Distrib. Comput. (2005) 18: 43–57
Digital Object Identifier (DOI) 10.1007/s00446-005-0126-7

DISTRIBUTED
COMPUTING

Broadcasting in undirected ad hoc radio networks

Dariusz R. Kowalski[1,*], **Andrzej Pelc**[2,**]

[1] Instytut Informatyki, Uniwersytet Warszawski, Banacha 2, 02-097 Warszawa, Poland (e-mail: darek@mimuw.edu.pl)
[2] Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada (e-mail: pelc@uqo.ca)

Received: August 1, 2003 / Accepted: March 18, 2005
Published online: June 15, 2005 –

Abstract. We consider distributed broadcasting in radio networks, modeled as undirected graphs, whose nodes have no information on the topology of the network, nor even on their immediate neighborhood. For randomized broadcasting, we give an algorithm working in expected time $\mathcal{O}(D\log(n/D)+\log^2 n)$ in n-node radio networks of diameter D, which is optimal, as it matches the lower bounds of Alon et al. [1] and Kushilevitz and Mansour [16]. Our algorithm improves the best previously known randomized broadcasting algorithm of Bar-Yehuda, Goldreich and Itai [3], running in expected time $\mathcal{O}(D\log n+\log^2 n)$. (In fact, our result holds also in the setting of n-node *directed* radio networks of radius D.) For deterministic broadcasting, we show the lower bound $\Omega(n\frac{\log n}{\log(n/D)})$ on broadcasting time in n-node radio networks of diameter D. This implies previously known lower bounds of Bar-Yehuda, Goldreich and Itai [3] and Bruschi and Del Pinto [5], and is sharper than any of them in many cases. We also give an algorithm working in time $\mathcal{O}(n\log n)$, thus shrinking – for the first time – the gap between the upper and the lower bound on deterministic broadcasting time to a logarithmic factor.

Keywords: Broadcasting – Distributed – Deterministic – Randomized – Radio network

1 Introduction

A radio network is a collection of transmitter-receiver devices referred to as *nodes*. It is modeled as an undirected connected graph on the set of these nodes. An edge e between two nodes means that the transmitter of one end of e can reach the other end. Nodes send messages in synchronous *steps* (time slots). In every step every node acts either as a *transmitter* or as a *receiver*. A node acting as a transmitter sends a message which can potentially reach all of its neighbors. A node acting as a receiver in a given step gets a message, if and only if, exactly one of its neighbors transmits in this step. The message received in this case is the one that was transmitted. If at least two neighbors v and v' of u transmit simultaneously in a given step, none of the messages is received 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 more than one of its neighbors transmitting is the same as that of no neighbor transmitting, i.e., a node cannot distinguish a collision from silence.

Broadcasting is one of basic tasks in network communication. Its goal is to transmit a message from one node of the network, called the *source*, to all other nodes. Remote nodes get the source message via intermediate nodes, along paths in the network. It is assumed that only nodes that already received the source message can transmit, i.e., there are no "spontaneous" transmissions of nodes other than the source. In this paper we concentrate on one of the most important and widely studied performance parameters of a broadcasting scheme, which is the total time, i.e., the number of steps it uses to inform all the nodes of the network. Broadcasting time is considered as a function of two parameters of the network: the number n of nodes, and the radius D, which is the largest distance from the source to any node of the network. (For undirected graphs, the diameter is of the order of the radius.)

We consider distributed broadcasting in ad hoc radio networks. This means that nodes do not have any *a priori* knowledge about the topology of the network, nor even on their immediate neighborhood: the only *a priori* knowledge of a node is its own label, and a linear upper bound on the number of nodes. Broadcasting in ad hoc radio networks was investigated, e.g., in [4,5,7–10].

1.1 Related work

Deterministic centralized broadcasting assuming complete knowledge of the network was considered in [6], where a $\mathcal{O}(D\log^2 n)$-time broadcasting algorithm was given for all n-node networks of radius D. In [12], $\mathcal{O}(D+\log^5 n)$-time broadcasting was proposed. On the other hand, in [1] the au-

* This work was done during the stay of Dariusz Kowalski at the Research Chair in Distributed Computing of the Université du Québec en Outaouais, as a postdoctoral fellow. Research supported in part by KBN grant 4T11C04425.

** Research of Andrzej Pelc was supported in part by NSERC discovery grant and by the Research Chair in Distributed Computing of the Université du Québec en Outaouais.

thors proved the existence of a family of n-node networks of radius 2, for which any broadcast requires time $\Omega(\log^2 n)$.

One of the first papers to study deterministic distributed broadcasting in radio networks whose nodes have only limited knowledge of the topology, was [3]. The authors assumed that nodes know only their own label and labels of their neighbors. Under this scenario, a simple linear-time broadcasting algorithm based on DFS follows from [2].

Many authors [4,5,7–10] studied deterministic distributed broadcasting in radio networks under the assumption (also used in the present paper) that nodes know only their own label (but not labels of their neighbors). In [7] the authors gave a broadcasting algorithm working in time $\mathcal{O}(n)$ for arbitrary n-node networks, assuming that nodes can transmit spontaneously, before getting the source message. For this model, a matching lower bound $\Omega(n)$ on deterministic broadcasting time was proved in [15] even for the class of networks of constant diameter. On the other hand, in [5] a lower bound $\Omega(D \log n)$ was proved for n-node networks of radius D, if spontaneous transmissions are not allowed.

In [7–9,14,11] the model of directed graphs was used. Increasingly faster broadcasting algorithms working on arbitrary (directed) radio networks were constructed, the currently fastest being the $\mathcal{O}(n \log^2 D)$-time algorithm from [11]. On the other hand, in [10] a lower bound $\Omega(n \log D)$ on broadcasting time was proved for directed n-node networks of radius D.

Finally, randomized broadcasting algorithms in radio networks were studied, e.g., in [3,16]. For these algorithms, no topological knowledge of the network was assumed. In [3] the authors showed a randomized broadcasting algorithm running in expected time $\mathcal{O}(D \log n + \log^2 n)$. In [16] it was shown that for any randomized broadcasting algorithm and parameters $D \leq n$, there exists an n-node network of radius D requiring expected time $\Omega(D \log(n/D))$ to execute this algorithm (even if labels of nodes are from the set $\{1, \dots n\}$). 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.

1.2 Our results

Our main result for randomized broadcasting is an algorithm working in expected time $\mathcal{O}(D \log(n/D) + \log^2 n)$ in n-node radio networks of radius D. This result holds also for n-node *directed* radio networks of radius D, and we carry out the analysis in this more general setting. This complexity is optimal in view of the lower bounds $\Omega(\log^2 n)$ of Alon et al. [1] and $\Omega(D \log(n/D))$ of Kushilevitz and Mansour [16]. Our algorithm improves the best previously known randomized broadcasting algorithm of Bar-Yehuda, Goldreich and Itai [3], running in expected time $\mathcal{O}(D \log n + \log^2 n)$. Our algorithm is faster, e.g., for radius $D \in \Theta(n/\text{polylog}\,(n))$.

Shortly after the conference version of the present paper, a randomized broadcasting algorithm working in the same time $\mathcal{O}(D \log(n/D) + \log^2 n)$, with high probability, has been independently presented in [11].

Our main result for deterministic broadcasting is the lower bound $\Omega(n \frac{\log n}{\log(n/D)})$ on broadcasting time in n-node radio networks of radius D: for any broadcasting algorithm we construct an n-node network of radius $\Theta(D)$, on which this algorithm requires time $\Omega(n \frac{\log n}{\log(n/D)})$ to broadcast. The two sharpest lower bounds known previously on deterministic broadcasting time were: the lower bound $\Omega(n)$ from [3,15] (even for the class of networks with constant radius), and the lower bound $\Omega(D \log n)$ from [1,5]. (It should be noted that the linear lower bound in [3] was incorrectly claimed under the stronger scenario assuming knowledge of the neighborhood, but the argument can be modified to correctly prove this lower bound under our present scenario, cf. [15]).

Our lower bound implies both these results, and is sharper than any of them, e.g., for $D \in \Theta(n/\text{polylog}(n))$. The technique used to prove it is different from both previous lower bounds. Given any broadcasting algorithm, we construct a network on which it works slowly, by combining two types of objects: families of sets used for "jamming" potential messages, and selective families of sets. Jamming is used to be able to derive lower bounds on broadcasting time from lower bounds on the size of selective families.

As for upper bounds on deterministic broadcasting time, the fastest algorithm known to date was the algorithm from [14], running in time $\mathcal{O}(n \log n \log D)$ and designed for directed networks but working for undirected ones as well. We construct a faster broadcasting algorithm working in time $\mathcal{O}(n \log n)$ for undirected n-node networks. Thus our algorithm is the first to exceed the optimal time by at most a logarithmic factor, for arbitrary (undirected) networks. Together with our result, the new $\mathcal{O}(n \log^2 D)$-time algorithm presented in [11] after the conference version of the present paper, gives an upper bound $\mathcal{O}(n \cdot \min\{\log^2 D, \log n\})$ on broadcasting time in undirected radio networks.

In [10], the authors proved a lower bound $\Omega(n \log D)$ on deterministic broadcasting time in *directed* n-node networks of radius D. More precisely, for any broadcasting algorithm, they constructed a directed n-node network of radius D, in which there are edges from all nodes of the ith layer to all nodes of the $(i+1)$th layer (such networks are called *complete layered networks*, cf. [9]), such that this algorithm requires time $\Omega(n \log D)$ to broadcast on this network. It was claimed in [10] that the same argument shows the lower bound $\Omega(n \log D)$ on broadcasting time for *undirected* networks, if spontaneous transmissions are not allowed. We prove that this extension is incorrect. Indeed, we construct a broadcasting algorithm which works in time $\mathcal{O}(n + D \log n)$ for all undirected complete layered n-node networks of radius D. For all unbounded $D \in o(n)$ this is faster than the claimed lower bound. Our algorithm is optimal for complete layered networks, in view of [5,3].

An interesting corollary of our results is the following: n-node complete layered networks of radius D are the most difficult for randomized broadcasting but they are not for deterministic broadcasting. Indeed, the lower bound from [16] on randomized broadcasting time (which is tight in view of our upper bound) was established for complete layered networks. On the other hand, in the case of deterministic broadcasting, the comparison of our upper bound $\mathcal{O}(n + D \log n)$ for complete layered networks with our lower bound $\Omega(n \frac{\log n}{\log(n/D)})$ for arbitrary networks shows that complete layered networks are *not* the most difficult to broadcast deterministically, for some values of D.

1.3 Model and terminology

We consider networks modeled as undirected connected graphs whose nodes have distinct labels belonging to the set $\{0, 1, ..., r\}$, where r is linear in the number n of nodes. We assume that each node knows *a priori* only its own label and the parameter r. A distinguished node with label 0 is called the *source*. We denote by D the *radius* of the graph, i.e., the distance from the source to the farthest node. The jth *layer* of a graph is the set of nodes at distance j from the source. A *complete layered* network is a graph in which adjacent pairs of nodes are exactly those from consecutive layers. We adopt the same definition of broadcasting time, as e.g., in [9,16]. A broadcasting algorithm works in time t, if every node gets the source message after at most t steps.

It should be noted that our assumption that nodes know the parameter $r = \mathcal{O}(n)$ such that labels belong to the set $\{0, 1, ..., r\}$ is *not* an insignificant generalization of a stronger assumption that nodes know n and all labels are from the set $\{0, 1, ..., n-1\}$. An example of the difference between these assumptions is the problem of deterministic broadcasting in the class of networks considered in [3]. In our present model where each node knows only its own label (and not labels of neighbors, as in [3]) the difference between the above assumptions can be seen in a rather dramatic way. If all nodes know n and labels are from the set $\{0, 1, ..., n-1\}$ then, using procedure Echo and Algorithm Binary-Selection described in Sect. 4, it is possible to broadcast in time $\mathcal{O}(\log n)$ in these networks. However, if only r is known to nodes, the arguments from [3,15] show that the lower bound on broadcasting time for this class of networks is $\Omega(n)$.

2 Randomized broadcasting

In this section we design and analyze a randomized broadcasting algorithm whose expected running time on any n-node network of radius D is $\mathcal{O}(D\log(n/D) + \log^2 n)$. Our algorithm is optimal in view of the lower bounds from [1,16].

