Available online at www.sciencedirect.com
ScienceDirect
Journal of Discrete Algorithms 5 (2007) 187–201
JOURNAL OF DISCRETE ALGORITHMS
www.elsevier.com/locate/jda

Broadcasting in geometric radio networks ☆

Anders Dessmark [a,1], Andrzej Pelc [b,*,2]

[a] *Department of Computer Science, Lund University, Box 118, S-22100 Lund, Sweden*
[b] *Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada*

Received 5 November 2004; received in revised form 28 February 2005; accepted 17 July 2006

Available online 23 August 2006

Abstract

We consider deterministic broadcasting in *geometric radio networks* (GRN) whose nodes know only a limited part of the network. Nodes of a GRN are situated in the plane and each of them is equipped with a transmitter of some range r. A signal from this node can reach all nodes at distance at most r from it but if a node u is situated within the range of two nodes transmitting simultaneously, then a collision occurs at u and u cannot get any message. Each node knows the part of the network within *knowledge radius* s from it, i.e., it knows the positions, labels and ranges of all nodes at distance at most s.

The aim of this paper is to study the impact of knowledge radius s on the time of deterministic broadcasting in a GRN with n nodes and eccentricity D of the source. Our results show sharp contrasts between the efficiency of broadcasting in geometric radio networks as compared to broadcasting in arbitrary graphs. They also show quantitatively the impact of various types of knowledge available to nodes on broadcasting time in GRN. Efficiency of broadcasting is influenced by knowledge radius, knowledge of individual positions when knowledge radius is zero, and awareness of collisions.

Keywords: Broadcasting; Knowledge radius; Radio network

1. Introduction

A radio network consists of stations (nodes) equipped with transmitting and receiving capabilities. Every node can *reach* a given subset of other nodes, depending on the power of its transmitter and on topographic characteristics of the surrounding region. Hence a radio network can be modeled by its *reachability graph* in which the existence of a directed edge (uv) means that node v can be reached from u. In this case u is called a *neighbor* of v. If the power of all transmitters is the same then any node u can reach v, if and only if, it can be reached by v, i.e., the reachability graph is symmetric.

☆ A preliminary version of this paper appeared as: Tradeoffs between knowledge and time of communication in geometric radio networks, in: Proc. 13th Ann. ACM Symposium on Parallel Algorithms and Architectures (SPAA'2001), pp. 59–66.

* Corresponding author.
E-mail addresses: andersd@cs.lth.se (A. Dessmark), pelc@uqo.ca (A. Pelc).

[1] This research was done during the stay of Anders Dessmark at the Université du Québec en Outaouais as a postdoctoral fellow.

[2] 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.

1570-8667/$ – see front matter
doi:10.1016/j.jda.2006.07.001

In the general case, the presence of obstacles can cause a situation in which a given transmitter is capable of reaching much farther in one direction than in another, thus yielding a very rich class of corresponding reachability graphs. However, in the case of an approximately flat region without large obstacles, nodes that can be reached from u are those within a circle of radius r centered at u, and the positive real r, called the *range* of u, depends on the power of the transmitter located at u. Reachability graphs corresponding to such radio networks will be called *geometric radio networks* (GRN); communication in GRN is the subject of this paper.

Nodes send messages in synchronous *rounds*. In every round every node acts either as a *transmitter* or as a *receiver*. A node gets a message in a given round, if and only if, it acts as a receiver and exactly one of its neighbors transmits in this round. The message received in this case is the one that was transmitted. If at least two neighbors of a receiving node u transmit simultaneously in a given round, none of the messages is received by u in this round. In this case we say that a *collision* occurred at u.

Broadcasting is one of the most important network communication primitives. One node of the network, called the *source*, has to transmit a message to all other nodes. Remote nodes are informed via intermediate nodes, along directed paths in the network. One of the basic performance measures of a broadcasting scheme is the total time, i.e., the number of rounds it uses to inform all the nodes of the network.

For a fixed real $s \geqslant 0$, called the *knowledge radius*, we assume that each node knows the part of the network within the circle of radius s centered at it, i.e., it knows the positions, labels and ranges of all nodes at distance at most s. We show how the size of s influences deterministic broadcasting time in GRN, and demonstrate how our results differ from those concerning broadcasting time in arbitrary graphs.

