S}_k^t$ plus (possibly) some new equations of the form $\mathbf{V}(A)^{\mathrm{T}}\mathbf{X}_k = b$, where $A \subseteq T_k^t$ and $b \in \{0, 1\}$.

Let $\mu = \lfloor 2\log\rho \rfloor - \lceil \log\log\rho \rceil - 2$. The adversary treats each cluster L_k in three phases. The first phase lasts from the beginning of the execution until round $\varphi_k = \min\left\{\mu\left\lfloor\frac{k-1}{2}\right\rfloor, \left\lfloor\frac{\rho^2}{2}\right\rfloor\right\}$. On round t, for every $1 \le t \le \varphi_k$, the adversary adds the linear equation $\mathbf{V}(T_k^t)^{\mathrm{T}}\mathbf{X}_k = 0$ to $\mathcal{S}_k$, thus ensuring that $\#\mathcal{O}(T_k^t) \neq 1$ and messages scheduled to be transmitted from cells in Γ_k are not heard anywhere. We will show later on that $\varphi_k < \min\{\lambda_{k+i} \mid -2 \le i \le 2\}$, hence, by Proposition 1, there is no need for the adversary to decide at this stage whether $\#\mathcal{O}(T_k^t) = 0$ (meaning the round results in silence) or $\#\mathcal{O}(T_k^t) \ge 2$ (resulting in a collision). Consequently, in each round of the first phase the adversary reports $\#\mathcal{O}(T_k^t) \neq 1$.

The second phase lasts from round $\varphi_k + 1$ until round $\varphi_k + \mu$. The adversarial policy in this phase is a little bit trickier since now, the clusters L_{k-1} and L_{k-2} may transmit without collisions.[1] Consider round t, for some $\varphi_k < t \le \varphi_k + \mu$, and let $i = t - \varphi_k$. Let Q be the collection of linear equations over $GF(2)$ that contains the equation $\mathbf{V}(C_{k,j})^{\mathrm{T}}\mathbf{X}_k = 0$ for every $C_{k,j} \in T_k^t$. The adversary tests whether

$$\text{rank}\,(\mathcal{S}_k \cup Q) \le \text{rank}\,(\mathcal{S}_k) + 2^{\lfloor 2\log\rho \rfloor - i - 1}. \tag{3}$$

If inequality (3) holds, then the adversary adds the linear equations of Q to $\mathcal{S}_k$, thus ensuring that the cells in T_k^t are all empty. Otherwise, the adversary adds the equation $\mathbf{V}(T_k^t)^{\mathrm{T}}\mathbf{X}_k = 0$ to $\mathcal{S}_k$ and marks T_k^t as *'noisy'*. This latter action can be interpreted as a commitment that the adversary makes to satisfy the inequality $\#\mathcal{O}(T_k^t) \ge 2$, hence to ensure collision. We will see below how this is done. Therefore on each round of the second phase the adversary reports $\#\mathcal{O}(T_k^t) = 0$ if inequality (3) holds, and $\#\mathcal{O}(T_k^t) \ge 2$ if inequality (3) does not hold.

Note that up to this stage, the linear system $\mathcal{S}_k$ is homogeneous, namely, the vector $\mathbf{h}_k$ consists of zeros only. When the second phase ends (at the end of round $\varphi_k + \mu$), the adversary performs the following process (referred to as the "*noise producing*" process). It first computes a basis B_k for the row space of M_k and replaces the system $\mathcal{S}_k$ by the (equivalent) system $\mathcal{R}_k$ that consists of the linear equations $\mathbf{U}^{\mathrm{T}}\mathbf{X}_k = 0$ for all $\mathbf{U}^{\mathrm{T}} \in B_k$. Next, for every $\varphi_k < t' \le \varphi_k + \mu$ such that $T_k^{t'}$ is marked 'noisy', the adversary finds some $C_{k,j} \in T_k^{t'}$ such that $\mathbf{V}(C_{k,j})^{\mathrm{T}}$ is linearly independent with B_k (we will prove soon that such $C_{k,j}$ exists). The adversary then adds the linear equation $\mathbf{V}(C_{k,j})^{\mathrm{T}}\mathbf{X}_k = 1$ to $\mathcal{R}_k$ and the row vector $\mathbf{V}(C_{k,j})^{\mathrm{T}}$ to B_k. Note that since the equation $\mathbf{V}(T_k^{t'})^{\mathrm{T}}\mathbf{X}_k = 0$ (which implies an even number of occupied cells) was already added to $\mathcal{S}_k$ (on round t'), and since $C_{k,j} \in T_k^{t'}$, the equation $\mathbf{V}(C_{k,j})^{\mathrm{T}}\mathbf{X}_k = 1$ implies $\#\mathcal{O}(T_k^{t'}) \ge 2$.