Although in this paper we consider undirected graphs, this particular result holds in the more general setting of directed graphs as well (undirected graphs can be considered as directed with every edge replaced by two directed edges in opposite directions). Hence (only in this section) we work in the directed setting. In this setting D denotes the *directed radius*, which is the minimum length of the *directed* path from the source to the furthest node.

First suppose that r and D are powers of 2, and that D is known to all nodes. We will later show how these assumptions can be removed without changing our result.

We start by presenting the general idea of the algorithm. It works in $\mathcal{O}(D)$ stages, each consisting of $\log(r/D) + 2$ steps. During the first $\log(r/D)+1$ steps of each stage, transmission probabilities of nodes are chosen in such a way that every node with at most r/D informed in-neighbors gets the source message with a constant probability. Nodes with many informed in-neighbors are taken care of in the last step of each stage. In these steps, transmission probabilities are carefully constructed to ensure that after x in-neighbors of a node get the source message (for $x > r/D$), this node is also informed with a constant probability, after $\mathcal{O}(r/x)$ stages, or, in some cases, after $\mathcal{O}((r\log r)/x)$ stages. Our analysis shows that last steps of $\mathcal{O}(D)$ stages are enough to inform all such nodes. This analysis is significantly complicated by the fact that broadcasting progress occurs (with high probability) in different time intervals. Notice that nodes do not have to know when broadcasting starts: although the algorithm works in stages, every node that gets the source message knows when to start transmissions and what is the number of its stage.

It should be noted that an algorithm based on procedure Decay from [3] could not be used to obtain optimal performance because, with high probability, it requires time $\Omega(D\log n + \log^2 n)$. On the other hand, trying to shorten the length of procedure Decay would not work either, as nodes with many informed in-neighbors could not be informed fast with high probability. The novelty and strength of our approach consists in simultaneous shortening of stage lengths to $\log(r/D) + 1$ steps, and adding only one extra step to each of them (with a particular corresponding transmission probability) in order to handle uninformed nodes with many informed in-neighbors.

We now define a sequence of probabilities which will be used in the last step of each stage. An infinite sequence $(p_i)_{i=1}^{\infty}$ of reals from the interval $[0, 1]$ is called *universal*, if the following conditions hold:

U1. for every $j = \log(r/D) + 1, \ldots, \lceil\log\frac{r}{4\log r}\rceil$, the sequence $p_{i+1}, p_{i+2}, \ldots, p_{i+3\frac{D\cdot 2^j}{r}}$ contains at least one value $\frac{1}{2^j}$;
U2. for every $j = \lceil\log\frac{r}{4\log r}\rceil + 1, \ldots, \log r$, the sequence $p_{i+1}, p_{i+2}, \ldots, p_{i+3\frac{D\cdot 2^j}{r\cdot 2^{\lceil\log\log r\rceil+1}}}$ contains at least one value $\frac{1}{2^j}$.

Lemma 1 *For sufficiently large r and every D such that $32r^{2/3} < D \le r$, there exists a universal sequence.*

Proof. For every $j = \log(r/D) + 1, \ldots, \lceil\log\frac{r}{4\log r}\rceil$, we attach the real $\frac{1}{2^j}$ to every node in level $\log(2r/2^j)$ of the complete binary tree of depth $\log D$. For every $j = \lceil\log\frac{r}{4\log r}\rceil + 1, \ldots, \log r$, we attach the real $\frac{1}{2^j}$ to every node in level $\log\frac{2r2^{\lceil\log\log r\rceil+1}}{2^j}$ of the complete binary tree of depth $\log D$.

Next, starting with nodes in level $\log D - 1$, we move reals from their current locations to leaves, in the following way. Consider a node v, such that all non-leaf descendants of v have their assigned reals already moved to leaves. Let z be the leftmost leaf in the subtree of v which has fewer reals already moved to it than leaves to the left of it in this subtree (or, let it be the leftmost leaf in the subtree of v if all leaves hold the same number of reals). We move the real from v (or, in the case when v had two reals initially assigned, the smaller of them) to z. The new real is appended to the end of the sequence of reals already moved to z.

The total number of reals distributed in the tree is

$$\sum_{j=\log(r/D)+1}^{\lceil\log\frac{r}{4\log r}\rceil}\frac{2r}{2^j} + \sum_{j=\lceil\log\frac{r}{4\log r}\rceil+1}^{\log r}\frac{2r2^{\lceil\log\log r\rceil+1}}{2^j} \le$$

$$\le 2\frac{2r}{2r/D} + 2\frac{8r\log r}{r/(2\log r)} = 2D + 32\log^2 r < 3D\,.$$

After all moves, these reals are distributed almost evenly among leaves of the tree (the difference between the number of reals in different leaves can be at most 1). Hence there are at most 3 reals in every leaf. Now the sequence $(p_i)_{i=1}^{\infty}$ is constructed in two further steps. First, all sequences of reals in leaves are concatenated from left to right. Then copies of the obtained sequence are infinitely concatenated.

The proof of property U1 of the sequence $(p_i)_{i=1}^{\infty}$ follows from the fact, that the distances between two consecutive equal values $1/2^j$, for $j = \log(r/D) + 1, \ldots, \lceil \log \frac{r}{4\log r} \rceil$, are at most

$$3 \cdot 2 \cdot 2^{\log D - \log(2r/2^j)} = \frac{6D}{2r/2^j} = \frac{3D \cdot 2^j}{r}.$$

The factor 3 is the maximal number of reals in one leaf, the factor 2 is because a real may be moved to the leftmost or to the rightmost leaf, and the factor $2^{\log D - \log(2r/2^j)}$ is the number of leaves in the subtree of a node in layer $\log(2r/2^j)$, where values $1/2^j$ are placed at the beginning of the construction.

The proof of property U2, for $j = \lceil \log \frac{r}{4\log r} \rceil + 1, \ldots, \log r$, is similar to the above - the only difference is replacing $2r/2^j$ by $\frac{2r2^{\lceil \log\log r \rceil + 1}}{2^j}$.

Notice that the above bounds also hold when we count distance modulo the length of the string whose copies are infinitely concatenated. Hence, both U1 and U2 hold after infinite concatenation. □

Fix a universal sequence $(p_i)_{i=1}^{\infty}$. We define the following procedures.

Procedure Stage(D, i)
for $l = 0$ **to** $\log(r/D)$ **do** transmit with probability $\frac{1}{2^l}$
transmit with probability p_i

Procedure Randomized-Broadcasting(D)
if $D \le 32r^{2/3}$ **then** perform Procedure Broadcast from [3]
else
the source transmits
for $i = 1$ **to** $4660D$ **do**
if node v received source message before Stage(D, i)
then v performs Stage(D, i)

If $D \le 32r^{2/3}$ then $\Omega(\log(r/D)) = \Omega(\log r)$ and Procedure Randomized-Broadcasting(D) works in time $\mathcal{O}(D\log(n/D) + \log^2 n)$, in view of [3]. We now analyze Procedure Randomized-Broadcasting(D), assuming $D > 32r^{2/3}$.

Fix a directed graph $G = (V, E)$ with n nodes and radius D. In what follows, v_0 denotes the source. Fix a node $v \in V$ and consider a shortest directed path $v_0, v_1, \ldots, v_k$, where $v_k = v$. Obviously $k \le D$. Let P_v denote the subgraph of G including the path $v_0, v_1, \ldots, v_k$, and all in-neighbors of any v_j, for $j = 1, \ldots, k$. For any such node w, we put the edge (w, v_j) in P_v, for $j = \max\{j' : (w, v_{j'}) \in E\}$. Let d_j, for $j = 1, \ldots, k$, be the in-degree of node v_j in P_v. Note that $\sum_{i=1}^{k} d_i \le n$.

Let $\mathcal{E}_{i,j}$, for $j < k$, be the event that after stage i of Procedure Randomized-Broadcasting(D) all nodes $v_j, \ldots, v_k$ do not have the source message, and all nodes $v_{j+1}, \ldots, v_k$ have no in-neighbor in P_v having the source message but v_j has such an in-neighbor. Denote by $\mathcal{E}_{i,k}$ the event that v_k has the source message at the end of stage i. Our aim is to show that, for some constant γ, $\Pr[\, \mathcal{E}_{\gamma D,k} \,] \ge 1 - 1/r^2$. The proof of this fact is split into a series of lemmas. By definition of $\mathcal{E}_{i,j}$, $\Pr[\, \mathcal{E}_{i+x,j} \mid \mathcal{E}_{i,j} \,]$ is the probability that, during x stages of the procedure, the "information front" in P_v does not move from v_j. Lemmas 2, 3 and 4 estimate this probability for different ranges of in-degree d_j (notice that x varies depending on the range of d_j). Lemma 6 uses these estimates to show that information front reaches v in $\mathcal{O}(D)$ stages, with high probability.

Lemma 2 *For every $j \le k$, if $d_j \le r/D$ then*

$$\Pr[\, \mathcal{E}_{i+1,j} \mid \mathcal{E}_{i,j} \,] < \tfrac{7}{8}\,.$$

Proof. Fix an elementary event from $\mathcal{E}_{i,j}$: an execution of Procedure Randomized-Broadcasting(D) to the end of stage i for a fixed random seed . Suppose that at the beginning of stage $i+1$, node v_j has more than 2^{l-1} and at most 2^l informed in-neighbors, for some $l = 0, \ldots, \lceil \log d_j \rceil$. Consider step l during stage $i+1$ (it exists, since $l \le \lceil \log d_j \rceil \le \log(r/D)$). Since no additional in-neighbor of node v_j can transmit during stage $i+1$ (if such in-neighbor receives the source message during stage $i+1$ for the first time, it will start transmitting in stage $i+2$), there are still more than 2^{l-1} and at most 2^l in-neighbors of v_j having the source message and possibly transmitting, each with probability $1/2^l$. If $l = 0$ then the proof is obvious: the transmission is successful with probability 1 during step $l = 0$ in stage $i+1$. If $l = 1$ then, with probability $2 \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$, the transmission in step $l = 1$ in stage $i+1$ is successful. Assume $l > 1$. The probability that, during step l of stage $i+1$, node v_j receives the source message, is at least

$$x \cdot \frac{1}{2^l}\Big(1 - \frac{1}{2^l}\Big)^{x-1} > 2^{l-1} \cdot \frac{1}{2^l}\Big(1 - \frac{1}{2^l}\Big)^{2^l} \ge \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}\,,$$

where x is the number of in-neighbors of v_j in P_v having the source message at the beginning of stage $i+1$. □

Let $d_j(x)$, for integer $x \ge 0$, be the number of in-neighbors of v_j having the source message after stage $i+x$ if $\mathcal{E}_{i+x,j}$, or equal to $r+1$ otherwise.

Lemma 3 *For every $j \le k$, if the inequality $\log \frac{r}{D} < \log d_j \le \lceil \log \frac{r}{4\log r} \rceil$ holds then*

$$\Pr\Big[\, \mathcal{E}_{i+\frac{6D \cdot 2^{\lceil \log d_j \rceil}}{r},j} \,\Big|\, \mathcal{E}_{i,j} \,\Big] < \frac{7}{8}.$$

Proof. Fix an elementary event from $\mathcal{E}_{i,j}$.

Case 1. $d_j(0) \le r/D$. The proof as in Lemma 2.

Case 2. $r/D < d_j(0) \le d_j$. We prove that for every positive integer m and $\log(r/D) < l \le \lceil \log d_j \rceil$,

$$\Pr\Big[d_j\Big(m + \frac{3D \cdot 2^l}{r}\Big) \le 2^l \,\Big|\, \mathcal{E}_{i,j}, d_j(m) > 2^{l-1} \Big] < \frac{7}{8}\,. \quad (1)$$

In the event $(\mathcal{E}_{i,j}, d_j(m) > 2^{l-1})$, in all stages $i + m + 1, \ldots, i + m + \frac{3D \cdot 2^l}{r}$, we have that the number of in-neighbors of v_j in P_v having the source message is more that 2^{l-1}. From universality condition U1 (for $\log d_j \leq \lceil \log \frac{r}{4\log r} \rceil$), in this interval of stages there is at least one stage i_0 such that $p_{i_0} = 1/2^l$. We have the inequality

$$\mathsf{Pr}\left[\, d_j\Big(m + \frac{3D\cdot 2^l}{r}\Big) \leq 2^l \;\Big|\; \mathcal{E}_{i,j}, d_j(m) > 2^{l-1} \,\right] \leq$$

$$\leq \mathsf{Pr}\left[\, d_j\Big(m + \frac{3D\cdot 2^l}{r}\Big) \leq 2^l \;\Big|\; \mathcal{E}_{i,j}, \mathcal{E}^* \,\right],$$

where $\mathcal{E}^*$ denotes the event:

$$d_j(m) > 2^{l-1} \text{ and } d_j(m + i_0 - 1) \leq 2^l .$$

We can bound the right-hand side of the above inequality by the probability of transmission failure to v_j during stage i_0, under condition that at the beginning of stage i_0 the number of informed in-neighbors of v_j is greater than 2^{l-1} and at most 2^l. This probability is at most $1/8$, by the same argument as in Lemma 2. Hence we proved inequality (1).