1.1. Previous work

The prevailing model in the literature on broadcasting in radio networks [1,3,9–11,21,30] is that of arbitrary undirected graphs, or equivalently, of directed symmetric reachability graphs. When deterministic broadcasting is investigated, often full knowledge of nodes concerning the network is assumed, and consequently research is focused on centralized broadcasting algorithms. The first paper to deal with this subject was [9]. Many authors [20,21,23,28] provided deterministic algorithms to produce fast broadcasting schemes given the underlying graph. The fastest such scheme has broadcast time $O(D + \log^2 n)$ [28]. On the other hand, in [3] a randomized protocol was given for arbitrary radio networks where nodes have no topological knowledge of the network, not even about neighbors. This randomized protocol runs in expected time $O(D \log n + \log^2 n)$. A deterministic centralized protocol of the same complexity has been obtained in [27]. In [30] it was shown that for any randomized broadcast protocol and parameters D and n, there exists an n-node network of diameter D requiring expected time $\Omega(D \log(n/D))$ to execute this protocol. A randomized algorithm working in optimal expected time of $O(D \log(n/D) + \log^2 n)$ has been obtained in [26] and later in [16].

To the best of our knowledge, the first paper to deal with deterministic broadcasting in radio networks without awareness of topology (also called ad hoc multi-hop radio networks) was [3]. The authors assumed that every node knows only its own label and labels of its neighbors in the graph. They constructed a class of symmetric networks of constant diameter and claimed that every deterministic broadcasting algorithm requires time $\Omega(n)$ on some networks of this class. However, as shown in [24], this result is incorrect.

In [1] the authors proved the existence of a family of n-node networks of radius 2, such that any deterministic broadcasting algorithm requires time $\Omega(\log^2 n)$ on some network of this class.

Deterministic broadcasting protocols working under the assumption that every node knows only its own label, were investigated, e.g., in [7]. The authors constructed a broadcasting scheme working in time $O(D \frac{\Delta^2}{\log^2 \Delta} \log^2 n)$, for arbitrary n-node networks with diameter D and maximum degree Δ. (While the result was stated only for undirected graphs, it is clear that it holds for arbitrary directed graphs, not just symmetric. In this case D is the eccentricity of the source, i.e., the maximum length of all shortest paths in the graph from the source to any other node.) This result was further investigated, both theoretically and using simulations, in [5,8]. On the other hand, a protocol working in time $O(D\Delta \log^{\log \Delta} n)$ was constructed in [4]. Finally, an $O(D\Delta \log^a n)$ protocol, for any $a > 2$, was described in [15]. (The exponent a can be decreased to 2 if nodes know n, and to 1 if nodes know n and Δ. The protocol works for arbitrary directed graphs.)

While the above algorithms are efficient for small values of D and Δ, their execution time becomes larger than quadratic in n, if these parameters are linear in n. Several authors investigated deterministic broadcasting algorithms whose efficiency depends only on the size n of the ad hoc radio network, and not on D or Δ. It was proved in [12] that time $O(n)$ always suffices if the network is symmetric, while for arbitrary networks the authors constructed an $O(n^{11/6})$ broadcasting algorithm working for any network. Then a series of faster algorithms were proposed, with execution times $O(n^{5/3}\log^{1/3} n)$ [17], $O(n^{3/2}\sqrt{\log n})$ [32], $O(n^{3/2})$ [13], and $O(n\log^2 n)$ [14]. The latter result has been sharpened to $O(n \log n \log D)$ in [25] and to $O(n\log^2 D)$ in [16]. In [12] the authors showed a class of networks requiring $\Omega(n\log n)$ broadcasting time. This result also follows from [6].

Broadcasting in geometric radio networks and some of their variations was considered, e.g., in [19,29,33,34]. In [34] the authors proved that scheduling optimal broadcasting is NP-hard even when restricted to such graphs, and gave an $O(n\log n)$ algorithm to schedule an optimal broadcast when nodes are situated on a line. In [33] broadcasting was considered in networks with nodes randomly placed on a line. In [29] the authors discussed fault-tolerant broadcasting in radio networks arising from regular locations of nodes on the line and in the plane, with reachability regions being squares and hexagons, rather than circles. Finally, in [19] broadcasting with restricted knowledge was considered but the authors studied only the special case of nodes situated on the line. Deterministic broadcasting with restricted knowledge for models other than radio communication was studied, e.g., in [2]. Ours is the first paper to study deterministic broadcasting in arbitrary geometric radio networks with restricted knowledge of topology.

