Broadcasting in Undirected Ad hoc Radio Networks

Dariusz R. Kowalski [*]
Instytut Informatyki
Uniwersytet Warszawski
Banacha 2, 02-097 Warszawa, Poland
darek@mimuw.edu.pl

Andrzej Pelc [†]
Département d'Informatique
Université du Québec en Outaouais
Hull, Québec J8X 3X7 Canada
pelc@uqo.ca

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 [14]. Our algorithm improves the best previously known randomized broadcasting algorithm of Bar-Yehuda, Goldreich and Itai [3], running in expected time $\mathcal{O}(D\log n+\log^2 n)$. 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.

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, 8, 9, 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 [11], $\mathcal{O}(D+\log^5 n)$-time broadcasting was proposed. 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 $\Omega(\log^2 n)$.

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]. In [3], the authors constructed a class of n-node graphs of radius 2, and claimed that every broadcasting algorithm re-

[*]This work was done during this author's stay at the Research Chair in Distributed Computing of the Université du Québec en Outaouais, as a postdoctoral fellow.

[†]Research supported in part by NSERC grant OGP 0008136 and by the Research Chair in Distributed Computing of the Université du Québec en Outaouais.

quires time $\Omega(n)$ on one of these graphs. Unfortunately, as we showed in [12], this result is incorrect. However, the linear lower bound and the argument hold if every node knows only its own label.

Many authors [4, 5, 7, 8, 9, 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. On the other hand, in [5] a lower bound $\Omega(D \log n)$ on deterministic broadcasting time was proved for n-node networks of radius D, if spontaneous transmissions are not allowed.

In [7, 8, 9, 13] the model of directed graphs was used. Increasingly faster broadcasting algorithms working on arbitrary (directed) radio networks were constructed, the currently fastest being the $\mathcal{O}(n \log n \log D)$-time algorithm from [13]. Although designed for arbitrary directed networks, it obviously works for undirected networks as well, and is, in fact, the currently fastest known deterministic algorithm in our model, working for arbitrary n-node networks of radius D. 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, 14]. 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 [14] 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. 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. It 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 [14]. 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 $D \in \Theta(n/\text{polylog }(n))$.

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 D, on which this algorithm requires time $\Omega(n \frac{\log n}{\log(n/D)})$ to broadcast. The two sharpest previously known lower bounds on deterministic broadcasting time were: the lower bound $\Omega(n)$ from [3] (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). 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 $\mathcal{O}(n \log^2 n)$-time algorithm from [9], 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. A simple round robin scheme yields an algorithm working in time $\mathcal{O}(nD)$, for n-node networks of radius D. Together with our algorithm, this gives an upper bound $\mathcal{O}(n \cdot \min(D, \log n))$ which is of optimal order of magnitude for extreme values of the radius: for constant D and for $D \in \Theta(n)$, in view of our lower bound $\Omega(n \frac{\log n}{\log(n/D)})$.

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 wrong. Indeed, we construct a broadcasting algorithm which works in time $O(n + D \log n)$ 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 [14] 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 $O(n + D \log n)$ 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 undirected connected graphs (networks) 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, 14]. A broadcasting algorithm works in time t, if every node gets the source message after at most t steps.

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, 14].

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 neighbors gets the source message with high probability. Nodes with many informed 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 neighbors of a node get the source message (for $x > r/D$), this node is also informed with high 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 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 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 2.1. *For sufficiently large r and every D such that $32r^{2/3} < D \le r$, there exists a universal sequence.*

PROOF. The proof will appear in the full version of the paper. □

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 the 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]. Until the formulation of our main Algorithm Optimal-Randomized-Broadcasting, we assume $D>32r^{2/3}$.

Fix a 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 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 nodes adjacent to 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 degree of node v_j in P_v.

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, all nodes $v_{j+1},\ldots,v_k$ have no neighbor in P_v having the source message but v_j has such a 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 whose fairly technical proofs will appear in the full version of the paper. 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 2.2, 2.3 and 2.4 estimate this probability for different ranges of degree d_j (notice that x varies depending on the range of d_j). Lemma 2.5 uses these estimates to show that information front reaches v in $\mathcal{O}(D)$ stages, with high probability.