Let $\mathcal{A}$ be the event that $d_j(\frac{3D\cdot d_j}{r}) < r+1$. We define events $\mathcal{A}_l(m)$, for $l = \log(r/D)+1, \ldots, \lceil \log d_j \rceil$ and positive integer m, by induction. Fix $\log(r/D)+1 \leq l \leq \lceil \log d_j \rceil$ and a positive integer m. Assume that $\mathcal{A}_{l'}(m')$, for $l' \leq l$ and $m' < m$, are already defined. We define $\mathcal{A}_l(m)$ as the set of elementary events in $\mathcal{A} \setminus \bigcup_{l'\leq l} \bigcup_{m'<m} \mathcal{A}_{l'}(m')$, such that $2^{l-1} < d_j(m+1) \leq \ldots \leq d_j(m + \frac{3D\cdot 2^l}{r}) \leq 2^l$. It follows, that for every l and m, events $\mathcal{A}_l(m)$ are disjoint.

$$\begin{aligned}
&\mathsf{Pr}\left[\, \mathcal{E}_{i+2\frac{3D\cdot 2^{\lceil \log d_j\rceil}}{r},j} \;\Big|\; \mathcal{E}_{i,j} \,\right] = \\
&= \mathsf{Pr}\left[\, \bigcup_{l=\log(r/D)+1}^{\lceil \log d_j\rceil} \bigcup_m \mathcal{A}_l(m) \;\Big|\; \mathcal{E}_{i,j} \,\right] \\
&\leq \sum_{l=\log\frac{r}{D}+1}^{\lceil \log d_j\rceil} \sum_m q_{l,m} \cdot \mathsf{Pr}\left[\, 2^l \geq d_j(m) > 2^{l-1} \;|\; \mathcal{E}_{i,j} \,\right] \\
&\leq \max_{\log(r/D)+1\leq l\leq \lceil \log d_j\rceil} \max_m q'_{l,m} \\
&< \frac{7}{8},
\end{aligned}$$

where

$$q_{l,m} = \mathsf{Pr}\left[d_j\Big(m+\frac{3D\cdot 2^l}{r}\Big) \leq 2^l \Big| \mathcal{E}_{i,j}, 2^l \geq d_j(m) > 2^{l-1} \right]$$

$$q'_{l,m} = \mathsf{Pr}\left[\, d_j\Big(m+\frac{3D\cdot 2^l}{r}\Big) \leq 2^l \;\Big|\; \mathcal{E}_{i,j}, d_j(m) > 2^{l-1} \right].$$

□

Lemma 4 *For every $j \leq k$, if $\lceil \log \frac{r}{4\log r} \rceil < \log d_j \leq \log r$ then*

$$\mathsf{Pr}\left[\mathcal{E}_{i+\frac{6D\cdot 2^{\lceil \log d_j\rceil}}{r\cdot 2^{\lceil \log\log r\rceil+1}},j} \Big| \mathcal{E}_{i,j} \right] < \frac{7}{8}.$$

Proof. Fix an elementary event from $\mathcal{E}_{i,j}$.

Case 1. $d_j(0) \leq r/D$. The proof is similar to that in Lemma 2.

Case 2. $r/D < d_j(0) \leq d_j$. First we prove that for every $l > \lceil \log \frac{r}{4\log r} \rceil$ and positive integer m,

$$\mathsf{Pr}\left[d_j\Big(m+\frac{3D\cdot 2^l}{r\cdot 2^{\lceil \log\log r\rceil+1}}\Big) \leq 2^l \Big| \mathcal{E}_{i,j}, d_j(m) > 2^{l-1} \right] < \frac{7}{8}.$$

The proof is similar to that of inequality (1) in Lemma 3: we replace $\frac{3D\cdot 2^l}{r}$ by $\frac{3D\cdot 2^l}{r\cdot 2^{\lceil \log\log r\rceil+1}}$ and use universality condition U2 (for $l > \lceil \log \frac{r}{4\log r} \rceil$), instead of U1.

Let $\mathcal{A}$ be the event such that $d_j(\frac{3D\cdot 2^l}{r\cdot 2^{\lceil \log\log r\rceil+1}}) < r+1$. We define events $\mathcal{A}_l(m)$, for $l = \log(r/D)+1, \ldots, \lceil \log d_j \rceil$ and positive integer m, by induction. Fix $\log(r/D)+1 \leq l \leq \lceil \log d_j \rceil$ and a positive integer m. Assume that $\mathcal{A}_{l'}(m')$, for $l' \leq l$ and $m' < m$, are already defined. We define $\mathcal{A}_l(m)$ as the set of elementary events in $\mathcal{A} \setminus \bigcup_{l'\leq l} \bigcup_{m'<m} \mathcal{A}_{l'}(m')$, such that

$$2^{l-1} < d_j(m+1) \leq \ldots \leq d_j\Big(m + \frac{3D\cdot 2^l}{r}\Big) \leq 2^l ,$$

if $\log(r/D) < l \leq \lceil \log \frac{r}{4\log r} \rceil$, and

$$2^{l-1} < d_j(m+1) \leq \ldots \leq d_j\Big(m + \frac{3D\cdot 2^l}{r\cdot 2^{\lceil \log\log r\rceil+1}}\Big) \leq 2^l$$

otherwise. It follows, that for every l and m, events $\mathcal{A}_l(m)$ are disjoint. Similarly as in Lemma 3 we obtain, that the probability

$$\mathsf{Pr}\left[\, \mathcal{E}_{i+2\frac{3D\cdot 2^{\lceil \log d_j\rceil}}{r\cdot 2^{\lceil \log\log r\rceil+1}},j} \;\Big|\; \mathcal{E}_{i,j} \,\right]$$

is at most

$$\begin{aligned}
&\mathsf{Pr}\left[\, \bigcup_{l=\log(r/D)+1}^{\lceil \log d_j\rceil} \bigcup_m \mathcal{A}_l(m) \;\Big|\; \mathcal{E}_{i,j} \,\right] \\
&\leq \max_m \left\{ \max_{\log(r/D)<l\leq \lceil \log \frac{r}{4\log r}\rceil} q_l, \max_{\lceil \log \frac{r}{4\log r}\rceil < l \leq \lceil \log d_j\rceil} q'_l \right\} \\
&< \frac{7}{8},
\end{aligned}$$

where

$$q_l = \mathsf{Pr}\left[\, d_j\Big(m + \frac{3D\cdot 2^l}{r}\Big) \leq 2^l \;\Big|\; \mathcal{E}_{i,j}, d_j(m) > 2^{l-1} \,\right]$$

$$q'_l = \mathsf{Pr}\left[d_j\Big(m+\frac{3D\cdot 2^l}{r2^{\lceil \log\log r\rceil+1}}\Big) \leq 2^l \;\Big|\; \mathcal{E}_{i,j}, d_j(m) > 2^{l-1} \right].$$

□

Lemma 5 *For sufficiently large $p \geq x$ we have*

$$\sum_{m=48p+1}^{\infty} \binom{m+x}{x} \cdot \Big(\frac{7}{8}\Big)^m \leq (0.4)^p .$$

Proof. Using inequalities $\frac{x^x}{e^x} \le x! \le \frac{x^{x+1}}{e^x}$, for $x \ge 2$, we obtain

$$\sum_{m=48p+1}^{\infty} \binom{m+x}{x} \cdot \left(\frac{7}{8}\right)^m \le$$
$$\le \sum_{m=48p+1}^{\infty} \frac{\frac{(m+x)^{m+x+1}}{e^{m+x}}}{\frac{x^x}{e^x}\cdot\frac{m^m}{e^m}} \cdot \left(\frac{7}{8}\right)^m$$
$$= \sum_{m=48p+1}^{\infty} (m+x) \cdot \left(\frac{m+x}{x}\right)^x \cdot \left[\frac{7}{8}\cdot\left(1+\frac{x}{m}\right)\right]^m$$
$$\le \sum_{m=48p+1}^{\infty} (m+x) \cdot \left(\frac{m+x}{x}\right)^x \cdot (0.89)^m .$$

Using the inequality $\left(\frac{m+x}{x}\right)^x \le (1.1)^m$, for $m \ge 48p \ge 48x$, we finally obtain

$$\sum_{m=48p+1}^{\infty} \binom{m+x}{x} \cdot \left(\frac{7}{8}\right)^m \le$$
$$\le \sum_{m=48p+1}^{\infty} (m+x) \cdot (1.1)^m \cdot (0.89)^m$$
$$\le (0.98)^{48p} \cdot (48p+1) \cdot \sum_{m=0}^{\infty} m \cdot (0.98)^m$$
$$\le (0.4)^p ,$$

for sufficiently large p. □

Lemma 6 $\mathsf{Pr}[\,\mathcal{E}_{4660D,k}\,] \ge 1 - \frac{1}{r^2}$.

Proof. Let a be a sequence of nodes $(v_{j_1}, \ldots, v_{j_{4660D}})$, where $(j_i)_{i\le 4660D}$, $j_i \le k$, is a non-decreasing sequence of indices. Let $\mathcal{E}(a) = \bigcap_{i=1}^{4660D} \mathcal{E}_{i,j_i}$. Let b be a sequence $(v_{t_1}, \ldots, v_{t_u})$, where $(t_i)_{i\le u}$, $t_i \le k$, is a strictly increasing sequence of indices. If after deleting all repetitions from a we obtain a sequence b, we denote this situation by $a \sqsubset b$. Let $x_\ell = |\{v_j : \exists_i\, v_{j_i} = v_j \wedge \lceil \log d_j \rceil = \ell\}|$ and let $n_j = |\{i : v_{j_i} = v_j\}|$.

Fix a sequence b as above. Define the following events:

$$\mathcal{C}_{\mathrm{small}} = \left(\left|\left\{i : \mathcal{E}_{i,j_i} \wedge d_{j_i} \le \frac{r}{D}\right\}\right| > 49D\right),$$

$$\mathcal{C}_\ell = \left(\left|\left\{i : \mathcal{E}_{i,j_i} \wedge \lceil \log d_{j_i} \rceil = \ell\right\}\right| > 48(x_\ell + 2\log r)\frac{6D2^\ell}{r}\right),$$

for every integer $\log(r/D) < \ell \le \lceil \log \frac{r}{4\log r} \rceil$,

$$\mathcal{C}_\ell = \left(\left|\left\{i : \mathcal{E}_{i,j_i} \wedge \lceil \log d_{j_i} \rceil = \ell\right\}\right| > 48(x_\ell + 2\log r)\frac{6D2^\ell}{2r\log r}\right),$$

for every integer $\lceil \log \frac{r}{4\log r} \rceil < \ell \le \log r$.

We have

$$\mathsf{Pr}[\,\neg\mathcal{E}_{4660D,k} \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,] \le$$
$$\le \mathsf{Pr}[\,\mathcal{C}_{small} \cup \bigcup_{\ell=\log(r/D)}^{\log r} \mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,] \tag{2}$$
$$\le \mathsf{Pr}[\,\mathcal{C}_{small} \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,] +$$
$$+ \sum_{\ell=\log(r/D)}^{\log r} \mathsf{Pr}[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,] .$$

Indeed, we argue that inequality (2) holds, since the other one is Boole inequality. Suppose, to the contrary, that $\neg\mathcal{E}_{4660D,k}$ and $\neg(\mathcal{C}_{small} \cup \bigcup_{\ell=\log(r/D)}^{\log r} \mathcal{C}_\ell)$. Note that since $v_0, \ldots, v_k$ is the shortest directed path from the source to node v, and the inequality $\sum_{i=j}^{k} d_j \le n$ holds for P_v, we have

$$49D + \sum_{\ell=\log(r/D)+1}^{\lceil \log \frac{r}{4\log r}\rceil} 48(x_\ell + 2\log r)\cdot\frac{6D2^\ell}{r} +$$
$$+ \sum_{\ell=\lceil \log \frac{r}{4\log r}\rceil+1}^{\log r} 48(x_\ell + 2\log r)\cdot\frac{6D2^l}{2r\log r}$$
$$\le 49D + 48\frac{6D}{r}\sum_{\ell=\log(r/D)+1}^{\log r} x_\ell \cdot 2^\ell +$$
$$+ 48(2\log r)\cdot\frac{6D}{r}\cdot\frac{r}{\log r} + 48(2\log r)\frac{6D}{2r\log r}\cdot 2r$$
$$\le 49D + 48\cdot 3\cdot 12D$$
$$< 4660D ,$$

which is a contradiction with $\neg\mathcal{E}_{4660D,k}$. Hence the inequality (2) has been proved.