1.2. Terminology and model description

We fix a finite set $R = \{r_1, \dots, r_\rho\}$ of positive reals such that $r_1 < \dots < r_\rho$. Reals r_i are called *ranges*. A *node* v is a triple $[l, (x, y), r_i]$, where l is a binary sequence called the *label* of v, (x, y) are coordinates of a point in the plane, called the *position* of v, and $r_i \in R$ is called the *range* of v. We assume that labels are consecutive integers 1 to n, where n is the number of nodes, but all our results hold if labels are integers in the set $\{1, \dots, M\}$, where $M \in O(n)$. Moreover, we assume that all nodes know an upper bound Γ on n, where Γ is polynomial in n. One of the nodes is distinguished and called the *source*. Any set of nodes C with a distinguished source, such that positions and labels of distinct nodes are different is called a *configuration*. With any configuration C we associate a directed graph $\mathcal{G}(C)$ defined as follows. Nodes of the graph are nodes of the configuration and a directed edge (uv) exists in the graph, if and only if the distance between u and v does not exceed the range of u. (In the entire paper the word "distance" means the geometric distance in the plane and not the distance in a graph.) In this case we say that u is a neighbor of v. Graphs of the form $\mathcal{G}(C)$ for some configuration C are called *geometric radio networks* (GRN). In what follows we consider only configurations C such that in $\mathcal{G}(C)$ there exists a directed path from the source to any other node. If the size of the set R of ranges is ρ, a resulting configuration and the corresponding GRN are called a ρ-configuration and ρ-GRN, respectively. Clearly, all 1-GRN are symmetric graphs. We denote by D the *eccentricity* of the source in a GRN, i.e., the maximum length of all shortest paths in this graph from the source to all other nodes. D is of order of the diameter if the graph is symmetric but may be much smaller in general. $\Omega(D)$ is an obvious lower bound on broadcasting time.

Given any configuration we fix a non-negative real s, called the *knowledge radius*, and assume that every node of C has initial input consisting of all nodes whose positions are at distance at most s from its own. Thus we assume that every node knows *a priori* labels, positions and ranges of all nodes within a circle of radius s centered at it. All nodes also know the set R of available ranges.

We do not assume that nodes know any global parameters of the network, such as its size or diameter. The only global information that nodes have about the network is a polynomial upper bound on its size. Consequently, the broadcast process may be finished but no node needs to be aware of this fact. Hence we adopt the same definition of broadcasting time as in [12]. An algorithm accomplishes broadcasting in t rounds, if all nodes know the source message after round t, and no messages are sent after round t.

We consider only deterministic algorithms. Nodes can transmit messages even before getting the source message, which enables preprocessing in some cases. Our algorithms are *adaptive*, i.e., nodes can schedule their actions based on their local history. A node can obviously gain knowledge from previously obtained messages. There is, however, another potential way of acquiring information during the communication process. The availability of this method depends on what happens during a collision, i.e., when u acts as a receiver and two or more neighbors of u transmit simultaneously. As mentioned above, u does not get any of the messages in this case. However, two scenarios are possible. Node u may either hear nothing (except for the background noise), or it may receive *interference noise* different from any message received properly but also different from background noise. In the first case we say that there is no *collision detection*, and in the second case—that collision detection is available (cf., e.g., [3]). A discussion justifying both scenarios can be found in [3,22].

1.3. Overview of results

We first present our results for arbitrary GRN in the model without collision detection. (Clearly all upper bounds and algorithms are valid in the model with collision detection as well.) In the case when knowledge radius exceeds the largest of all ranges, or when it exceeds the largest distance between any two nodes, we show a broadcasting algorithm working in time $O(D)$, which is optimal. In particular, this yields a centralized $O(D)$ broadcasting algorithm when global knowledge of the GRN is available. This is in sharp contrast with broadcasting in arbitrary graphs, as witnessed by the graph from [31] which has bounded diameter but requires time $\Omega(\log n)$ for broadcasting.

Next we turn attention to the case when knowledge radius $s = 0$, i.e., when every node knows only its own label, position and range. In this case we construct a broadcasting algorithm working in time $O(n)$ for arbitrary GRN. It should be stressed that our algorithm works for arbitrary GRN, not only symmetric, unlike the algorithm from [12] designed for arbitrary symmetric graphs.

Our linear time algorithm for GRN should be contrasted with the lower bound from [6,12] showing that some graphs require broadcasting time $\Omega(n \log n)$. Indeed, the graphs constructed in [6,12] and witnessing to this lower bound are not GRN. Surprisingly, we show that this sharper lower bound does not require very unusual graphs. While counterexamples from [6,12] are not GRN, it turns out that the reason for a longer broadcasting time is really not the topology of the graph but the difference in knowledge available to nodes. Recall that in GRN with knowledge radius 0 it is assumed that each node knows its own position (apart from its label and range): the upper bound $O(n)$ uses this geometric information extensively. If this knowledge is not available to nodes (i.e., each node knows only its label and range) then we show a family of GRN requiring broadcasting time $\Omega(n \log n)$. Moreover we show such GRN resulting from configurations with only 2 distinct ranges. (Obviously for 1-configurations this lower bound does not hold, as these configurations yield symmetric GRN and in [12] the authors showed an $O(n)$ algorithm working for arbitrary symmetric graphs).