LEMMA 2.2. *For every $j\le k$, if $d_j\le r/D$ then*
$$\Pr[\,\mathcal{E}_{i+1,j}\mid\mathcal{E}_{i,j}\,]<\tfrac{7}{8}\ .$$

LEMMA 2.3. *For every $j\le k$, if $\log\frac{r}{D}<\log d_j\le\lceil\log\frac{r}{4\log r}\rceil$ then*
$$\Pr\left[\,\mathcal{E}_{i+\frac{6D\cdot 2^{\lceil\log d_j\rceil}}{r},j}\;\middle|\;\mathcal{E}_{i,j}\,\right]<\tfrac{7}{8}\ .$$

LEMMA 2.4. *For every $j\le k$, if $\lceil\log\frac{r}{4\log r}\rceil<\log d_j\le\log r$ then*
$$\Pr\left[\,\mathcal{E}_{i+\frac{6D\cdot 2^{\lceil\log d_j\rceil}}{r\cdot 2^{\lceil\log\log r\rceil+1}},j}\;\middle|\;\mathcal{E}_{i,j}\,\right]<\tfrac{7}{8}\ .$$

FACT 1. *Let B be any nonempty set of indices, $(\alpha_j)_{j\in B}$ and $(\beta_j)_{j\in B}$ be two sequences of positive reals. If $\sum_{j\in B}\alpha_j\ge\alpha\sum_{j\in B}\beta_j$, for some $\alpha\ge 1$, then*
$$\sum_{j\in B}\frac{\alpha_j}{\beta_j}\ge\alpha\cdot|B|\ .$$

PROOF. By induction on the size of B. $\square$

LEMMA 2.5. $\Pr[\,\mathcal{E}_{4660D,k}\,] \geq 1 - \frac{1}{r^2}$.

PROOF. (Sketch) Let a be a sequence $(v_{j_1}, \ldots, v_{j_{4660D}})$, where $(j_i)_{i \leq 4660D}$, $j_i \leq k$, 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 $(t_i)_{i \leq u}$, $t_i \leq k$, 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 $x_l = |\{v_j : \exists_i\ v_{j_i} = v_j \wedge \lceil \log d_j \rceil = l\}|$.

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

$$\mathcal{C}_1 = \left(\left|\left\{v_{j_i} \colon \mathcal{E}_{i,j_i} \wedge d_{j_i} \leq \frac{r}{D}\right\}\right| > 49D\right),$$

$$\mathcal{C}_2 = \left(\left|\left\{v_{j_i} \colon \mathcal{E}_{i,j_i} \wedge \log\frac{r}{D} < \lceil \log d_{j_i} \rceil \leq \lceil \log \frac{r}{4\log r} \rceil\right\}\right| > 48^2 D \wedge \sum_{l=\log(r/D)+1}^{\lceil \log \frac{r}{4\log r}\rceil} x_l > \log r\right),$$

$$\mathcal{C}_3 = \left(\left|\left\{v_{j_i} \colon \mathcal{E}_{i,j_i} \wedge \log\frac{r}{D} < \lceil \log d_{j_i} \rceil \leq \lceil \log \frac{r}{4\log r} \rceil\right\}\right| > 48^2 D \wedge \sum_{l=\log(r/D)+1}^{\lceil \log \frac{r}{4\log r}\rceil} x_l \leq \log r\right),$$

$$\mathcal{C}_4 = \left(\left|\left\{v_{j_i} \colon \mathcal{E}_{i,j_i} \wedge \lceil \log \frac{r}{4\log r} \rceil < \lceil \log d_{j_i} \rceil \leq \log r\right\}\right| > 48^2 D\right).$$

We have

$$\Pr\left[\neg\mathcal{E}_{4660D,k} \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\right] \leq \Pr\left[\mathcal{C}_1 \vee \mathcal{C}_2 \vee \mathcal{C}_3 \vee \mathcal{C}_4 \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\right].$$

We now separately estimate $\Pr[\,\mathcal{C}_i \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$, for $i = 1, 2, 3, 4$.