Finally, the adversary computes a non-trivial solution for $\mathcal{R}_k$ and reveals it to the algorithm (we will prove soon that a non-trivial solution exists). Consequently, in the third phase, that lasts from round $\varphi_k + \mu + 1$ until the end of the execution, the adversary does not make any decisions regarding the cluster L_k (as the algorithm is fully aware of the occupied and empty cells in Γ_k). The adversarial policy is described formally in Fig. 10.

We now turn to establish the validity of the noise producing process.

[1] Note that once the cluster L_{k-1} can transmit without collisions, the algorithm may be able to run Procedure `Echo` (see Sect. 2.1.1) and to select a unique node in L_k (which in turn, can transmit alone) within a logarithmic number of rounds.

Initially, set $\mathcal{S}_k \leftarrow \emptyset$ for every $1 \le k \le D$.
In round $1 \le t \le \varphi_k + \mu$ of the execution, for $k = 1, \ldots, D$, do:

1. If $t \le \varphi_k$, then add the equation $\mathbf{V}(T_k^t)^{\mathrm{T}}\mathbf{X}_k = 0$ to $\mathcal{S}_k$ and report $\#\mathcal{O}(T_k^t) \ne 1$.
2. If $\varphi_k < t \le \varphi_k + \mu$, then do:
   (a) Set $i \leftarrow t - \varphi_k$.
   (b) Set $Q \leftarrow \left\{\mathbf{V}(C_{k,j})^{\mathrm{T}}\mathbf{X}_k = 0 \mid C_{k,j} \in T_k^t\right\}$.
   (c) If $\mathrm{rank}\,(\mathcal{S}_k \cup Q) \le \mathrm{rank}\,(\mathcal{S}_k) + 2^{\lfloor 2\log\rho \rfloor - i - 1}$, then set $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup Q$ and report $\#\mathcal{O}(T_k^t) = 0$.
   (d) Else, do:
      i. Add the equation $\mathbf{V}(T_k^t)^{\mathrm{T}}\mathbf{X}_k = 0$ to $\mathcal{S}_k$.
      ii. Mark T_k^t as 'noisy'.
      iii. Report $\#\mathcal{O}(T_k^t) \ge 2$.

At the end of round $\varphi_k + \mu$, do:

1. Let B_k be a basis for the row space of M_k and set $\mathcal{R}_k \leftarrow \{\mathbf{U}^{\mathrm{T}}\mathbf{X}_k = 0 \mid \mathbf{U}^{\mathrm{T}} \in B_k\}$.
2. For every 'noisy' $T_k^{t'}$, do:
   (a) Find a cell $C_{k,j} \in T_k^{t'}$ such that $\mathbf{V}(C_{k,j})^{\mathrm{T}}$ is not spanned by B_k.
   (b) Set $\mathcal{R}_k \leftarrow \mathcal{R}_k \cup \{\mathbf{V}(C_{k,j})^{\mathrm{T}}\mathbf{X}_k = 1\}$ and $B_k \leftarrow B_k \cup \{\mathbf{V}(C_{k,j})^{\mathrm{T}}\}$.
3. Compute a non-trivial solution for the linear system $\mathcal{R}_k$ and reveal it.

Fig. 10 The policy of the adversary in the spontaneous wake up model

Lemma 9 *The adversary succeeds in maintaining a linear system $\mathcal{R}_k$ that eventually (at the end of the noise producing process) admits a non-trivial solution $\mathbf{X}_k$ for every $1 \le k \le D$.*