The remaining part of the paper is devoted to symmetric GRN. We first consider the scenario when collision detection is available. For symmetric GRN with knowledge radius $s = 0$ (every node knows only its own label, position, and range), we show an $O(D + \log n)$ broadcasting algorithm and prove the lower bound $\Omega(\log n)$ for a family of symmetric bounded-diameter GRN. This, together with the obvious lower bound $\Omega(D)$ shows that our algorithm is asymptotically optimal under the collision detection scenario. We then show how our algorithm can be modified to work with the same complexity, even if collision detection is not available, provided that knowledge radius is positive—even arbitrarily small.

The above results show quantitatively the impact of various types of knowledge available to nodes on broadcasting time in GRN. Efficiency of broadcasting is influenced by knowledge radius, knowledge of individual positions when knowledge radius is zero, and awareness of collisions.

2. Broadcasting in arbitrary GRN without collision detection

In this section we consider the broadcasting problem in arbitrary geometric radio networks, not necessarily symmetric. We do not assume the collision detection capability.

2.1. Large knowledge radius

We first show how to broadcast in time $O(D)$ if knowledge radius s is large. We carry out the argument for knowledge radius larger than the largest range r_ρ but it also obviously holds when all nodes have global knowledge of the network.

We start by describing the following partition of the plane, known to all nodes as part of the algorithm. If $R = \{r_1, \ldots, r_\rho\}$ is the set of ranges, partition the plane into a mesh of squares of side $z = \min(r_1, s - r_\rho)/\sqrt{2}$, called *tiles*, with a corner of one of them in $(0, 0)$. Include the left and bottom side and exclude the top an right side from every square. Knowing its position, every node knows to which tile it belongs. Observe that any two nodes belonging to the same tile are within each other's range.

Next, let $x = 2\lceil \frac{r_\rho}{z} \rceil + 1$, and define a *block* to be a $x \times x$ matrix of tiles. By the choice of x, nodes in the central tile of a block cannot reach any node outside of this block. Partition the set of all tiles into a mesh of *canonical* blocks with a corner of one of them in $(0, 0)$. Fix an enumeration of tiles in one canonical block using integers 1 to x^2 and translate this enumeration to all other canonical blocks in such a way that corresponding tiles in all blocks get the same number. Call this enumeration of all tiles *canonical*. Let t be the number of the central tile of each block in this enumeration. Let σ_i, for $i \leqslant x^2$, be a (parallel) shift of the canonical enumeration in which the tile with canonical number i gets number t.

Algorithm Large-Knowledge. In the first round the source transmits the message, and the tile in which the source is contained is in state *informed*. All the other tiles are in state *uninformed*.

The rest of the algorithm works in identical consecutive phases. Each phase consists of $y = x^2(x^2 - 1)$ steps of two rounds each. The steps in each phase correspond to couples (i, j), where $i \in \{1, \dots, x^2\}$ and $j \in \{1, \dots, x^2\} \setminus \{t\}$, ordered arbitrarily.

In step (i, j) we consider enumeration σ_i and perform the following two rounds for all pairs of tiles with numbers t and j in this enumeration.

First round of step (i, j).
If tile t is in state uninformed or has been in state informed for an entire phase, do nothing. Else, if there exists a node v in tile t for which there exists a node in tile j within the range of v, pick the lowest labeled node u among such nodes v. Node u transmits the source message.

Second round of step (i, j).
If tile j is in state uninformed, the lowest labeled node v in this tile that received the message in the previous round, transmits it. Tile j enters state informed.

Lemma 2.1. *Algorithm Large-Knowledge performs broadcasting in* $\mathrm{O}(D)$ *rounds.*

Proof. First note that all nodes in a tile know the state of the tile in every round. All nodes within the range of a node of tile t are known by every other node in tile t, since the diameter of a tile is not larger than $s - r_\rho$. Hence, the set of nodes in tile t that can reach nodes in tile j is known to all nodes in tile t, and they can therefore agree on the transmitting node u. It also follows that receiving nodes in the first round of every step are able to compute the set of receiving nodes in the same tile and elect the lowest labeled among them without need of communication. Due to large distances between tiles transmitting in a particular round, no collisions occur.

Fix a node v and let T be the tile to which v belongs. Consider a step in which v first heard the source message. If the message was transmitted by a node in another tile T_1, T_1 must have entered state informed in that step at the latest. After at most y steps, T will enter state informed, since some node in T is within the range of some node in T_1. (If the message informing v originated in the same tile T, T clearly enters state informed in the same round.) Consider any tile T' containing nodes in the range of v. After at most y more steps, T' enters state informed. It follows that after at most $2y$ steps (i.e., $4y$ rounds) from the time when v gets the message, all nodes within the range of v get the message. Since y is constant, this implies that broadcasting is completed in time $\mathrm{O}(D)$. □

Lemma 2.1 together with the obvious lower bound of $\Omega(D)$ gives the following theorem.