Estimation of $\Pr[\,\mathcal{C}_1 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$.
Let $x_{small} = \sum_{l \leq \log(r/D)} x_l \leq D$. Using Lemma 2.2, and the fact that, for different nodes v_j, we consider disjoint parts of the computation, we have that the probability

$$\Pr[\,\mathcal{C}_1 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,] = \Pr\left[\,\left|\left\{v_{j_i} : \mathcal{E}_{i,j_i} \wedge d_{j_i} \leq \frac{r}{D}\right\}\right| > 48D + D \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,\right]$$

is at most

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

Using inequalities $\frac{x^x}{e^x} \leq x! \leq \frac{x^{x+1}}{e^x}$, for $x \geq 2$, we obtain

$$\sum_{m=48D+1}^{\infty} \binom{m+x}{x}\left(\frac{7}{8}\right)^m \leq \sum_{m=48D+1}^{\infty} (m+x)\left(\frac{m+x}{x}\right)^x (0.89)^m.$$

Using the inequality $\left(\frac{m+x}{x}\right)^x \leq (1.08)^m$, for $m > 48x$, we finally obtain

$$\Pr\left[\,\mathcal{C}_1 \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,\right] \leq (0.9)^{48D} + \sum_{x_{small}=3}^{D} \sum_{m=48D+1}^{\infty} (m+x) \cdot (1.08)^m (0.89)^m,$$

which is at most $(0.4)^D$ for sufficiently large r. Since $D > 32r^{2/3}$ we obtain $(0.4)^D \leq \frac{1}{3r^2}$, for sufficiently large r.

Estimation of $\Pr[\,\mathcal{C}_2 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$.
Let

$$B_{med} = \left\{j : \exists_i\ v_{j_i} = v_j \wedge \log(r/D) < \lceil \log d_j \rceil \leq \lceil \log \frac{r}{4\log r} \rceil\right\}.$$

Let $n_j = |\{i : v_{j_i} = v_j\}|$ and $x_{med} = |B_{med}| = \sum_{l=\log(r/D)+1}^{\lceil \log \frac{r}{4\log r}\rceil} x_l$. Using Fact 1 (for $\alpha_j = \frac{n_j}{D}$, $\beta_j = \frac{2^{\lceil \log d_j \rceil}}{r}$, $\alpha = 48 \cdot 24$ and $B = B_{med}$) we obtain that $\Pr[\,\mathcal{C}_2 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$ is at most

$$\Pr\left[\,\sum_{j \in B_{med}} \frac{n_j}{2 \cdot \frac{6D2^{\lceil \log d_j \rceil}}{r}} > \frac{1}{12} \cdot (48 \cdot 24) \cdot x_{med} \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,\right].$$

Using Lemma 2.3 and the fact that for different nodes v_j we consider disjoint parts of the computation, we conclude that

$$\Pr\left[\,\mathcal{C}_2 \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,\right] \leq (0.4)^{2x_{med}} \leq \frac{1}{3r^2 \log r},$$

for sufficiently large r (calculations are similar to those in the estimation of $\Pr[\,\mathcal{C}_1 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$, replacing x_{small} by x_{med} and D by $2x_{med}$).

Estimation of $\Pr[\,\mathcal{C}_3 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$.
Observe that

$$\Pr\left[\,\mathcal{C}_3 \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,\right] = 0,$$

because the condition

$$\left|\left\{v_{j_i} : \mathcal{E}_{i,j_i} \wedge \log(r/D) < \lceil \log d_{j_i} \rceil \leq \lceil \log \frac{r}{4\log r} \rceil\right\}\right| > 48^2 D$$

is never satisfied under the assumption $\bigcup_{a:a\sqsubset b} \mathcal{E}(a)$.

Estimation of $\Pr[\,\mathcal{C}_4 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$.
Let

$$B_{large} = \left\{j : \exists_i\ v_{j_i} = v_j \wedge \lceil \log \frac{r}{4\log r} \rceil < \lceil \log d_j \rceil \leq \log r\right\}$$

and $x_{large} = |B_{large}|$. Using Fact 1 (for $\alpha_j = \frac{n_j}{D}$, $\beta_j = \frac{2^{\lceil \log d_j \rceil}}{r}$, $\alpha = 48 \cdot 24$ and $B = B_{large}$) we obtain that

$$\Pr[\,\mathcal{C}_4 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$$

is at most

$$\Pr\left[\,\sum_{j \in B_{large}} \frac{n_j}{2 \cdot \frac{6D2^{\lceil \log d_j \rceil}}{r \cdot 2^{\lceil \log\log r \rceil + 1}}} > \frac{\log r}{6} \cdot (48 \cdot 24) \cdot 1 \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,\right].$$

Using Lemma 2.4 and the fact that, for different nodes v_j, we consider disjoint parts of the computation, we get the bound

$$\Pr\left[\,\mathcal{C}_4 \;\middle|\; \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,\right] \leq (0.4)^{4\log r} \leq \frac{1}{3r^2 \log r},$$

for sufficiently large r (calculations are similar to those in the estimation of $\Pr[\,\mathcal{C}_1 \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$, replacing x_{small} by x_{large} and D by $4\log r$).