Proof We first show that for every 'noisy' $T_k^{t'}$, there exists some $C_{k,j} \in T_k^{t'}$ such that $\mathbf{V}(C_{k,j})^{\mathrm{T}}$ is not spanned by the vectors in B_k (recall that this is essential for the operation of the adversary in the noise producing process). Consider round t' for some $\varphi_k < t' \le \varphi_k + \mu$ and let $Q = \left\{\mathbf{V}(C_{k,j})^{\mathrm{T}}\mathbf{X}_k = 0 \mid C_{k,j} \in T_k^{t'}\right\}$. Suppose that $\mathrm{rank}\left(\mathcal{S}_k^{t'} \cup Q\right) > \mathrm{rank}\left(\mathcal{S}_k^{t'}\right) + 2^{\lfloor 2\log\rho \rfloor - i' - 1}$, where $i' = t' - \varphi_k$, so that $T_k^{t'}$ is marked 'noisy'. The system $\mathcal{S}_k^{t'+1}$ is obtained by adding the (sole) linear equation $\mathbf{V}(T_k^{t'})^{\mathrm{T}}\mathbf{X}_k = 0$ to $\mathcal{S}_k^{t'}$, thus $\mathrm{rank}\left(\mathcal{S}_k^{t'+1}\right) \le \mathrm{rank}\left(\mathcal{S}_k^{t'}\right) + 1$ and $\mathrm{rank}\left(\mathcal{S}_k^{t'+1} \cup Q\right) \ge \mathrm{rank}\left(\mathcal{S}_k^{t'+1}\right) + 2^{\lfloor 2\log\rho \rfloor - i' - 1}$.

On each round $t' < t \le \varphi_k$, some more homogeneous linear equations might have been added to $\mathcal{S}_k$, but by the time the noise producing process begins (at the end of the second phase), the rank of $\mathcal{S}_k$ increases by at most

$$\sum_{i=i'+1}^{\mu} 2^{\lfloor 2\log\rho \rfloor - i - 1} = \sum_{i=\lceil \log\log\rho \rceil + 1}^{\lfloor 2\log\rho \rfloor - i' - 2} 2^i = 2^{\lfloor 2\log\rho \rfloor - i' - 1} - 2^{\lceil \log\log\rho \rceil + 1} \le 2^{\lfloor 2\log\rho \rfloor - i' - 1} - 2\log\rho\ ,$$

hence upon initialization of the linear system $\mathcal{R}_k$, we have $\mathrm{rank}\,(\mathcal{R}_k \cup Q) \ge \mathrm{rank}\,(\mathcal{R}_k) + 2\log\rho$. The rank of $\mathcal{R}_k$ increases by at most $\mu < 2\log\rho$ due to the linear equations we add during the noise producing process, thus when we come to deal with $T_k^{t'}$, we still have $\mathrm{rank}\,(\mathcal{R}_k \cup Q) > \mathrm{rank}\,(\mathcal{R}_k)$ and there must be at least one unit row vector in $\left\{\mathbf{V}(C_{k,j})^{\mathrm{T}} \mid C_{k,j} \in T_k^{t'}\right\}$ that is not spanned by the vectors in B_k.

It remains to show that at the end of the noise producing process the linear system $\mathcal{R}_k$ admits a non-trivial solution. First, note that when the second phase begins, there are at most $\lfloor \rho^2/2 \rfloor$ equations in $\mathcal{S}_k$. The number of equations added to $\mathcal{S}_k$ in the second phase may be large but the rank of $\mathcal{S}_k$ increases by at most

$$\sum_{i=1}^{\mu} 2^{2\log\rho - i - 1} = \sum_{i=\lceil \log\log\rho \rceil + 1}^{\lfloor 2\log\rho \rfloor - 2} 2^i = 2^{\lfloor 2\log\rho \rfloor - 1} - 2^{\lceil \log\log\rho \rceil + 1} \le \frac{\rho^2}{2} - 2\log\rho\ ,$$

thus upon construction of $\mathcal{R}_k$, its rank is at most $\rho^2 - 2\log\rho$. Since the total number of equations added to $\mathcal{R}_k$ during the noise producing process is at most $\mu < 2\log\rho$, it follows that the rank of $\mathcal{R}_k$ at the end of this process is smaller than ρ^2 (the number of variables). Therefore, as the algorithm succeeds in maintaining the vector collection B_k linearly independent, the linear system $\mathcal{R}_k$ must admit a non-trivial solution. The lemma follows. □

Next, we prove that the reports made by the adversary throughout the execution are consistent with (future) decisions relating to which cells are occupied and which cells are empty.