Theorem 2.1. *The minimum time to perform broadcasting in an arbitrary GRN with source eccentricity D and knowledge radius* $s > r_\rho$ *(or with global knowledge of the network) is* $\Theta(D)$.

2.2. Knowledge radius 0

Our first result for networks with knowledge radius 0 is a $\mathrm{O}(n)$ time broadcasting algorithm working for arbitrary GRN. With knowledge radius 0, it is not possible to rely entirely on the techniques of the previous section. There are two reasons for this. First, there is no knowledge of other nodes in the same tile. This will be solved by a preprocessing step that attains knowledge radius r_1 in linear time. The second problem is that nodes of a tile no longer know which other tiles are reachable and by which nodes. We propose an algorithm that informs the entire area reachable from a tile, simply by letting every node of a tile transmit in its own round. We first formulate the algorithm assuming that the number n of nodes is known, and then we show how this assumption can be removed.

The partition of the plane into tiles and blocks is done as in Section 2.1, except for the size of the tiles that now have side $z = r_1/\sqrt{2}$. We also keep the notation from this section. In particular, we number tiles in each block using the canonical enumeration.

Algorithm Zero-Knowledge. The algorithm begins by n rounds of preprocessing where in round i, node i transmits its label and position. By definition of z, every node knows all other nodes in its tile.

After the preprocessing, every node of a tile is given a transmission number according to the label of the node as compared to the labels of other nodes in its tile. The node with the smallest label gets transmission number 1, the second smallest gets 2, and so forth. Every tile has its own transmission counter which is initially set to 1.

In the next round, the source transmits the message, and the tile containing the source is in state informed. All other tiles are in state uninformed.

The rest of the algorithm works in identical consecutive phases. Every phase consists of x^2 steps of x^2 rounds each. In step i we perform the following rounds.

First round of step i.
If tile i is in state uninformed or if the transmission counter for the tile is larger than the highest transmission number of the tile, do nothing. Else, the node of the tile with transmission number equal the transmission counter transmits the message.

Rounds number $1 < j \leqslant i$.
If tile $j-1$ is in state uninformed and if any node of the tile received a message sent in round 1 of the step, the lowest labeled node v in this tile that received the message in the previous round transmits it. Tile $j-1$ enters state informed. Increase the transmission counter of the tile by one.

Rounds number $i < j \leqslant x^2$.
These rounds are the same as above, except that they are performed for tile j instead of $j-1$.

Theorem 2.2. *Algorithm Zero-Knowledge performs broadcasting in* $O(n)$ *rounds.*

Proof. After the preprocessing, every node knows the labels and positions of every node in the same tile. This guarantees the correct assignment of transmission numbers within a tile. In the remainder of the algorithm, simultaneously transmitting nodes belong to tiles with the same number. Large distance between such tiles prevents collisions. A tile enters the state informed only after the transmission of the source message by one of the nodes in the tile, thus every node in the tile knows the message at this point. It is also clear that all nodes of a tile agree on the current value of the transmission counter. Furthermore, for rounds after the first in every step, a node that receives the message is able to compute the set of those nodes belonging to its tile that have received the message, and therefore to decide whether it is the lowest labeled such node or not.

It remains to show that every node receives the message after $O(n)$ rounds. Let $|T|$ denote the number of nodes in tile T. The number of phases between the phase when T enters state informed and the round when all tiles with a node in the range of a node of T enter state informed, is not greater than $|T|$. The number of rounds in a phase is $x^4 \in O(1)$. There is a path of length at most D from the source to any node v. This path corresponds to a sequence of tiles $T_1, \ldots, T_d$, $d \leqslant D$, containing the nodes of the path. The node v of tile T_d gets the message when T_d enters state informed, at the latest. Hence, the time required for broadcasting is $O(|T_1| + \cdots + |T_d|) \subseteq O(n)$. □

The preprocessing step of Algorithm Zero-Knowledge relies on the assumption that n is known. However, it is not difficult to modify the algorithm to work for an unknown n. Let c be a constant such that Algorithm Zero-Knowledge runs in time at most cn for n-node networks. Apply the Algorithm Zero-Knowledge in consecutive stages numbered by positive integers. Stage j lasts $c2^j$ rounds, during which the algorithm is applied with 2^j in place of n, with the following changes. A node with label larger than 2^j will not transmit during stage j. If a node transmitted the source message in some stage, it stops (and does not transmit in any subsequent stage). A tile that entered state informed remains in this state in subsequent stages. During the preprocessing, in every stage each node appends the source message (if it already knows it) to the standard message containing the label, position and range, and the tile containing this node enters state informed, if not yet in this state.