Finally we get

$$\sum_{i=1}^{4} \mathsf{Pr}[\, \mathcal{C}_i \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,] \leq \mathsf{Pr}[\, \neg\mathcal{E}_{4\cdot 48\cdot D,k} \mid \bigcup_{a:a\sqsubset b} \mathcal{E}(a)\,]$$
$$\leq \frac{1}{3r^2} + \log D \cdot \frac{1}{3r^2\log r} + (\log\log r + 3)\cdot\frac{1}{3r^2\log r} ,$$

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

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

for sufficiently large r. □

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 2.1. *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 graph G of radius D. If $D \leq 32r^{2/3}$, the result follows from [3]. Suppose that $D > 32r^{2/3}$ and let $i = \lceil \log D\rceil$. By Lemma 2.5, 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 $j \leq i$ 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 $\mathcal{O}(2^i\log(n/2^i)) = \mathcal{O}(D\log(n/D))$. □

COROLLARY 2.1. *There exists a randomized algorithm which performs broadcasting on any n-node network of radius D in expected time $\mathcal{O}(D\log(n/D) + \log^2 n)$.*

PROOF. Interleave Algorithm Optimal-Randomized-Broadcasting with our deterministic Algorithm Select-and-Send (described in Section 3) working in $\mathcal{O}(r\log r)$ stages, i.e., $\mathcal{O}(r\log r\log(r/D))$ steps. Hence the expected broadcasting time of the resulting algorithm is at most

$$\mathcal{O}\big(D\log(r/D) + \frac{1}{r}\cdot r\log r\cdot\log(r/D)\big) =$$
$$= \mathcal{O}(D\log(n/D) + \log n\cdot\log(n/D)) = \mathcal{O}(D\log(n/D)) ,$$

if $D > 32r^{2/3}$. If $D \leq 32r^{2/3}$ then by [3] the expected time is $\mathcal{O}(D\log(n/D) + \log^2 n)$. □

Note that our bound on randomized broadcasting time also holds for directed graphs, with small modifications of the proof.

3. DETERMINISTIC BROADCASTING

3.1 The lower bound

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. 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. This is done using properties of function NUMTRANS (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.

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 $(0, \mathit{source\_message})$. $M_l(v)$ (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 *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: $\hat{T}_k(v) = \{w \in N_v : \hat{\pi}(w, \hat{H}_{k-1}(w)) = 1\}$.

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

THEOREM 3.1. *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 D, such that $\mathcal{A}$ 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. First assume that $\sqrt[4]{n^3} < D \leq \frac{n}{16}$. 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. We may assume that D is even. 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 into nonempty sets $L^{'}_{2i+1}$ and L^{*}_{2i+1}, each of size at most $k = \lfloor \frac{n}{4D} \rfloor$. Notice that $4 \leq k < \sqrt[4]{n/4}$. 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 $i = 0, \ldots, \frac{D}{2} - 1$. Layer L_D contains all nodes outside of any previous layer (see Fig. 1). We denote the family of all such networks by $C_{n,D}$.

The construction proceeds in stages. During stage $0 \leq i \leq \frac{D}{2} - 1$, 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 $0 \leq i < \frac{D}{2} - 1$ 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.

We define the procedure of obtaining 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 defined.

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 $\hat{M}_j(v) := (w, \hat{H}_{l-1}(w))$