Lemma 10 *For every $1 \leq k \leq D$, the solution $\mathbf{X}_k$ agrees with all the reports made by the adversary throughout the execution.*

Proof Clearly, if $\mathbf{X}_k$ solves the equation $\mathbf{V}(A)^{\mathrm{T}}\mathbf{X}_k = 0$, then $\#\mathcal{O}(A) \neq 1$. Moreover, since the linear system $\mathcal{S}_k$ at the end of the second phase is homogeneous, any solution for $\mathcal{R}_k$ also solves the linear system $\mathcal{S}_k$. Therefore the reports made during the first phase (all of the form $\#\mathcal{O}(T_k^t) \neq 1$) are valid. To see that the reports of the form $\#\mathcal{O}(T_k^t) = 0$ (made during the second phase) are valid, observe that the linear system $\mathcal{S}_k$ now contains the equation $\mathbf{V}(C_{k,j})^{\mathrm{T}}\mathbf{X}_k = 0$, for every $C_{k,j} \in T_k^t$. The validity of the $\#\mathcal{O}(T_k^t) \geq 2$ reports (the other case of the second phase) is justified as the linear equation $\mathbf{V}(T_k^t)^{\mathrm{T}}\mathbf{X}_k = 0$ was added to $\mathcal{S}_k$ in round t, and since the linear equation $\mathbf{V}(C_{k,j})^{\mathrm{T}}\mathbf{X}_k = 1$ was added to $\mathcal{R}_k$ during the noise producing process for some $C_{k,j} \in T_k^t$. □

As Lemmas 9 and 10 imply $\#\mathcal{O}(T_k^t) \neq 1$ for every $1 \leq k \leq D$ and $t \leq \varphi_k + \mu$, the next corollary can be established.

Corollary 1 *The adversary guarantees $\lambda_k > \varphi_k + \mu$ for every $1 \leq k \leq D$.*

Recall that we required $\varphi_k < \min\{\lambda_{k+i} \mid -2 \leq i \leq 2\}$ so that during the first phase the adversary does not have to distinguish between $\#\mathcal{O}(T_k^t) = 0$ and $\#\mathcal{O}(T_k^t) \geq 2$. This requirement is satisfied by Corollary 1 and since $\varphi_k + \mu \geq \varphi_{k+i}$ for all $-2 \leq i \leq 2$.

Let z be a node in L_D. We prove that $\mathcal{A}$ requires $\Omega\big(\min\big\{D + g^2, D\log g\big\}\big)$ rounds until z can hear the source message. First observe that z hears the source message for the first time after $\Omega(D)$ rounds as the shortest path (in hops) from the source to z is of length D. Thus the $\Omega\left(\min\left\{D+g^2, D\log g\right\}\right)$ lower bound holds if $g^2 < D$. Assume that $g^2 \geq D$. In this case we have to show that $\mathcal{A}$ requires $\Omega\left(\min\left\{D\log g, g^2\right\}\right)$ rounds to deliver the source message to z. This holds due to Corollary 1 by the definition of φ_{D-1}, and since z cannot hear the source message prior to time λ_{D-1}.

Theorem 5 *For every deterministic broadcasting algorithm $\mathcal{A}$, there exists a UDG radio network N such that $\mathcal{A}$ requires $\Omega\left(\min\left\{D + g^2, D\log g\right\}\right)$ rounds to broadcast in N under the spontaneous wake up model.*

4 Conclusion

We presented upper and lower bounds on the time of broadcasting in ad hoc radio networks modeled as unit disc graphs. While in the spontaneous wake up model our bounds match, thus establishing an optimal broadcasting time of $\Theta\big(\min\big\{D + g^2, D\log g\big\}\big)$, in the case of the conditional wake up model a gap is left between the $O(Dg)$ upper bound and the $\Omega(D\sqrt{g})$ lower bound. After the publication of the conference version of this paper, the problem of closing this gap has been addressed in the forthcoming paper [18]: the lower bound has been strengthened to $\Omega(Dg)$, thus establishing $\Theta(Dg)$ as optimal broadcasting time is this model. This shows that the separation between the conditional and the spontaneous wake up models for broadcasting in UDG radio networks is even more significant than established in the current paper.

Another issue deserving further discussion is the assumptions underlying our models of radio networks. Some of them, such as synchronous steps in which nodes of the network transmit, or lack of the ability to detect collisions, are common in many papers in the literature on algorithmic aspects of radio communication. Among others, more particular to the present paper, are the knowledge of the granularity (or, equivalently, the density) of the network and the knowledge of exact own Euclidean coordinates by every node.