If $2^{j-1} < n \leqslant 2^j$, every node learns labels and positions of all nodes in its tile, during the preprocessing in stage j. Consequently, broadcasting is completed at the end of stage j, and no messages are sent after this stage. It follows that the entire algorithm runs in time at most $c(2^1 + 2^2 + \cdots + 2^j) \in O(n)$, even without knowing the size n of the network.

In Theorem 2.2 it is assumed (following the general definition of knowledge radius s, in this case $s = 0$) that every node knows its own label, position and range. The final result of this section shows that knowledge of positions is necessary to perform broadcasting in time $O(n)$. More precisely, we show a class of GRN for which, if every node knows only its own label and range, broadcasting time $\Omega(n \log n)$ is required. Moreover, these GRN are obtained from configurations with a two-element set R of ranges, in fact for two arbitrary distinct ranges. (Clearly, for only one range, broadcasting can always be performed in time $O(n)$ in view of [12], as such configurations yield symmetric GRN.)

Theorem 2.3. *If every node knows only its own label and range* (*and does not know its position*) *then there exist n-node GRN requiring broadcasting time $\Omega(n \log n)$.*

Proof. The proof is based on an idea from [6,12] where a class of graphs requiring broadcasting time $\Omega(n \log n)$ has been constructed. A graph described in [12] is shown in Fig. 1(a). It was shown in [12] that the source message needs time $\Omega(\log n)$ to traverse each "eye" of the graph thus yielding the lower bound $\Omega(n \log n)$ for graphs with linearly many "eyes". However, these graphs cannot be used in our case, as it is easy to see that they are not GRN. (Indeed, in order to avoid backward arrows, which is essential for the proof, we would need an unbounded number of decreasing ranges.) Likewise, the graphs from [6] are not GRN either. Hence we need to modify the graphs from [12], preserving the general pattern of a chain of "eyes" but defining them in a way that the new graphs be GRN. The modified graphs are shown in Fig. 1(b). These graphs are indeed GRN and correspond to the following 2-configuration C. Without loss of generality we define the configuration for the set of ranges $R = \{1, r\}$, where $r > 1$. The modification for an arbitrary two-element set R is immediate. We call a node *strong* if its range is r, and *weak* if its range is 1.

Consider the following set B of 6 nodes (labels will be assigned later). Strong nodes in positions $(0,0)$, $(1+\sqrt{r}, \sqrt{r})$, $(1+\sqrt{r}, -\sqrt{r})$, and weak nodes in positions $(\sqrt{r}, \sqrt{r})$, $(\sqrt{r}, -\sqrt{r})$, $(1+2\sqrt{r}, 0)$. Configuration C is the union of shifts of the set B by vectors $[(2+2\sqrt{r})i, 0]$, for $i = 0, 1, \ldots, \beta$, where $\beta \in \Omega(n)$. This configuration is shown in Fig. 1(c), for $r = 2$. The source is the node in position $(0,0)$. It is easy to see that $\mathcal{G}(C)$ is the graph shown in Fig. 1(b).

Notice that we have defined only positions and ranges of nodes, and not their labels. The class of graphs that we will use are all GRN obtained as above, with arbitrary assignments of labels to nodes. Using a modification of the argument from [12] it can be shown that for any broadcasting algorithm working for the above defined class of GRN,

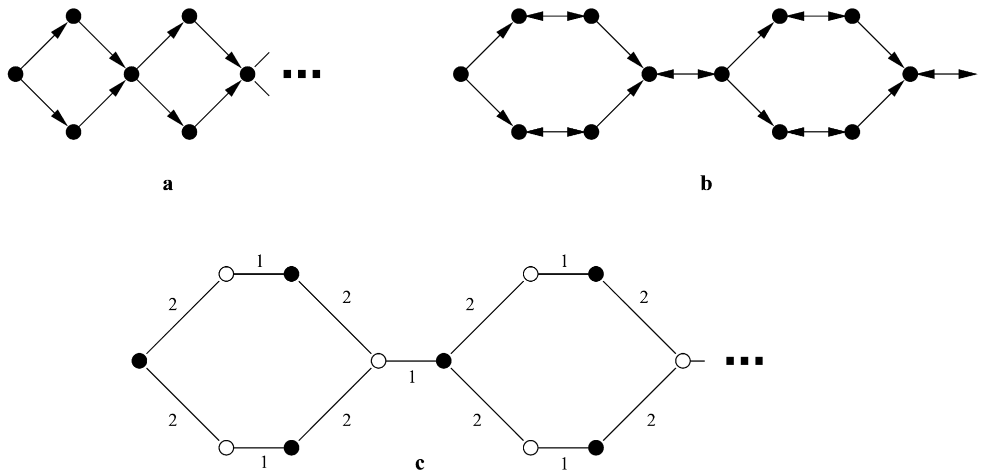

Fig. 1. (a) graph from [12] requiring broadcasting time $\Omega(n \log n)$. (b) GRN requiring broadcasting time $\Omega(n \log n)$. (c) configuration for $R = \{1, 2\}$ yielding the GRN in (b); black circles are nodes with range 2 and hollow circles are nodes with range 1. Distances between nodes are shown on joining segments.