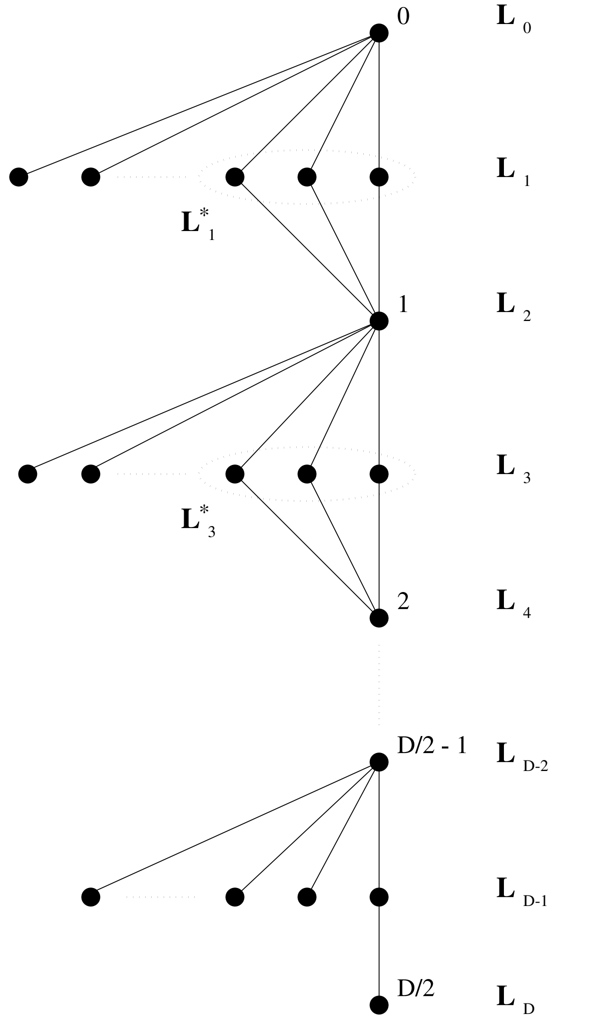

Figure 1: Network $G_{\mathcal{A}}$ used to prove the lower bound for deterministic broadcasting.

We now describe stage $i + 1$ of the construction. First we define the function $(i + 1)$-NUMTRANS, which is the main combinatorial ingredient of the proof. 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 $m > n/4$. 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 $p \leq k/2$. Let $A_l = \{p \leq k/2 : |B_l(p)| \geq k\}$. For $1 \leq l \leq \frac{k \log m}{8 \log k} - 1$ and for a set Y_l, we define the function $(i+1)$-NUMTRANS$_l(Y_l)$ which returns either the number 0, or a node v, or the symbol $\perp$.

We first describe the intuitive goal of executing function $(i + 1)$-NUMTRANS. 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 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 a subset L^*_{2i+1} of L_{2i+1}, such that no single node from this subset will transmit fast. To do so, function $(i{+}1)$-NumTrans$_l(Y_l)$ 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 $i+1$. It 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 $|X \cap B_l(p^*)| \geq 2$. 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 $(i+1)$-NumTrans$_l(Y_l)$

1. For all $p \leq k/2$ set $B_l(p) := B_{l-1}(p)$.
2. We modify sets $B_l(p)$ and define $(i{+}1)$-NumTrans$_l(Y_l)$ 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 $(i{+}1)$-NumTrans$_l(Y_l) :=\perp$ and $B_l(p_0) := B_{l-1}(p_0) \cap Y_l$. (Notice that $|B_l(p_0)| \geq 2$.) 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 $|B_{l-1}(p) \cap Y_l| \leq \frac{2}{k}\cdot|B_{l-1}(p)|$ 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 $(i+1)$-NumTrans$_l(Y_l) := 0$,

   (b) **if** $Y_l \cap \bigcup_{p\notin A_l} B_l(p) = \{v\}$
   then $(i+1)$-NumTrans$_l(Y_l) := v$,

   (c) **if** $|Y_l \cap \bigcup_{p\notin A_l} B_l(p)| > 1$
   then $(i+1)$-NumTrans$_l(Y_l) :=\perp$.

Using function $(i+1)$-NumTrans, we can formally describe stage $i+1$ of the construction.