We now separately estimate $\mathsf{Pr}[\,\mathcal{C}_{small} \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,]$, and and $\mathsf{Pr}[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,]$ for two disjoint ranges of integer ℓ: $\log(r/D) < \ell \le \lceil \log\frac{r}{4\log r}\rceil$ and $\lceil \log\frac{r}{4\log r}\rceil < \ell \le \log r$.

Estimation of $\mathsf{Pr}[\,\mathcal{C}_{small} \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,]$**.**

Let $x_{small} = \sum_{l\le\log(r/D)} x_l \le D$. Using Lemma 2 and the fact that, for different nodes v_j, we consider disjoint parts of the computation, we have that the probability

$$\mathsf{Pr}\left[\,\mathcal{C}_{small} \,\middle|\, \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,\right] =$$

$$= \mathsf{Pr}\left[\,\left|\left\{v_{j_i} : \mathcal{E}_{i,j_i} \wedge d_{j_i} \le \frac{r}{D}\right\}\right| > 48D + D \,\middle|\, \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,\right]$$

is at most

$$\sum_{x_{small}=1}^{D}\sum_{m=48D+1}^{\infty}\binom{m+x_{small}-1}{x_{small}-1}\cdot\left(\frac{7}{8}\right)^m \le$$
$$\le 8\cdot\left(\frac{7}{8}\right)^{48D+1} + 48D\cdot 8\cdot\left(\frac{7}{8}\right)^{48D} +$$
$$+ \sum_{x_{small}=3}^{D}\sum_{m=48D+1}^{\infty}\binom{m+x}{x}\cdot\left(\frac{7}{8}\right)^m .$$

Using Lemma 5 for $p = D$ and $x = x_{small}$ we get

$$\Pr\Big[\,\mathcal{C}_{small}\,\Big|\,\bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,\Big] \le (0.9)^{48D} + \sum_{x_{small}=3}^{D}(0.4)^{D}$$
$$\le (0.9)^{48D} + D\cdot(0.4)^{D}$$
$$\le (0.45)^{D}\ ,$$

for sufficiently large r. (Since we assumed that $D > r^{2/3}$ in our analysis, and r is sufficiently large, so D is also sufficiently large and we may correctly apply Lemma 5 in the above derivation.) Since $D > 32r^{2/3}$ we obtain $(0.45)^D \le \frac{1}{3r^2}$, for sufficiently large r.

Estimation of $\Pr[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,]$**, for** $\log(r/D) < \ell \le \lceil\log\frac{r}{4\log r}\rceil$**.**

Consider ℓ such that $\log(r/D) < \ell \le \lceil\log\frac{r}{4\log r}\rceil$. Applying Lemma 3 and the fact that, for different nodes v_j in the directed path, we consider disjoint parts of the computation, we have that

$$\Pr[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,] \le$$
$$\le \sum_{m=48(x_\ell+2\log r)+1}^{\infty}\binom{m+x_\ell-1}{x_\ell-1}\cdot\Big(\frac{7}{8}\Big)^{m}\ .$$

Using Lemma 5 we obtain

$$\Pr[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,] \ \le\ (0.4)^{x_\ell+2\log r}\ \le\ \frac{1}{2r^2\log r}$$

for sufficiently large r.

Estimation of $\Pr[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,]$**, for** $\lceil\log\frac{r}{4\log r}\rceil < \ell \le \log r$**.**

Consider ℓ such that $\lceil\log\frac{r}{4\log r}\rceil < \ell \le \log r$. Applying Lemma 4 and the fact that, for different nodes v_j in the directed path, we consider disjoint parts of the computation, we have that

$$\Pr[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,] \le$$
$$\le \sum_{m=48(x_\ell+2\log r)+1}^{\infty}\binom{m+x_\ell-1}{x_\ell-1}\cdot\Big(\frac{7}{8}\Big)^{m}\ .$$

Using Lemma 5 we obtain that

$$\Pr[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,] \ \le\ (0.4)^{x_\ell+2\log r}\ \le\ \frac{1}{2r^2\log r}$$

for sufficiently large r.

Finally we get that for sufficiently large r

$$\Pr[\,\neg\mathcal{E}_{4660\cdot D,k} \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,] \le$$
$$\le \Pr[\,\mathcal{C}_{small} \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,] +$$
$$+ \sum_{\ell=\log(r/D)+1}^{\log r}\Pr[\,\mathcal{C}_\ell \mid \bigcup_{a:a\sqsubset b}\mathcal{E}(a)\,]$$
$$\le \frac{1}{2r^2} + \log D\cdot\frac{1}{2r^2\log r}\ ,$$

which is at most $\frac{1}{r^2}$. Since for different sequences b the conditions $\bigcup_{a:a\sqsubset b}\mathcal{E}(a)$ are disjoint, we obtain

$$\Pr[\,\mathcal{E}_{4660\cdot D,k}\,] \ge 1 - \frac{1}{r^2}\ ,$$

for sufficiently large r. $\square$

Procedure Randomized-Broadcasting(D) used the following extra assumptions: knowledge of D and the assumption that r and D are powers of 2. Our final algorithm is formulated as follows. Use the upper bound $2^{\lceil\log r\rceil}$ instead of r. This does not change complexity and permits to use the assumption about r. The assumptions about D are eliminated using the "doubling technique", which probes possible values of D in exponentially increasing jumps.

Algorithm Optimal-Randomized-Broadcasting

for $i := 1$ **to** $\log r$ **do**

Procedure Randomized-Broadcasting(2^i)

Theorem 1 *Algorithm Optimal-Randomized-Broadcasting performs broadcasting on any n-node network of radius D in time $\mathcal{O}(D\log(n/D) + \log^2 n)$, with probability at least $1-\frac{1}{r}$, for sufficiently large r.*

Proof. Consider an n-node directed graph G of radius D. If $D \le 32r^{2/3}$, the result follows from [3]. Suppose that $D > 32r^{2/3}$ and let $i = \lceil\log D\rceil$. By Lemma 6, the probability that a fixed node v of G does not receive the source message by stage $4660\cdot 2^i$ of Procedure Randomized-Broadcasting(2^i) is at most $1/(r^2)$. Hence the probability that some node does not receive the source message by this stage is at most $r\cdot\frac{1}{r^2} = \frac{1}{r}$. For any $j \le i$ the execution of Procedure Randomized-Broadcasting(2^j) takes time $\mathcal{O}(2^j\log(n/2^j)+\log^2 n)$. (recall that r is linear in n). Hence the total time until Algorithm Optimal-Randomized-Broadcasting informs all nodes of the graph, with probability at least $1-1/r$, is $\mathcal{O}(2^i\log(n/2^i)) = \mathcal{O}(D\log(n/D))$. $\square$

The following corollary is a straightforward consequence of Theorem 1 (by iterating Algorithm Optimal-Randomized-Broadcasting).

Corollary 1 *There exists a randomized algorithm which performs broadcasting on any n-node network of radius D in expected time $\mathcal{O}(D\log(n/D) + \log^2 n)$.*

3 The lower bound for deterministic broadcasting

Our main result for deterministic broadcasting is the lower bound $\Omega(n\frac{\log n}{\log(n/D)})$ on broadcasting time in n-node radio networks of radius D. We first present an intuitive sketch of the argument. Given a broadcasting algorithm $\mathcal{A}$, we construct, step-by-step, a network on which this algorithm works slowly. We use consecutive steps of algorithm $\mathcal{A}$, and *assume* that particular nodes got particular messages in given steps. In order to express this, we use the notion of *abstract history* of a node, formally defined below. Intuitively, an abstract history of a node v at a given step k consists of a sequence of messages received by this node until step k. Since the network is not yet constructed, neighborhoods of some nodes are not determined by step k, and consequently it is not yet known which abstract history will become the real one — the one given by algorithm $\mathcal{A}$ running on the final network. We can ensure that, if a given node had some history of received messages up to a certain step, then it would behave in a given way. Based on that we do the next step of the construction of the network, and simultaneously define abstract histories of nodes in this step. These abstract histories are defined so as to prevent nodes in consecutive layers of the network from getting any message for a long time. Layers are of size $\Theta(n/D)$, and we are able to prevent transmissions from layer to layer for time $\Theta(\frac{n}{D} \cdot \frac{\log n}{\log(n/D)})$. This is done using properties of function JAMMING (defined later), and also a lower bound on the size of selective families from [10]. When the construction is finished, we prove that if the algorithm $\mathcal{A}$ runs on the resulting network then the real histories of all nodes are identical to the abstract (assumed) ones, and consequently, nodes of the last layer will indeed fail to receive the source message for $\Omega(n\frac{\log n}{\log(n/D)})$ steps.

3.1 Construction

Fix a deterministic broadcasting algorithm $\mathcal{A}$. For this algorithm, running on any network $G = (V, E)$, we define the following objects.

Histories and message format. H_k denotes the history of computation of algorithm $\mathcal{A}$ until the end of step k. This is the set $\{H_k(v) : v \in V\}$, where $H_k(v)$ is the history of computation at node v, until the end of step k. For any v and k, $H_k(v)$ is a sequence of *messages* $(M_0(v), M_1(v), \ldots, M_k(v))$. Messages are defined inductively, as follows. $M_0(v)$ is either the pair $(\emptyset, \emptyset)$, called the *empty message*, or the pair $(0, source\_message)$. $M_l(v)$ (for $l = 1, \ldots, k$) is the empty message if node v did not get any message in step l. Otherwise, it is a pair consisting of:

- the label of node w from which node v received a message in step l,
- history $H_{l-1}(w)$.

Notice that we restrict attention to messages conveying the entire history of the transmitter. If a particular protocol requires transmitting specific information, the receiver can deduce this information from the received history, since programs of all nodes are the same. History $H_k(v)$ containing only empty messages is called the *empty history*.

Action function and sets of transmitters. Given algorithm $\mathcal{A}$, we denote by $\pi(v, H_{k-1}(v))$ the action of node v in step k, if its history until the end of step $k-1$ is $H_{k-1}(v)$. The values of the function π can be 1 or 0: if the value is 1, node v is sending the message $(v, H_{k-1}(v))$ in step k, otherwise it is receiving in step k. Since spontaneous transmissions are not allowed, we assume that $\pi(v, H_{k-1}(v)) = 0$, if $v \neq 0$ and $H_{k-1}(v)$ is the empty history. Under a fixed history H_{k-1}, we define the set of neighbors of v transmitting in step k as follows: $T_k(v) = \{w \in N_v : \pi(w, H_{k-1}(w)) = 1\}$, where N_v denotes a set of all neighbors of node v.

Abstract objects. Let $v \in V$. An abstract history $\hat{H}_k(v)$ of node v, is defined as a sequence $(\hat{M}_0(v), \ldots, \hat{M}_k(v))$ of *abstract messages*. $\hat{M}_0(v) = M_0(v)$, and $\hat{M}_l(v)$, for $l > 0$, is either the empty message or is of the format $(w, \hat{H}_{l-1}(w))$, for some $w \in V$. Notice that, in general, abstract histories and abstract messages are not necessarily linked to any particular protocol.

We also define the notion of the *abstract action function* $\hat{\pi}(v, \hat{H}_{k-1}(v))$ as an extension of the action function π described above: if $\pi(v, \hat{H}_{k-1}(v))$ is defined for some v then $\hat{\pi}(v, \hat{H}_{k-1}(v)) = \pi(v, \hat{H}_{k-1}(v))$. Otherwise, $\hat{\pi}(v, \hat{H}_{k-1}(v)) = 0$. We now define sets of *abstract transmitters* to node v in step k by the formula: $\hat{T}_k(v) = \{w \in N_v : \hat{\pi}(w, \hat{H}_{k-1}(w)) = 1\}$.