there exists an assignment of labels such that this algorithm uses time $\Omega(n \log n)$ for broadcasting in the resulting graph. (Again it can be shown that the source message requires logarithmic time to traverse every “eye”.) □

3. Broadcasting in symmetric GRN

In this section we restrict attention to broadcasting in symmetric GRN, in both models: with and without collision detection. Under the scenario with collision detection, we obtain tight bounds $\Theta(D + \log n)$ on optimal broadcasting time for knowledge radius 0. If collision detection is not available, we show a broadcasting algorithm working with the same complexity, assuming arbitrary positive knowledge radius.

3.1. Model with collision detection

In this subsection we consider broadcasting in symmetric GRN in the model with collision detection, assuming knowledge radius 0. We show matching bounds on minimum time of broadcasting in this case. More specifically, we show a broadcasting algorithm working in time $O(D + \log n)$ for all symmetric GRN, and show a class of symmetric GRN of diameter 2 which require time $\Omega(\log n)$ for broadcasting. This lower bound, together with the obvious lower bound $\Omega(D)$ imply that the order of magnitude $O(D + \log n)$ cannot be improved in general.

We start with the description of the algorithm. It is based on an idea from [29] and consists of two parts: a preprocessing working in time $O(\log n)$ and proper broadcasting in time $O(D)$. Preprocessing is based on Procedure Elect-Couple described in [29]. Since this procedure will have to be further modified to get our result for the scenario without collision detection, and in order to make the paper self-contained, we describe below the version of the procedure suitable for our current purpose.

Let A and B be two disjoint sets of nodes such that every pair of nodes in A (resp. in B) are within each other’s range. Procedure Elect-Couple works in time $O(\log(|A| + |B|))$ and finds a pair of nodes $a \in A$ and $b \in B$ within each other’s range, if such a pair exists. (In [29] nodes were positioned on a grid and all had the same range, however some of them could be faulty. In our case positions are arbitrary and there are many possible ranges but we assume that the resulting GRN is symmetric, and this is enough for the procedure to work.)

First fix a binary partition of each of the sets A and B: Divide each of the sets into halves (or almost halves, in case of odd size), these halves into halves again, etc., down to singletons. For each member of the partition (except singletons), define the *first half* and the *second half*, in arbitrary order.

The basic step of Procedure Elect Couple (iterated $\lceil \log \Gamma \rceil$ times, where Γ is the polynomial upper bound on the size of the network, known to all nodes) is the following. Before this step, a member set X of the binary partition of A and a member set Y of the binary partition of B are *promoted*. (They contain nodes in each other’s range.) This means that all nodes in A know that nodes in X have the “promoted” status; similarly for Y and B. After this step, precisely one of the halves of X and one of the halves of Y (those containing nodes in each other’s range) are promoted, the other halves are *defeated*. This basic step works in 8 rounds as follows. We enumerate rounds in the step by integers $1, \dots, 8$ and refer to these internal round numbers.

Let X_1, X_2 and Y_1, Y_2 be the first and second halves of X and Y, respectively. For a given node we define two types of rounds. A round is LOUD for a node v, if either v acts as a transmitter in this round or v acts as a receiver and hears either a message or noise in this round. Otherwise (if v acts as a receiver and v does not hear anything in a round), the round is SILENT for v. Due to the collision detection capability, any node can tell if a given round is LOUD or SILENT for it. We use the phrase “a node transmits” in the sense “a node transmits its label”.

1. All nodes in X_1 transmit.
2. All nodes in Y_1 for which round 1 was LOUD, transmit. (If transmissions occur, Y_1 has been promoted: in this case, all nodes in B know that there are nodes $v \in Y_1$ and $w \in X_1$ in each other’s range.)
3. All nodes in Y_2 for which round 1 was LOUD but round 2 was SILENT, transmit. (If transmissions occur, Y_2 has been promoted: in this case, all nodes in B know that there are nodes $v \in Y_2$ and $w \in X_1$ in each other’s range, but there are no nodes $v' \in Y_1$ and $w' \in X_1$ in each other’s range.)
4. All nodes in X_1 for which round 2 or round 3 was LOUD, transmit. (If transmissions occur, X_1 has been promoted: in this case, all nodes in A know that there are nodes $v \in X_1$ and $w \in Y$ in each other’s range.)