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

1. Set $l := 0$
2. Iterate $\lfloor\frac{k\log(n/4)}{8\log k}\rfloor$ times
   - $l := l+1$;
   - define set $Y_l := \{v \in R_{i+1} : \bar{\pi}(v, \hat{H}_{t_i+l-1}(v)) = 1\}$;
   - perform function $(i+1)$-NumTrans$_l(Y_l)$;
   - for any node $v \in R_{i+1}$, if $\bar{\pi}(i, \hat{H}_{t_i+l-1}(i)) = 1$ and $\bar{\pi}(v, \hat{H}_{t_i+l-1}(v)) = 0$ 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 $\bar{\pi}(i, \hat{H}_{t_i+l-1}(i)) = 0$ and there is a unique $w \in L^*_{2i-1}$ such that
       $\bar{\pi}(w, \hat{H}_{t_i+l-1}(w)) = 1$ and
       $(i+1)$-NumTrans$_l(Y_l) = 0$;
     - $(w, \hat{H}_{t_i+l-1}(w))$, if $\bar{\pi}(i, \hat{H}_{t_i+l-1}(i)) = 0$ and there is $w \in R_{i+1}$ such that
       $w =(i{+}1)$-NumTrans$_l(Y_l)$ and for every node $w' \in L^*_{2i-1}$ we have $\bar{\pi}(w', \hat{H}_{t_i+l-1}(w')) = 0$;
     - 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. Define sets L'_{2i+1} and L^*_{2i+1} as follows.
   We will refer to objects p^*, X', X^* and $B_l(p)$ defined in the last execution of function $(i+1)$-NumTrans, or existing by invariant INV (formulated and proved later). Let p^* be such that $|B_l(p^*)| \geq k\sqrt[4]{m}$ (we show later on that such p^* exists). Let $L'_{2i+1} := X'$, where X' is such that $|X' \cap B_l(p)| = 2$, for $p \neq p^*$, and $|X'| < k$. Let $L^*_{2i+1} := X^*$, where $X^* \subseteq B_l(p^*)$ is such that $|X^*| \leq k$ and X^* is a witness that the family of sets $\{Y_{l'} \cap B_l(p^*)\}_{l'=t_i+1}^{l}$ (defined for set R_{i+1}) is not a $(|B_l(p^*)|, k)$-selective family.
4. Iterate until $\bar{\pi}(i+1, \hat{H}_{t_i+l}(i+1)) = 1$

   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
     $\bar{\pi}(w, \hat{H}_{t_i+l-1}(w)) = 1$, 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.

□

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

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

We define the following invariant INV after step $l \leq \lfloor\frac{k\log(n/4)}{8\log k}\rfloor$ of stage $i+1$ of the construction:

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

LEMMA 3.1. *The invariant INV holds after every step* $l \leq \lfloor \frac{k\log(n/4)}{8\log k} \rfloor$ *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 $l < \lfloor \frac{k\log(n/4)}{8\log k} \rfloor \leq \frac{k\log m}{8\log k}$ 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 $(i+1)$-NUMTRANS 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 $l_A(p) \leq l+1$ be the number of steps in which point 2.A was applied for $p_0 = p$ during the execution of function $(i+1)$-NUMTRANS. Let $l_B \leq l+1$ be the number of steps in which point 2.B was applied during the execution of function $(i+1)$-NUMTRANS. Clearly, $\sum_{p=1}^{k/2} l_A(p) \leq \frac{k\log m}{8\log k}$ and $l_B \leq \frac{k\log m}{8\log k}$. Hence there is p^* such that $l_A(p^*) \leq \frac{\log m}{4\log k}$. This implies

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

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

We finally prove point 3 of INV after step $l+1$. Let $p^* \leq k/2$ be any integer such that $|B_l(p^*)| \geq k\sqrt[4]{m}$. Let X' be any set such that $|X'| = k-2 > 0$ and for every $p \neq p^*$, $|X' \cap B_{l+1}(p)| = 2$. Let X^* be any nonempty subset of $B_{l+1}(p^*)$ of size at most k. We show that $(X' \cup X^*) \models (i+1)$-NUMTRANS$_{l'}(Y_{l'})$, for $l' \leq l+1$. Observe that $B_{l+1}(p) \subseteq X'$ for every $p \notin A_{l+1}$, and $X' \subseteq \bigcup_{p\neq p^*} B_{l+1}(p)$, since $|X'| = 2\cdot(\frac{k}{2}-1)$, $|X' \cap B_{l+1}(p)| = 2$ for $p \neq p^*$, 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 $(X' \cup X^*) \models (i+1)$-NUMTRANS$_{l'}(Y_{l'})$, for $l' \leq l$. Indeed, since $X' \subseteq \bigcup_{p\neq p^*} B_{l+1}(p)$ 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 $p \neq p^*$. Hence the condition $|X' \cap B_{l+1}(p)| = 2$ implies $|X' \cap B_l(p)| = 2$, for every $p \neq p^*$. Additionally $X \subseteq B_{l+1}(p^*) \subseteq B_l(p^*)$, and $|B_l(p^*)| \geq k\sqrt[4]{m}$. By INV after step l we have $(X' \cup X^*) \models (i+1)$-NUMTRANS$_{l'}(Y_{l'})$, for $l' \leq l$.