We finally define the procedure RADIO. It is used to obtain an abstract message $\hat{M}_j(v)$ in step j, provided that a neighborhood N_v of node v is constructed by step $j-1$, and provided that, for all $w \in N_v$, the abstract histories $\hat{H}_{j-1}(w)$ are already defined. This procedure corresponds to the natural way of receiving messages in radio networks, and hence of forming histories. It is more general in that it concerns arbitrary abstract objects (more precisely, abstract action functions and abstract transmitters) which were defined above. The parameters of the procedure are the step j and the receiving node v. It defines the abstract message $\hat{M}_j(v)$ depending on the value of the abstract function $\hat{\pi}(v, \hat{H}_{j-1}(v))$ and of the set of abstract transmitters $\hat{T}_j(v)$.

Procedure RADIO(j, v)
 if $\hat{\pi}(v, \hat{H}_{j-1}(v)) = 1$
  then $\hat{M}_j(v)$ is the empty message
 if $\hat{\pi}(v, \hat{H}_{j-1}(v)) = 0$ and $|\hat{T}_j(v)| \neq 1$
  then $\hat{M}_j(v)$ is the empty message
 if $\hat{\pi}(v, \hat{H}_{j-1}(v)) = 0$ and $\hat{T}_j(v) = \{w\}$
  then $\hat{M}_j(v) := (w, \hat{H}_{l-1}(w))$

The following result establishes the lower bound of $\Omega(\frac{n \log n}{\log(n/D)})$ on broadcasting time in n-node networks of radius D.

Theorem 2 *For all parameters n and $D \leq n$, and for every broadcasting algorithm $\mathcal{A}$, there is an n-node network $G_{\mathcal{A}}$ of radius $\Theta(D)$, such that algorithm $\mathcal{A}$ requires time $\Omega(n\frac{\log n}{\log(n/D)})$ to complete broadcasting on $G_{\mathcal{A}}$.*

In order to prove the theorem, fix a broadcasting algorithm $\mathcal{A}$, and parameters n, D. We construct the network $G_{\mathcal{A}}$ on which $\mathcal{A}$ requires time $\Omega(\frac{n \log n}{\log(n/D)})$ to broadcast, proceeding layer by layer. The construction and the major part of the analysis is carried out under the assumption that $\sqrt[4]{n^3} < D \leq \frac{n}{16}$. At the end we show how to handle other values of D. We may assume that D is even. Otherwise we perform the construction for $n-1$ nodes and even diameter $D-1$, and then we add the nth node as a neighbor of the only node in the last, already constructed, $(D-1)$th layer.

Fix $L_{2i} = \{i\}$ for $i = 0, 1, \ldots, \frac{D}{2} - 1$. For every $i = 0, 1, \ldots, \frac{D}{2} - 1$ we assume that layer L_{2i+1} is partitioned

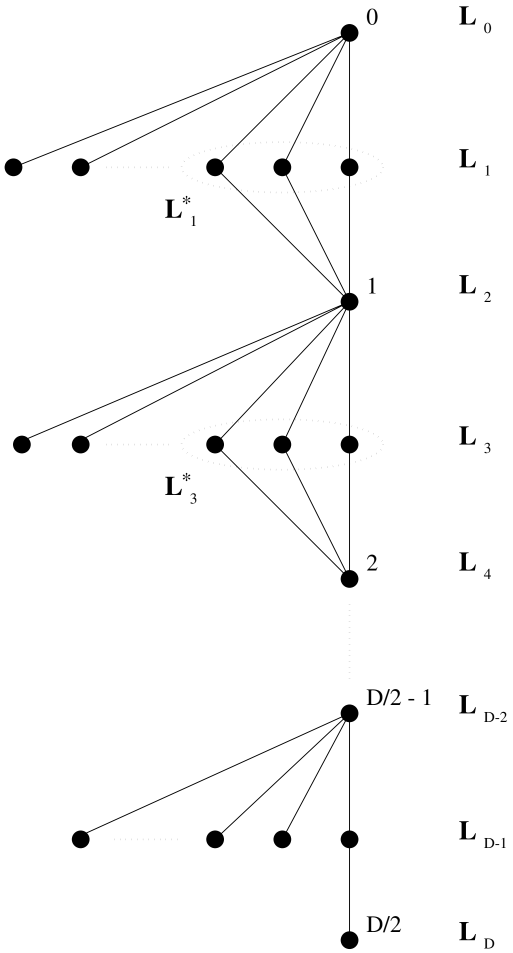

Fig. 1. Network $G_{\mathcal{A}}$ used to prove the lower bound for deterministic broadcasting

into nonempty sets L'_{2i+1} and L^*_{2i+1}, each of size at most $k = \lfloor \frac{n}{4D} \rfloor$. Notice that $4 \le k < \sqrt[4]{n/4}$. Edges in graph $G_{\mathcal{A}}$ are defined to be between the only $v \in L_{2i}$ and every $w \in L_{2i+1}$, and between every $w \in L^*_{2i+1}$ and the only $v \in L_{2(i+1)}$, for $i = 0, \ldots, \frac{D}{2} - 2$. Layer L_D contains all nodes outside of any previous layer, all of them attached to all nodes of L^*_{D-1} (see Fig. 1). We denote the family of all such networks by $C_{n,D}$.

The construction proceeds in stages. During stage $0 \le i \le \frac{D}{2} - 1$, layer L_{2i+1} is constructed. Let $R_{i+1} = \{1, \ldots, n\} \setminus \bigcup_{j=1}^{i}(L_{2j} \cup L_{2j-1})$. We assume that stage $0 \le i < \frac{D}{2} - 1$ ends in step t_i, and the following stage-invariant holds:

0. $\hat{\pi}(i, \hat{H}_{t_i}(i)) = 1$ and $\hat{\pi}(i, \hat{H}_t(i)) = 0$, for all $t < t_i$.
1. All layers up to L_{2i} are defined, sets L'_{2j+1} and L^*_{2j+1} are defined and are of size at most k each, for every $j < i$, and abstract histories $\hat{H}_{t_i}(v)$ are defined for all nodes v.
2. For every $v \notin \{0\} \cup \bigcup_{j=1}^{i}(L_{2j} \cup L_{2j-1})$ the abstract history $\hat{H}_{t_i}(v)$ is empty.

Jamming function. Assume that stage-invariant holds for stage i. Before describing stage $i+1$ of the construction, we define the function $(i+1)$-JAMMING, which is its main combinatorial ingredient. This function is used to construct the next layer of odd number so as to prevent fast transmission to this layer.

Denote $|R_{i+1}|$ by m. Notice that $m > n/4$, by stage-invariant for stage i. For simplicity, we assume that k is even and that it divides $2m$. In other cases the proof is easy to modify.

Let $\{B(p)\}_{p=1}^{k/2}$ be a fixed partition of R_{i+1} into $k/2$ blocks of size $|B(p)| = 2m/k$. Denote by $B_l(p)$ the pth block after step l of the construction, $B_0(p) = B(p)$. We will construct a set X such that, after every step l of the construction, $|B_l(p) \cap X| = 2$ and $B_l(p) \subseteq B_{l-1}(p)$, for every $p \le k/2$. Sets $B_l(p)$ have the important property that it is impossible to tell which of their elements are connected to the previous layer and which are not. The set X will become layer L_{2i+1}, currently under construction, depending on sets $B_l(p)$ which shrink as the algorithm progresses.

Let $A_l = \{p \le k/2 : |B_l(p)| \ge k\}$. For $1 \le l \le \frac{k \log m}{8 \log k} - 1$ and for a set Y_l, we define the function $(i+1)$-JAMMING$_l(Y_l)$ which returns either the number 0 (no node from Y_l transmits) , or a node v (v is the only transmitting node from Y_l), or the symbol $\perp$ (at least two nodes from Y_l transmit).

We first describe the intuitive goal of executing function $(i+1)$-JAMMING. Suppose that the network $G_{\mathcal{A}}$ is defined up to layer $L_{2i} = \{i\}$ and node i transmitted the source message for the first time.

We want to construct:

i) layer $L_{2i+1} \subseteq R_{i+1}$ so that node i will not receive a message from a single node in L_{2i+1} fast (in fewer than $\lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$ steps), and also
ii) a subset L^*_{2i+1} of L_{2i+1}, such that no single node from this subset will transmit fast.

To do so, function $(i+1)$-JAMMING$_l(Y_l)$ computes an approximate worst possible "answer" for node i during the lth step, where set $Y_l \subseteq R_{i+1}$ is a set of potential transmitters during step l of stage $i+1$. The computation uses sets $B_l(p)$. This function also modifies sets $B_l(p)$ in a way to preserve the property that every set X having at least two common elements with every $B_l(p)$ would produce the same answers if $L_{2i+1} = X$. Additionally we require the existence of a block $B_l(p^*)$ of size at least $k\sqrt[4]{m}$ to have the possibility of choosing many suitable sets X such that $|X \cap B_l(p^*)| \ge 2$. This property will imply that we can choose a subset L^*_{2i+1} of L_{2i+1} such that no single node in this subset will transmit fast, and hence node $i+1$ will not receive the source message fast.

Function $(i+1)$-JAMMING$_l(Y_l)$

1. For all $p \le k/2$ set $B_l(p) := B_{l-1}(p)$.
2. We modify sets $B_l(p)$ and define $(i+1)$-JAMMING$_l(Y_l)$ as follows:
   A. If there is $p_0 \in A_{l-1}$ such that $|B_{l-1}(p_0) \cap Y_l| > \frac{2}{k} \cdot |B_{l-1}(p_0)|$, then $(i+1)$-JAMMING$_l(Y_l) :=\perp$ and $B_l(p_0) := B_{l-1}(p_0) \cap Y_l$. (Notice that $|B_l(p_0)| \ge 2$.) If $|B_l(p_0)| < k$ then we choose two elements $v, w \in B_l(p_0)$ and set $B_l(p_0) := \{v, w\}$.
   B. If, for every $p \in A_{l-1}$, the inequality $|B_{l-1}(p) \cap Y_l| \le \frac{2}{k} \cdot |B_{l-1}(p)|$ holds, then we set $B_l(p) := B_{l-1}(p) \setminus Y_l$ for every $p \in A_{l-1}$. For every $p \in A_{l-1}$ such that

$|B_l(p)| < k$, we choose two elements $v, w \in B_l(p)$ and set $B_l(p) := \{v, w\}$. Then
(a) **if** $|Y_l \cap \bigcup_{p \notin A_l} B_l(p)| = 0$
then $(i+1)$-JAMMING$_l(Y_l) := 0$,
(b) **if** $Y_l \cap \bigcup_{p \notin A_l} B_l(p) = \{v\}$
then $(i+1)$-JAMMING$_l(Y_l) := v$,
(c) **if** $|Y_l \cap \bigcup_{p \notin A_l} B_l(p)| > 1$
then $(i+1)$-JAMMING$_l(Y_l) :=\perp$.

Using function $(i+1)$-JAMMING, we can formally define stage $i+1$ of the construction (see Fig. 2). Below we describe its intuitive meaning. In part 2, we fix an abstract history based on the results of function $(i+1)$-JAMMING applied to the set of abstract transmitters. We will preserve invariant INV (to be defined later) for a given l. In part 3, assuming that INV holds for the last value of l in the loop of part 2, we define sets L^*_{2i+1} and L'_{2i+1}. In part 4, we continue defining the abstract history until node $i+1$ (the unique node of layer L_{2i+2}) transmits for the first time. In part 5, we fix the time of the first transmission to layer $L_{2(i+1)+1}$. In part 6, we fix the empty abstract history at all nodes outside of already constructed layers $L_0, ..., L_{2i+2}$.

3.2 Analysis

We first show that stage $i+1$ of the construction is correct, assuming that stage-invariant holds after stage i. To do this, we define an invariant INV after each iteration of the loop in part 2 of stage $i+1$ of the construction. In Lemma 7 we show that invariant INV holds. This implies that the assumptions of part 3 of stage $i+1$ of the construction are satisfied.

Suppose $l \le \lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$. We say that set X models $(i+1)$-JAMMING$_l(Y_l)$, and we denote this fact by $X \models (i+1)$-JAMMING$_l(Y_l)$, if:

$X \cap Y_l = \emptyset$ when $(i+1)$-JAMMING$_l(Y_l) = 0$,
$X \cap Y_l = \{v\}$ when $(i+1)$-JAMMING$_l(Y_l) = v$, and
$|X \cap Y_l| \ge 2$ when $(i+1)$-JAMMING$_l(Y_l) =\perp$.