First recall that, for our results to hold, we do not need the exact value of the density d, but rather some linear lower bound on d. It might be argued that such a lower bound is usually provided, e.g., by the physical size of the sensors of which the wireless network is composed. The question of whether the minimum distance between such sensors is comparable with their physical size may depend on particular applications. Nevertheless, it is interesting to study if our broadcasting time can be preserved in the absence of any knowledge of density (and of diameter) of the network. Likewise, the assumption of the availability of exact Euclidean coordinates is fairly strong. It may be argued that, since such coordinates are obtained using a positioning device like GPS, an error is inherent in the perceived position. So, it is interesting to investigate how such an error in perceived coordinates affects the broadcasting time.

Both above issues (the total lack of knowledge of the density and the imprecise reading of coordinates by each node) have been recently addressed in the forthcoming paper [19], for the spontaneous wake up model. It turns out that it is the *combination* of lack of knowledge of the density and of imprecise perception of node positions that causes a major problem in preserving our broadcasting time. Nevertheless, a broadcasting algorithm (significantly different from the one presented in this paper) has been shown to work in the same time bound of $O\left(\min\left\{D + g^2, D\log g\right\}\right)$, under this much weaker scenario, for a large class of networks. In some other cases it has been shown that optimal broadcasting time may even be exponentially larger.

Our final comment concerns the use of unit disc graphs for modeling radio networks. While the geometric character of these graphs is a reasonable approximation of wireless networks deployed on flat terrains without large obstacles, it may be argued that the dichotomic assumption underlying their definition (the presence of an edge between nodes at distance at most 1 and the absence of such an edge between nodes at distance larger than 1) is too "sharp". One may argue that it is more realistic to introduce some "gray zone", such that nodes at distance in this zone may be joined by an edge or not, and the decision is made by an adversary in such cases. This idea is behind the notion of *quasi unit disc graphs*, cf., e.g., [30]. It would be interesting to extend our study to these more general graphs modeling radio networks.