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 $(i+1)$-NUMTRANS$_{l+1}(Y_{l+1}) = 0$, equals $\{v\}$ if $(i+1)$-NUMTRANS$_{l+1}(Y_{l+1}) = v$, and is of size at least two if $(i+1)$-NUMTRANS$_{l+1}(Y_{l+1}) = \perp$.

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

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

Suppose that $(i+1)$-NUMTRANS$_{l+1}(Y_{l+1}) = \perp$. Hence either point 2.A or point 2.B.c was applied in step $l+1$ during the execution of function $(i+1)$-NUMTRANS$_{l+1}(Y_{l+1})$. 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 $(X' \cup X^*) \models (i+1)$-NUMTRANS$_{l+1}(Y_{l+1})$. □

LEMMA 3.2. *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 3.1 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.

It follows that network $G_{\mathcal{A}}$ is well defined. □

The proof of the following lemma will appear in the full version of the paper.

LEMMA 3.3. *For every node* v *of* $G_{\mathcal{A}}$ *and every step* $l \leq (\frac{D}{2}-1)\cdot(\lfloor \frac{k\log(n/4)}{8\log k} \rfloor)$, *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 of Theorem 3.1. 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). 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 3.2. 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 $(\frac{D}{2}-1)\cdot\lfloor \frac{k\log(n/4)}{8\log k} \rfloor$ of algorithm $\mathcal{A}$, in view of Lemma 3.2 and Lemma 3.3. Hence algorithm $\mathcal{A}$ requires time $\Omega(n\frac{\log n}{\log(n/D)})$ to broadcast on graph $G_{\mathcal{A}}$. □

3.2 Simulating collision detection

Our algorithms in Section 3.3 use a technique of simulating collision detection which is not *a priori* available in our model. We introduced this technique in [12]. 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 $\textsc{Echo}(w, A)$

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 $\textsc{Echo}(w, A)$ 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 $\textsc{Echo}(w, R \cap S)$ 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 $\textsc{Echo}(w, R \cap S)$, described above. In case 1, a single neighbor from S is selected. In case 2, $R := \{y+1, \dots, y+(y-x+1)/2\}$. In case 3, $R := \{x, \dots, (y+x-1)/2\}$.

3.3 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 $i > 0$. 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 (waking up all neighbors and allowing them to transmit), and initiates Procedure $\textsc{Echo}(parent(v), S)$. If the token is at node 0 (after step 1), this node initiates Procedure $\textsc{Echo}(j, S)$. 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 $\textsc{Echo}(parent(v), S \cap [1, ..., 2^k])$, for $k = 1, 2, ...$, until $S \cap [1, ..., 2^k]$ is nonempty. (If $v = 0$, Procedures $\textsc{Echo}(j, S \cap [1, ..., 2^k])$ 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 3.2. *Algorithm Select-and-Send performs broadcasting in any n-node network in time $\mathcal{O}(n \log n)$.*

PROOF. The correctness of the algorithm is straightforward. 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.

3.4 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. Thus they proved 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 extension is wrong. 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.

Notice that the lower bound $\Omega(D \log n)$ from [5] holds even for the class of complete layered networks. The argument for the lower bound $\Omega(n)$ from [3], although originally using networks which are not of this class, can be modified to produce, for every broadcasting algorithm, a complete layered network of radius 3 on which this algorithm takes linear time. (This modification works only under our scenario where nodes do not know their neighborhood.) Hence our algorithm is optimal for the class of complete layered networks.

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.

THEOREM 3.3. *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 $\mathcal{O}(\log n)$. □

4. 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, Inf. Proc. Lett. 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, Distr. Comp. 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, Proc. 41st Symposium on Foundations of Computer Science (FOCS'2000), 575-581.

[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] I. Gaber and Y. Mansour, Broadcast in radio networks, Proc. 6th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'1995), 577-585.

[12] D. Kowalski and A. Pelc, Deterministic broadcasting time in radio networks of unknown topology, Proc. 43rd Ann. IEEE Symp. on Foundations of Computer Science (FOCS'2002), 63-72.

[13] D. Kowalski and A. Pelc, Faster deterministic broadcasting in ad hoc radio networks, Proc. 20th Ann. Symp. on Theoretical Aspects of Computer Science (STACS'2003), LNCS 2607, 109-120.

[14] E. Kushilevitz and Y. Mansour, An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks, SIAM J. Comp. 27 (1998), 702-712.