We define the following invariant INV after step $l \le \lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$ of stage $i+1$ of the construction (by "step" we mean an iteration of the loop in part 2 of this stage, for a given l):

0. Sets $B_{l'}(p)$ and values $(i+1)$-JAMMING$_{l'}(Y_{l'})$, for $l' \le l$ and $p \le k/2$, are defined; moreover $|B_l(p)| \ge k$ for all $p \in A_l$, and $|B_l(p)| = 2$ for $p \notin A_l$.
1. $B_l(p) \subseteq B_{l-1}(p)$, for all $p \le k/2$.
2. There exists $p \le k/2$ such that $|B_l(p)| \ge k\sqrt[4]{m}$.
3. Let $p^* \le k/2$ be any integer such that $|B_l(p^*)| \ge k\sqrt[4]{m}$. Let X' be any set of size $k-2 > 0$ such that $|X' \cap B_l(p)| = 2$ for every $p \ne p^*$. Then, for every nonempty set $X^* \subseteq B_l(p^*)$ of size at most k, we have $(X' \cup X^*) \models (i+1)$-JAMMING$_{l'}(Y_{l'})$, for all $l' \le l$.

Lemma 7 *The invariant INV holds after every step $l \le \lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$ of stage $i+1$ of the construction.*

Proof. The proof is by induction on l. For $l = 1$ it is obvious, by definition of $B_0(p)$ and the bounds on k. Suppose that after step $l < \lfloor \frac{k \log(n/4)}{8 \log k} \rfloor \le \frac{k \log m}{8 \log k}$ the invariant is satisfied. We show it for $l+1$.

Point 0 of INV holds in view of the invariant after previous steps, and because all sets which were decreased below k in function $(i+1)$-JAMMING are modified to contain exactly two elements. Notice that all sets $B_{l+1}(p)$, which have size at least k at the beginning of step $l+1$ of the construction, can be subsequently decreased by a factor of at most $2/k$, hence they are of size at least 2. Additionally, if the size is below k, they are again decreased to have size 2.

Point 1 of INV is straightforward.

We now prove point 2 of INV after step $l+1$. Let $l_A(p) \le l+1$ be the number of steps in which point 2.A was applied for $p_0 = p$ during the execution of function $(i+1)$-JAMMING. Let $l_B \le l+1$ be the number of steps in which point 2.B was applied during the execution of function $(i+1)$-JAMMING. Clearly, $\sum_{p=1}^{k/2} l_A(p) \le \frac{k \log m}{8 \log k}$ and $l_B \le \frac{k \log m}{8 \log k}$. Hence there is p^* such that $l_A(p^*) \le \frac{\log m}{4 \log k}$. This implies

$$\begin{aligned}
|B_{l+1}(p^*)| &\ge \left(\frac{2}{k}\right)^{l_A(p^*)} \cdot \left(1 - \frac{2}{k}\right)^{l_B} \cdot |B(p^*)| \\
&\ge \left(\frac{1}{k}\right)^{\frac{\log m}{4 \log k}} \cdot \left(1 - \frac{2}{k}\right)^{\frac{k \log m}{8 \log k}} \cdot \frac{2m}{k} \\
&\ge 2^{-\frac{\log m}{4} - 2\frac{\log m}{4 \log k} + \log m - \log k} \\
&\ge 2^{\log \sqrt{m} - \log k},
\end{aligned}$$

which is greater than $k \cdot \sqrt[4]{m} = 2^{\log k + \frac{1}{2} \log \sqrt{m}}$ since $\log m > 4 \log k \ge 8$. This proves point 2 of INV after step $l+1$.

We finally prove point 3 of INV after step $l+1$. Let $p^* \le k/2$ be any integer such that $|B_l(p^*)| \ge k\sqrt[4]{m}$. Let X' be any set such that $|X'| = k-2 > 0$ and for every $p \ne p^*$, $|X' \cap B_{l+1}(p)| = 2$. Let X^* be any nonempty subset of $B_{l+1}(p^*)$ of size at most k. We show that $(X' \cup X^*) \models (i+1)$-JAMMING$_{l'}(Y_{l'})$, for $l' \le l+1$. Observe that $B_{l+1}(p) \subseteq X'$ for every $p \notin A_{l+1}$, and $X' \subseteq \bigcup_{p \ne p^*} B_{l+1}(p)$, since $|X'| = 2 \cdot (\frac{k}{2} - 1)$, $|X' \cap B_{l+1}(p)| = 2$ for $p \ne p^*$, and sets B_{l+1} are pairwise disjoint. This implies

$$\bigcup_{p \notin A_{l+1}} B_{l+1}(p) \subseteq (X' \cup X^*) \subseteq \bigcup_{p=1}^{k/2} B_{l+1}(p) .$$

First we show that the above defined sets X' and X^* satisfy the assumptions of point 3 of INV after step l, which will prove that $(X' \cup X^*) \models (i+1)$-JAMMING$_{l'}(Y_{l'})$, for $l' \le l$. Indeed, since $X' \subseteq \bigcup_{p \ne p^*} B_{l+1}(p)$ and $B_{l+1}(p) \subseteq B_l(p) \subseteq B(p)$, we have $X' \cap B_{l+1}(p) = X' \cap B_l(p)$ for every $p \ne p^*$. Hence the condition $|X' \cap B_{l+1}(p)| = 2$ implies $|X' \cap B_l(p)| = 2$, for every $p \ne p^*$. Additionally $X \subseteq B_{l+1}(p^*) \subseteq B_l(p^*)$, and $|B_l(p^*)| \ge k\sqrt[4]{m}$. By INV after step l we have $(X' \cup X^*) \models (i+1)$-JAMMING$_{l'}(Y_{l'})$, for $l' \le l$.

In order to conclude the proof of point 3 of INV after step $l+1$, it is sufficient to show that $Y_{l+1} \cap (X' \cup X^*)$ is the empty set if $(i+1)$-JAMMING$_{l+1}(Y_{l+1}) = 0$, equals $\{v\}$ if $(i+1)$-JAMMING$_{l+1}(Y_{l+1}) = v$, and is of size at least two if $(i+1)$-JAMMING$_{l+1}(Y_{l+1}) =\perp$.

Suppose that $(i+1)$-JAMMING$_{l+1}(Y_{l+1}) = 0$. Hence point 2.B.a was applied in step $l+1$ during the execution of

1. Set $l := 0$
2. Iterate $\lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$ times
   - $l := l + 1$;
   - define set $Y_l := \{v \in R_{i+1} : \hat{\pi}(v, \hat{H}_{t_i+l-1}(v)) = 1\}$;
   - perform function $(i+1)$-JAMMING$_l(Y_l)$;
   - for any node $v \in R_{i+1}$, if $\hat{\pi}(i, \hat{H}_{t_i+l-1}(i)) = 1$ and $\hat{\pi}(v, \hat{H}_{t_i+l-1}(v)) = 0$ then define $\hat{M}_{t_i+l}(v) := (i, \hat{H}_{t_i+l-1}(i))$. Otherwise define $\hat{M}_{t_i+l}(v)$ to be the empty message;
   - for $v = i$, define $\hat{M}_{t_i+l}(i)$ as
     - $(w, \hat{H}_{t_i+l-1}(w))$, if $\hat{\pi}(i, \hat{H}_{t_i+l-1}(i)) = 0$ and there is a unique $w \in L^*_{2i-1}$ such that
       $\hat{\pi}(w, \hat{H}_{t_i+l-1}(w)) = 1$, and
       $(i+1)$-JAMMING$_l(Y_l) = 0$;
     - $(w, \hat{H}_{t_i+l-1}(w))$, if $\hat{\pi}(i, \hat{H}_{t_i+l-1}(i)) = 0$ and there is $w \in R_{i+1}$ such that
       $w = (i+1)$-JAMMING$_l(Y_l)$, and
       for every node $w' \in L^*_{2i-1}$ we have $\hat{\pi}(w', \hat{H}_{t_i+l-1}(w')) = 0$;
     - the empty message in other cases;
   - for $v \in \bigcup_{j=0}^{i-1}(L_{2j} \cup L_{2j+1})$, define $\hat{M}_{t_i+l}(v)$ using RADIO$(t_i + l, v)$
   - for other nodes v, define $\hat{M}_{t_i+l}(v)$ as the empty message.
3. Consider sets $B_l(p)$ defined in the last execution of function $(i+1)$-JAMMING. Let p^*, X', X^* be such that:
   - $|B_l(p^*)| \geq k\sqrt[4]{m}$,
   - $|X' \cap B_l(p)| = 2$, for $p \neq p^*$, and $|X'| < k$,
   - $X^* \subseteq B_l(p^*)$ is such that $|X^*| \leq k$ and X^* is a witness that the family of sets $\{Y_{l'} \cap B_l(p^*)\}_{l'=1}^{l}$ (defined for set R_{i+1}) is not a $(|B_l(p^*)|, k)$-selective family.

   (The existence of these objects will follow from invariant INV holding in part 2.)
   Define sets $L'_{2i+1} := X'$ and $L^*_{2i+1} := X^*$.
4. Iterate until $\hat{\pi}(i+1, \hat{H}_{t_i+l}(i+1)) = 1$
   Set $l := l + 1$
   - for $v = i + 1$, define $\hat{M}_{t_i+l}(v)$ as $(w, \hat{H}_{t_i+l-1}(w))$, if there is a unique $w \in L^*_{2i+1}$ such that $\hat{\pi}(w, \hat{H}_{t_i+l-1}(w)) = 1$, and as the empty message otherwise;
   - for $v \in \bigcup_{j=0}^{i}(L_{2j} \cup L_{2j+1}) \setminus \{i+1\}$, define $\hat{M}_{t_i+l}(v)$ using RADIO$(t_i + l, v)$;
   - for other nodes v, define $\hat{M}_{t_i+l}(v)$ as the empty message.
5. Set $t_{i+1} := t_i + l$
6. For every node $v \in R_{i+2}$ and step $0 \leq l' \leq t_{i+1}$, define $\hat{H}_{l'}(v)$ as the empty history.

Fig. 2. Description of stage $i+1$ of the recursive construction of network $G_{\mathcal{A}}$.

function $(i+1)$-JAMMING$_{l+1}(Y_{l+1})$. It follows that $Y_{l+1} \cap B_{l+1}(p) = \emptyset$ for every p. In this case $Y_{l+1} \cap (X' \cup X^*) \subseteq Y_{l+1} \cap \bigcup_p B_{l+1}(p) = \emptyset$.

Suppose that $(i+1)$-JAMMING$_{l+1}(Y_{l+1}) = v$. Hence point 2.B.b was applied in step $l+1$ during the execution of function $(i+1)$-JAMMING$_{l+1}(Y_{l+1})$. It follows that $Y_{l+1} \cap \bigcup_p B_{l+1}(p) = \{v\}$. Moreover $Y_{l+1} \cap B_{l+1}(p) = \{v\}$ for the unique $p = p_v$, and additionally $B_{l+1}(p_v)$ has size 2. Hence $B_{l+1}(p_v) \subseteq X'$ and

$$Y_{l+1} \cap (X' \cup X^*) \supseteq Y_{l+1} \cap B_{l+1}(p_v) = \{v\}$$
$$Y_{l+1} \cap (X' \cup X^*) \subseteq Y_{l+1} \cap \bigcup_p B_{l+1}(p) = \{v\} .$$

Suppose that $(i+1)$-JAMMING$_{l+1}(Y_{l+1}) = \perp$. Hence either point 2.A or point 2.B.c was applied in step $l+1$ during the execution of function $(i+1)$-JAMMING$_{l+1}(Y_{l+1})$. First consider the case when point 2.A was applied. Hence $B_{l+1}(p_0) \subseteq Y_{l+1}$, which implies that the set $Y_{l+1} \cap (X' \cup X^*) \supseteq B_{l+1}(p_0) \cap (X' \cup X^*)$ is of size at least 2. Finally, consider the case when point 2.B.c was applied. Then the set $Y_{l+1} \cap (X' \cup X^*) \supseteq Y_{l+1} \cap \bigcup_{p \notin A_{l+1}} B_{l+1}(p)$ is of size at least 2. This completes the proof of $(X' \cup X^*) \models (i+1)$-JAMMING$_{l+1}(Y_{l+1})$. □

Lemma 8 *The stage-invariant is satisfied after every stage $i \leq \frac{D}{2} - 1$.*

Proof. The proof is by induction on i. For $i = 1$ it is obvious. Suppose that the stage-invariant holds for $i < \frac{D}{2} - 1$. We show it for $i + 1$.