References

1. Alon, N., Bar-Noy, A., Linial, N., Peleg, D.: A lower bound for radio broadcast. J. Comp. Syst. Sci. **43**, 290–298 (1991)
2. Avin, C., Ercal, G.: On the cover time of random geometric graphs. In: Proc. 32th Int. Colloq. on Automata, Languages and Programming (ICALP 2005). LNCS, vol. 3580, pp. 677–689 (2005)
3. Bar-Yehuda, R., Goldreich, O., Itai, A.: On the time complexity of broadcast in radio networks: an exponential gap between determinism and randomization. J. Comp. Syst. Sci. **45**, 104–126 (1992)
4. Bruschi, D., Del Pinto, M.: Lower bounds for the broadcast problem in mobile radio networks. Distributed Comput. **10**, 129–135 (1997)
5. Chlamtac, I., Kutten, S.: On broadcasting in radio networks—problem analysis and protocol design. IEEE Trans. Commun. **33**, 1240–1246 (1985)
6. Chlamtac, I., Weinstein, O.: The wave expansion approach to broadcasting in multihop radio networks. IEEE Trans. Commun. **39**, 426–433 (1991)
7. Chlebus, B., Gąsieniec, L., Gibbons, A., Pelc, A., Rytter, W.: Deterministic broadcasting in unknown radio networks. Distributed Comput. **15**, 27–38 (2002)
8. Chlebus, B., Gąsieniec, L., Östlin, A., Robson, J.M.: Deterministic radio broadcasting. In: Proc. 27th Int. Colloq. on Automata, Languages and Programming (ICALP 2000). LNCS, vol. 1853, pp. 717–728 (2000)
9. Chlebus, B., Kowalski, D.: A better wake-up in radio networks. In: Proc. 23rd Symp. on Principles of Distributed Computing (PODC 2004) (2004)
10. Chrobak, M., Gąsieniec, L., Kowalski, D.: The wake-up problem in multi-hop radio networks. In: Proc. 15th ACM-SIAM Symp. on Discrete Algorithms (SODA 2004), pp. 985–993 (2004)
11. Chrobak, M., Gąsieniec, L., Rytter, W.: Fast broadcasting and gossiping in radio networks. In: Proc. 41st Symp. on Foundations of Computer Science (FOCS 2000), pp. 575–581 (2000)
12. Clementi, A.E.F., Monti, A., Silvestri, R.: Selective families, superimposed codes, and broadcasting on unknown radio networks. In: Proc. 12th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA 2001), pp. 709–718 (2001)
13. Czumaj, A., Rytter, W.: Broadcasting algorithms in radio networks with unknown topology. In: Proc. 44th Symp. on Foundations of Computer Science (FOCS 2003), pp. 492–501 (2003)
14. De Marco, G.: Distributed broadcast in unknown radio networks. In: Proc. 19th ACM-SIAM Symp. on Discrete Algorithms (SODA 2008) (2008)
15. Dessmark, A., Pelc, A.: Broadcasting in geometric radio networks. J. Discrete Algorithms **5**, 187–201 (2007)
16. Diks, K., Kranakis, E., Krizanc, D., Pelc, A.: The impact of knowledge on broadcasting time in linear radio networks. Theor. Comp. Sci. **287**, 449–471 (2002)
17. Elkin, M., Kortsarz, G.: Improved broadcast schedule for radio networks. In: Proc. 16th ACM-SIAM Symp. on Discrete Algorithms (SODA 2005), pp. 222–231 (2005)
18. Emek, Y., Kantor, E., Peleg, D.: On the effect of the deployment setting on broadcasting in Euclidean radio networks (submitted)
19. Fusco, E., Pelc, A.: Broadcasting in UDG radio networks with missing and inaccurate information (submitted)
20. Gaber, I., Mansour, Y.: Centralized broadcast in multihop radio networks. J. Algorithms **46**, 1–20 (2003)
21. Gąsieniec, L., Pelc, A., Peleg, D.: The wakeup problem in synchronous broadcast systems. SIAM J. Discrete Math. **14**, 207–222 (2001)
22. Gąsieniec, L., Peleg, D., Xin, Q.: Faster communication in known topology radio networks. In: Proc. 24th ACM Symp. on Principles Of Distributed Computing (PODC 2005), pp. 129–137 (2005)
23. Jurdzinski, T., Stachowiak, G.: Probabilistic algorithms for the wakeup problem in single-hop radio networks. In: Proc. 13th Int. Symp. on Algorithms and Computation (ISAAC 2002). LNCS, vol. 2518, pp. 535–549 (2002)
24. Kesselman, A., Kowalski, D.: Fast Distributed Algorithm for Convergecast in Ad Hoc Geometric Radio Networks. In: Proc. 2nd Int. Conf. on Wireless on Demand Network Systems and Service (WONS 2005), pp. 119–124 (2005)
25. Kowalski, D., Pelc, A.: Time of deterministic broadcasting in radio networks with local knowledge. SIAM J. Comput. **33**, 870–891 (2004)
26. Kowalski, D., Pelc, A.: Time complexity of radio broadcasting: adaptiveness vs. obliviousness and randomization vs. determinism. Theor. Comp. Sci. **333**, 355–371 (2005)
27. Kowalski, D., Pelc, A.: Broadcasting in undirected ad hoc radio networks. Distributed Comput. **18**, 43–57 (2005)
28. Kowalski, D., Pelc, A.: Optimal deterministic broadcasting in known topology radio networks. Distributed Comput. **19**, 185–195 (2007)
29. Kranakis, E., Krizanc, D., Pelc, A.: Fault-tolerant broadcasting in radio networks. J. Algorithms **39**, 47–67 (2001)
30. Kuhn, F., Zollinger, A.: Ad-hoc networks beyond unit disk graphs. In: Proc. DIALM-POMC Joint Workshop on Foundations of Mobile Computing, pp. 69–78 (2003)
31. Kushilevitz, E., Mansour, Y.: An $\Omega(D\log(N/D))$ lower bound for broadcast in radio networks. SIAM J. Comput. **27**, 702–712 (1998)
32. Moscibroda, T., Wattenhofer, R.: Maximal independent sets in radio networks. In: Proc. 24th ACM Symp. on Principles of Distributed Computing (PODC 2005), pp. 148–157 (2005)
33. Moscibroda, T., Wattenhofer, R.: Coloring unstructured radio networks. In: Proc. 17th ACM Symp. on Parallel Algorithms (SPAA 2005), pp. 39–48 (2005)
34. Muthukrishnan, S., Pandurangan, G.: The bin-covering technique for thresholding random geometric graph properties. In: Proc. 16th ACM-SIAM Symp. on Discrete Algorithms (SODA 2005), pp. 989–998 (2005)
35. Ravishankar, K., Singh, S.: Broadcasting on $[0, L]$. Discrete Appl. Math. **53**, 299–319 (1994)
36. Sen, A., Huson, M. L.: A New Model for Scheduling Packet Radio Networks. In: Proc. 15th Joint Conf. of the IEEE Computer and Communication Societies (IEEE INFOCOM 1996), pp. 1116–1124 (1996)