During the execution of point 2 of stage $i+1$, the abstract history of node $i+1$ is empty. During the execution of point 4 of stage $i+1$ the condition $\hat{\pi}(i+1, \hat{H}_{t_i+l}(i+1)) = 0$ holds, for $t_i + l < t_{i+1}$. This proves point 0 of the stage-invariant.

Abstract histories are defined for all nodes. We need to prove that sets L'_{2i+1} and L^*_{2i+1} are defined. This follows from Lemma 7 and, more precisely, from point 3 of INV after step $\lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$ in the execution of point 2 of stage $i+1$.

The fact that, for every v outside of $\{0\} \cup \bigcup_{j=1}^{i+1}(L_{2j} \cup L_{2j-1})$, the abstract history $\hat{H}_{t_{i+1}}(v)$ is empty, follows from the definition of histories in points 2, 4 and 6 during stage $i+1$ of the construction. □

From stage-invariant point 1, for $\frac{D}{2} - 1$, from Lemma 8, and from the definition of layer L_D it follows that network $G_{\mathcal{A}}$ is well defined.

The following lemma states that abstract and actual histories at all nodes are identical for a large number of steps.

Lemma 9 *For every node v of $G_{\mathcal{A}}$ and every step $l \le (\frac{D}{2} - 1) \cdot (\lfloor \frac{k \log(n/4)}{8 \log k} \rfloor)$, we have $H_l(v) = \hat{H}_l(v)$, where H_l is the history yielded by the run of algorithm $\mathcal{A}$ on network $G_{\mathcal{A}}$.*

Proof. The proof is by induction on l. For $l = 0$ all histories and abstract histories are empty, except for node 0. $H_0(0) = \hat{H}_0(0) = (0, source\_message)$.

Suppose that $H_l(v) = \hat{H}_l(v)$ for every node v. We will show that $H_{l+1}(v) = \hat{H}_{l+1}(v)$, for every node v, by proving that $\hat{M}_{l+1}(v) = M_{l+1}(v)$.

First suppose that step $l + 1$ occurs during the execution of point 2 of some stage $i + 1 \le D/2 - 1$.

- For $v \in R_{i+1}$ we defined $\hat{M}_{l+1}(v)$ as $(i, \hat{H}_l(i))$ if $\hat{\pi}(i, \hat{H}_l(i)) = 1$ and $\hat{\pi}(v, \hat{H}_l(v)) = 0$, and as the empty message otherwise. Notice that if $v \in R_{i+1} \setminus L_{2i+1}$ then $\hat{M}_{l+1}(v)$ is the empty message by point 6 of the construction, and also $M_{l+1}(v)$ is the empty message since $H_l(i + 1) = \hat{H}_l(i + 1)$ is the empty history and node v belongs to the layer with index larger than $2i + 2$.
  Assume that $v \in L_{2i+1}$. First consider the case when $\hat{\pi}(i, \hat{H}_l(i)) = 1$ and $\hat{\pi}(v, \hat{H}_l(v)) = 0$. Then $\hat{M}_{l+1}(v) = (i, \hat{H}_l(i))$. Consequently, we have $\pi(i, H_l(i)) = 1$ and $\pi(v, H_l(v)) = 0$ by the inductive assumption and by properties of function $\hat{\pi}$. Moreover, $H_l(i + 1) = \hat{H}_l(i + 1)$ is the empty history by the stage-invariant after stage i and by the definition of $\hat{M}_{l'}(i + 1)$ for $t_i < l' \le l$, hence $\pi(i+1, H_l(i+1)) = 0$. This proves $M_{l+1}(v) = (i, H_l(i))$ in this case.
  Now suppose that $\hat{\pi}(i, \hat{H}_l(i)) = 0$ or $\hat{\pi}(v, \hat{H}_l(v)) = 1$. This implies that $\pi(i, H_l(i)) = 0$ or $\pi(v, H_l(v)) = 1$. If $\pi(i, H_l(i)) = 0$ then, by the fact that $H_l(i + 1)$ is the empty history, we have $\pi(i+1, H_l(i+1)) = 0$, and consequently $M_{l+1}(v)$ is the empty message. If $\pi(v, H_l(v)) = 1$ then $M_{l+1}(v)$ is also the empty message. Hence it equals $\hat{M}_{l+1}(v)$
- For $v = i$ we defined $\hat{M}_{l+1}(i)$ as
  - $(w, \hat{H}_l(w))$, if $\hat{\pi}(i, \hat{H}_l(i)) = 0$ and there is a unique $w \in L^*_{2i-1}$ such that $\hat{\pi}(w, \hat{H}_l(w)) = 1$ and $(i + 1)$-JAMMING$_{l+1-t_i}(Y_{l+1-t_i}) = 0$. It follows from Lemma 7, that invariant INV holds for $X' = L'_{2i+1}$ and $X^* = L^*_{2i+1}$ after $\lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$ steps in point 2 of stage $i + 1$. This implies that $L_{2i+1} \cap Y_{l+1-t_i} = \emptyset$. By the inductive assumption about histories, we obtain that no node in L_{2i+1} transmits in step $l + 1$ of algorithm $\mathcal{A}$. On the other hand, by the inductive assumption, w is the unique node in L^*_{2i-1} which transmits in step $l + 1$ of algorithm $\mathcal{A}$. Consequently $M_{l+1}(i) = (w, H_l(w)) = (w, \hat{H}_l(w))$.
  - $(w, \hat{H}_l(w))$, if $\hat{\pi}(i, \hat{H}_l(i)) = 0$ and there is $w \in R_{i+1}$ such that $w = (i + 1)$-JAMMING$_{l+1-t_i}(Y_{l+1-t_i})$ and for every $w' \in L^*_{2i-1}$ we have $\hat{\pi}(w', \hat{H}_l(w')) = 0$. Similar arguments as in the previous case show, that w is the unique node in L_{2l+1} which transmits in step $l+1$ of algorithm $\mathcal{A}$, and no node in L^*_{2i-1} transmits in this step. Hence $M_{l+1}(i) = (w, H_l(w)) = (w, \hat{H}_l(w))$.
  - the empty message in other cases. First note that if $\hat{\pi}(i, \hat{H}_l(i)) = 1$ then, by the inductive assumption, $\pi(i, H_l(i)) = 1$, and consequently $M_{l+1}(i)$ is the empty message. Suppose that $\hat{\pi}(i, \hat{H}_l(i)) = 0$. If $(i+1)$-JAMMING$_{l+1-t_i}(Y_{l+1-t_i}) = \perp$ then, similarly as above, at least two nodes from L_{2i+1} transmit in step $l + 1$ of algorithm $\mathcal{A}$, and then $M_{l+1}(i)$ is the empty message.
    If $(i + 1)$-JAMMING$_{l+1-t_i}(Y_{l+1-t_i}) = w$ but there is a node $w' \in L^*_{2i-1}$ such that $\hat{\pi}(w', \hat{H}_l(w')) = 1$ then, by the inductive assumption, we have also $\pi(w, H_l(w)) = 1$ and $\pi(w', H_l(w')) = 1$. Consequently, $M_{l+1}(i)$ is the empty message. Finally, if $(i + 1)$-JAMMING$_{l+1-t_i}(Y_{l+1-t_i}) = 0$ but $|T_{l+1} \cap L^*_{2i-1}| \ne 1$, then $M_{l+1}(i)$ is also the empty message.
- For $v \in \bigcup_{j=0}^{i-1}(L_{2j} \cup L_{2j+1})$ we defined $\hat{M}_{l+1}(v)$ using RADIO$(l + 1, v)$. By properties of procedure RADIO and the inductive assumption, we obtain that $M_{l+1}(v)$ is equal to $\hat{M}_{l+1}(v)$.
- For all other nodes v we defined $\hat{M}_{l+1}(v)$ as the empty message. We have just proved that $H_l(i + 1)$ is the empty history and consequently $\pi(i + 1, H_l(i + 1)) = 0$. If v is different from $i + 1$, then it is in a layer with index larger than $2i + 2$, and we conclude that $H_{l+1}(v)$ is the empty history.
  It remains to consider the case $v = i + 1$, which is more dificult. We show that $M_{l+1}(i + 1)$ is the empty message. We have just proved that $H_l(i + 1)$ is the empty history. It follows that every neighbor from L_{2i+3} of node $i + 1$ has also the empty history after step l and hence does not transmit in step $l + 1$. Consider neighbors in L^*_{2i+1}. This set was defined after the end of point 2 of stage $i+1$, considering abstract histories of nodes in block $B_{t_i+\lfloor \frac{k \log(n/4)}{8 \log k} \rfloor}(p^*) \subseteq R_{i+1}$. By definition, in step l such that $t_i < l \le t_i + \lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$, the set $\{w \in L^*_{2i+1} : \hat{\pi}(w, \hat{H}_l(w)) = 1\}$ is of size different from 1. In view of the equality $\hat{H}_l(w) = H_l(w)$ and of the definition of function $\hat{\pi}$, we obtain $|\{w \in L^*_{2i+1} : \pi(w, H_l(w)) = 1\}| \ne 1$, and consequently $M_{l+1}(i + 1)$ is the empty message.

Now suppose that step $l + 1$ occurs during the execution of point 4 of some stage $i + 1 \le D/2 - 1$.

- For $v = i + 1$, $\hat{M}_{l+1}(v)$ was defined as $(w, \hat{H}_l(w))$, if there is a unique $w \in L^*_{2i+1}$ such that $\hat{\pi}(w, \hat{H}_l(w)) = 1$, and as the empty message otherwise. In the first case we have also $\pi(w, H_l(w)) = 1$, for other nodes $z \in L^*_{2i+1}$ we have $\pi(z, H_l(z)) = 0$, and for nodes $z \in L_{2i+3}$ we have $\pi(z, H_l(z)) = 0$, in view of $\hat{H}_l = H_l$ and becsuse $\hat{H}_l(z)$ is the empty history for every $z \in \bigcup_{j=i+1}^{D/2-1}(L_{2j+1} \cup L_{2j+2})$. Hence $M_{l+1}(v) = (w, H_l(w)) = (w, \hat{H}_l(w))$. In the second case similar arguments show that $M_{l+1}(v)$ is the empty message.

- For $v \in \bigcup_{j=0}^{i}(L_{2j} \cup L_{2j+1}) \setminus \{i+1\}$, we defined $\hat{M}_{l+1}(v)$ using $\text{RADIO}(l+1, v)$. By the formulation of procedure $\text{RADIO}(l+1, v)$ and by the inductive assumption $\hat{H}_l = H_l$, we obtain the required equality.
- For other nodes v, we defined $\hat{M}_{l+1}(v)$ as the empty message. In this case v is in a layer with index larger than $2(i+1)$. By the stage-invariant (cf. Lemma 8), after stage $i+1$ we know that $\hat{\pi}(i+1, \hat{H}_l(i+1)) = 0$. Consequently, by the inductive assumption $\hat{H}_l = H_l$, we obtain $\pi(i+1, H_l(i+1)) = 0$. Hence $M_{l+1}(v)$ is the empty message. □

Proof of Theorem 2. For $D > \frac{n}{16}$, the result holds in view of the lower bound $\Omega(D \log n)$ proved in [5]. For $D \leq \sqrt[4]{n^3}$, the result follows from the lower bound $\Omega(n)$ from [3] (correct under our scenario), cf. also [15]. Hence it is sufficient to consider the case $\sqrt[4]{n^3} < D \leq \frac{n}{16}$, for which the previous construction was made.

Every layer of graph $G_{\mathcal{A}}$ is well defined in view of Lemma 8. The graph contains all nodes from 0 to n and has radius D. Node $\frac{D}{2} - 1$ in layer L_{D-2} does not transmit before step $\left(\frac{D}{2} - 1\right) \cdot \lfloor \frac{k \log(n/4)}{8 \log k} \rfloor$ of algorithm $\mathcal{A}$, in view of Lemma 8 and Lemma 9. Hence algorithm $\mathcal{A}$ requires time $\Omega(n \frac{\log n}{\log(n/D)})$ to broadcast on graph $G_{\mathcal{A}}$. □

4 Deterministic broadcasting algorithms

4.1 Simulating collision detection

Our algorithms in Sect. 4.2 use a technique of simulating collision detection which is not *a priori* available in our model. We introduced this technique in [13]. Consider a node v which already has the source message, a set A of neighbors of v, and a distinguished neighbor $w \notin A$. Our goal is to let the the node v distinguish whether A has 0,1, or more than 1 element. This can be done using the following 2-step procedure:

Procedure $\text{ECHO}(w, A)$

Step 1. Every node in A transmits its label.
Step 2. Every node in $A \cup \{w\}$ transmits its label.

There are 3 possible effects of Procedure $\text{ECHO}(w, A)$ at node v:

Case 1. v receives a message in step 1 and no message in step 2. In this case v knows that A has 1 node and knows the label of this unique node.
Case 2. v receives no message in step 1 and receives a message (from w) in step 2. In this case v knows that A is empty.
Case 3. v receives no message in either step. In this case v knows that A has at least 2 nodes.

Suppose that node v knows one of its neighbors w. Denote by S a set of neighbors of v different from w. Suppose that v also knows that some nodes in S have labels at most m, where m is a power of 2. Then Procedure ECHO can be used by v to select one neighbor from S in time $\mathcal{O}(\log m)$. This is done by the following algorithm:

Algorithm Binary-Selection

Time steps are divided into segments of length 3. In the first step of each segment, node v transmits a range R of labels, and orders the execution of Procedure $\text{ECHO}(w, R \cap S)$ during the last two steps of the segment. In the first segment, $R := \{1, ..., m/2\}$. If a range $R = \{x, ..., y\}$ is transmitted in a given segment, the range to transmit in the next segment is chosen according to the three possible effects of $\text{ECHO}(w, R \cap S)$, described above. In case 1, a single neighbor from S is selected. In case 2, $R := \{y+1, \ldots, y + (y - x + 1)/2\}$. In case 3, $R := \{x, \ldots, (y + x - 1)/2\}$.

4.2 Algorithms for arbitrary networks

We present an algorithm working in time $\mathcal{O}(n \log n)$ for arbitrary n-node networks. The algorithm is based on a DFS visit of all nodes of the network by a token, where the next node to visit is chosen using Algorithm Binary-Selection.

Algorithm Select-and-Send

1. In the beginning the token is at node 0 (the source). In step 1 the source sends a message ordering its neighbor with label i to transmit a message in step $2i$, for all $i > 0$. After receiving the first message in step $2j$ the source sends a message in step $2j+1$ ordering to stop this procedure, and sends the token to node with label j.
2. For $v \neq 0$, $parent(v)$ denotes the node from which v got the token for the first time. At each step, V denotes the set of nodes already visited by the token. If the token is at node v, S denotes the set of neighbors of v outside V.
   If the token is at node $v \neq 0$, this node sends the source message (which results in waking up all neighbors and allowing them to transmit), and initiates Procedure $\text{ECHO}(parent(v), S)$. If the token is at node 0 (after step 1), this node calls Procedure $\text{ECHO}(j, S)$. Depending on the outcome of this procedure, node v acts as follows:
   - If $|S| = 0$ and $v \neq 0$ then v sends the token to $parent(v)$ and stops.
   - If $|S| = 0$ and $v = 0$ then v stops.
   - If $|S| = 1$ then v sends the token to the unique node in S.
   - If $|S| > 1$ then v initiates Procedure $\text{ECHO}(parent(v), S \cap [1, ..., 2^k])$, for $k = 1, 2, ...$, until $S \cap [1, ..., 2^k]$ is nonempty. (If $v = 0$, Procedures $\text{ECHO}(j, S \cap [1, ..., 2^k])$ are used instead). Then v selects one neighbor w in $S \cap [1, ..., 2^k]$ using Algorithm Binary-Selection. v sends the token to w.

Theorem 3 *Algorithm Select-and-Send performs broadcasting in any n-node network in time $\mathcal{O}(n \log n)$.*

Proof. The algorithm sends a token equipped with the source message from one node at a time, hence no collisions can occur during this sending. Since the token visits all nodes in a DFS manner, all nodes get the message. The only difficulty is the selection of exactly one unvisited neighbor of the currently visited node: this node will receive the token next. In the beginning, when the token starts at the source, this selection is done by reserving a different time unit for each possible neighbor and waiting for the lowest-labeled neighbor to reply. In the remaining steps of the algorithm the selection is done using Algorithm Binary-Selection. Hence exactly one unvisited neighbor of the current node can be selected whenever such a node exists. Consequently the DFS traversal of the graph by the token can be performed which proves that the algorithm is correct.

The complexity of the algorithm can be computed as follows. Part 1 takes time $\mathcal{O}(n)$, because the upper bound r on all labels is linear in n. In part 2, selecting a new node takes $\mathcal{O}(\log n)$ calls of Procedure ECHO, and additionally, every node can make at most one call of this procedure resulting in an empty set S. Thus the total number of calls of Procedure ECHO is $\mathcal{O}(n \log n)$. Consequently, the algorithm works in time $\mathcal{O}(n \log n)$. $\square$

Observe that repeated use of the round-robin scheme gives a broadcasting algorithm working in time $\mathcal{O}(nD)$ which is faster than $\mathcal{O}(n \log n)$ for very small D. Interleaving both algorithms, we get broadcasting in time $\mathcal{O}(n \min(D, \log n))$, in any n-node network of radius D.

4.3 Broadcasting in complete layered networks

Let L_i denote the ith layer of a network. A complete layered (undirected) network of radius D is a network $G = (V, E)$, such that $E = \{\{x, y\} : \exists i < D \ (x \in L_i, y \in L_{i+1})\}$. A directed version of this notion is simply obtained by replacing undirected edges $\{x, y\}$ by directed edges (x, y).

In [10] the authors constructed, for any broadcasting algorithm, a directed n-node complete layered network of radius D, such that this algorithm requires time $\Omega(n \log D)$ to broadcast on this network. This implies the lower bound $\Omega(n \log D)$ on broadcasting time in directed n-node networks of radius D. It is claimed in [10] that the same argument shows the lower bound $\Omega(n \log D)$ on broadcasting time for *undirected* networks, if spontaneous transmissions are not allowed. Unfortunately, this claim is incorrect. Indeed, we construct a broadcasting algorithm which works in time $\mathcal{O}(n + D \log n)$ for all undirected complete layered n-node networks of radius D, even if spontaneous transmissions are not allowed. For all unbounded $D \in o(n)$ this is faster than the claimed lower bound.

Algorithm Complete-Layered

The algorithm works in phases. In phase 1 the source first sends the source message and orders its neighbor with label i to transmit a message in step $2i$, for $i > 0$. After receiving the first message in step $2j$ the source sends a message in step $2j + 1$ ordering to end this phase.

We preserve the following invariant after phase k, where $k \geq 1$.

- All nodes of layers L_i, for $i \leq k$ got the source message and know their layer number.
- A node $v_i \in L_i$ is selected, for all $i \leq k$, node v_i knowing v_{i-1}. v_0 is the source and $v_1 = j$.
- All nodes in layers L_i, for $i \leq k - 2$ have stopped.

In phase $k + 1$, node v_k first sends the source message (waking up all nodes in L_{k+1}). Then it initiates Procedure ECHO(v_{k-1}, S), where S is the set of neighbors of v_k which obtained the source message in the previous step (these are all nodes in L_{k+1}). If the outcome of this procedure is $|S| = 0$ (this means that $D = k$) then node v_k sends a message ordering all of its neighbors to stop and stops itself. Otherwise, v_k initiates Procedure ECHO$(v_{k-1}, S \cap [1, ..., 2^p])$, for $p = 1, 2, ...$, until $S \cap [1, ..., 2^p]$ is nonempty. Then v_k selects one neighbor v_{k+1} in $S \cap [1, ..., 2^p]$ using Algorithm Binary-Selection. After being selected, v_{k+1} knows the identity of v_k. Finally, node v_k orders all of its neighbors in L_{k-1} to stop. This terminates phase $k + 1$. The invariant is preserved.

Notice that the lower bound $\Omega(D \log n)$ from [5] holds even for the class of complete layered networks. The lower bound $\Omega(n)$ holds even for the class of complete layered networks of radius 2 [3,15]. (It should be noted that the latter bound holds only under our scenario where nodes do not know their neighborhood). Hence our algorithm is optimal for the class of complete layered networks.

Theorem 4 *Algorithm Complete-Layered performs broadcasting in an arbitrary n-node complete layered network of radius D, in time $\mathcal{O}(n + D \log n)$.*

Proof. The correctness of the algorithm is straightforward. Phase 1 takes time $\mathcal{O}(n)$ and each of the subsequent $D - 1$ phases takes time $\mathcal{O}(\log n)$. $\square$

5 Conclusion and open problems

We considered broadcasting time in undirected radio networks of unknown topology. We proved the upper bound $\mathcal{O}(D \log(n/D) + \log^2 n)$ for randomized broadcasting (it also holds in the more general setting of directed graphs). This closes the gap on randomized broadcasting time, in view of the lower bounds of Alon et al. [1] and Kushilevitz and Mansour [16]. For deterministic broadcasting time, a small gap still remains. In this paper we obtained the best lower bound $\Omega(n \frac{\log n}{\log(n/D)})$ known to date. As for the best upper bound, the $\mathcal{O}(n \log n)$-time algorithm obtained in this paper, together with the recent result from [11], give $\mathcal{O}(n \cdot \min\{\log^2 D, \log n\})$. Closing the gap between these bounds is an interesting open problem.

Acknowledgements. We are grateful to an anonymous referee who pointed out an error in the first version of one of the proofs. We also benefited from valuable editorial suggestions of all referees.

References

1. N. Alon, A. Bar-Noy, N. Linial and D. Peleg, A lower bound for radio broadcast, Journal of Computer and System Sciences 43 (1991), 290–298
2. B. Awerbuch, A new distributed depth-first-search algorithm, Information Processing Letters 20 (1985), 147–150
3. R. Bar-Yehuda, O. Goldreich and A. Itai, On the time complexity of broadcast in radio networks: an exponential gap between determinism and randomization, Journal of Computer and System Sciences 45 (1992), 104–126
4. S. Basagni, D. Bruschi and I. Chlamtac, A mobility–transparent deterministic broadcast mechanism for ad hoc networks, IEEE/ACM Trans. on Networking 7 (1999), 799–807
5. D. Bruschi and M. Del Pinto, Lower bounds for the broadcast problem in mobile radio networks, Distributed Computing 10 (1997), 129–135
6. I. Chlamtac and O. Weinstein, The wave expansion approach to broadcasting in multihop radio networks, IEEE Trans. on Communications 39 (1991), 426–433
7. B. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc and W. Rytter, Deterministic broadcasting in unknown radio networks, Distributed Computing 15 (2002), 27–38.
8. B. Chlebus, L. Gąsieniec, A. Östlin and J.M. Robson, Deterministic radio broadcasting, Proc. 27th Int. Coll. on Automata, Languages and Programming (ICALP'2000), 717–728
9. M. Chrobak, L. Gąsieniec and W. Rytter, Fast broadcasting and gossiping in radio networks, Journal of Algorithms 43 (2002), 177–189
10. A.E.F. Clementi, A. Monti and R. Silvestri, Selective families, superimposed codes, and broadcasting on unknown radio networks, Proc. 12th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'2001), 709–718
11. A. Czumaj and W. Rytter, Broadcasting algorithms in radio networks with unknown topology, Proc. 44th Ann.Symposium on Foundations of Computer Science (FOCS'2003), 492–501
12. I. Gaber and Y. Mansour, Centralized broadcast in multihop radio networks, Journal of Algorithms 46(1) (2003), 1–20
13. D. Kowalski and A. Pelc, Time of deterministic broadcasting in radio networks with local knowledge, SIAM Journal on Computing 33 (2004), 870–891
14. D. Kowalski and A. Pelc, Faster deterministic broadcasting in ad hoc radio networks, SIAM Journal on Discrete Mathematics 18 (2004), 332–346
15. D. Kowalski and A. Pelc, Time complexity of radio broadcasting: adaptiveness vs. obliviousness and randomization vs. determinism, Theoretical Computer Science 333 (2005), 355–371
16. E. Kushilevitz and Y. Mansour, An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks, SIAM Journal on Computing 27 (1998), 702–712