5. All nodes in X_2 for which round 4 was SILENT, transmit. (If transmissions occur, all nodes in A know that there are no nodes $v \in X_1$ and $w \in Y$ in each other's range.)
6. All nodes in Y_1 for which round 5 was LOUD, transmit. (If transmissions occur, Y_1 has been promoted: similar comment as in 2.)
7. All nodes in Y_2 for which round 5 was LOUD but round 6 was SILENT, transmit. (If transmissions occur, Y_2 has been promoted: similar comment as in 3.)
8. All nodes in X_2 for which round 6 or round 7 was LOUD, transmit. (If transmissions occur, X_2 has been promoted: similar comment as in 4.)

After this basic step, exactly one of the sets X_1 or X_2, and exactly one of the sets Y_1 or Y_2, are promoted. The first iteration of the above basic step is performed with $X = A$ and $Y = B$. Hence, after at most $\lceil \log \Gamma \rceil$ iterations, exactly two nodes $a \in A$ and $b \in B$ are finally promoted (if nodes in each other's range existed in sets A and B). They are said to be *elected* and are called *partners*. They are in each other's range and they know each other's labels. All other nodes have "defeated" status, they know it, and never transmit again.

We use the partition of the plane described in Section 2.1, with the following change: the side of tiles is now $z = r_1/\sqrt{2}$.

Algorithm Elect-and-Broadcast.

1. Preprocessing
Preprocessing works in $y = x^2(x^2 - 1)$ phases, numbered 0 to $y - 1$, corresponding to couples (i, j), where $i \in \{1, \dots, x^2\}$ and $j \in \{1, \dots, x^2\} \setminus \{t\}$. In the phase corresponding to (i, j) we consider enumeration σ_i, described in Section 2.1, and carry out Procedure Elect-Couple in parallel for all pairs of tiles with numbers t and j in this enumeration. (To maintain synchronicity, we allow for each phase time $\tau = \lceil \log \Gamma \rceil \in \mathrm{O}(\log n)$, which is sufficient for all pairs of tiles. Recall that Γ is known to all nodes. The tile with number t in the enumeration corresponds to set A in the description of the basic step of Procedure Elect-Couple.) Since in this way the procedure would be performed twice for any such pair of tiles, we always skip the second execution for each pair. Nodes elected in phase $0 \leqslant p \leqslant y - 1$ are called *p-partners*.
2. Broadcasting
Divide all rounds into 2-round *steps* and enumerate the steps with consecutive integers modulo y called *tags*. The source transmits in round 1. Nodes that were not elected during preprocessing do not transmit. After an elected node heard the source message for the first time, it transmits it in the first step tagged p, such that it is a p-partner, and then stops. More precisely, it transmits in the first round of this step if its label is larger than that of its p-partner, and in the second round otherwise.

Theorem 3.1. *Algorithm Elect-and-Broadcast performs broadcasting in any n-node symmetric GRN of diameter D in time* $\mathrm{O}(D + \log n)$.

Proof. During preprocessing, Procedure Elect-Couple is performed for any pair of tiles such that one is in the block centered at the other. Executions of the procedure carried out in the same phase cannot interfere due to the large distance between respective pairs of tiles. Since y is constant and each phase takes time $\mathrm{O}(\log n)$, the entire preprocessing is completed in time $\mathrm{O}(\log n)$. Upon its completion, partners are elected for each pair of tiles containing nodes within each other's range.

The broadcasting part is carried out in such a way that no collisions occur. This is due to large distances between pairs of p-partners and to time division between p-partners. Fix a node v and let T be the tile to which v belongs. Consider a step α in which v first heard the source message. This message came from a tile T_1 such that T and T_1 contain nodes within each other's range. Consequently, p-partners $u \in T$ and $u_1 \in T_1$ were elected during preprocessing. Hence, in step $\alpha + y$ at the latest, u_1 transmits and u hears the message. In step $\alpha + 2y$ at the latest, u transmits and consequently all nodes in T get the message. Consider any tile T' containing nodes in the range of v. Let $w \in T$ and $w' \in T'$ be q-partners elected during preprocessing for this pair of tiles. In step $\alpha + 3y$ at the latest, w transmits and w' hears the message. In step $\alpha + 4y$ at the latest, w' transmits and hence all nodes in T' get the message. It follows that after at most $4y$ steps (i.e., $8y$ rounds) since the time when v gets the message, all nodes within the range of v get the message. Since y is constant, this implies that the broadcasting part is completed in time $O(D)$, thus proving time $O(D + \log n)$ for the entire algorithm. □

The final result of this subsection is the lower bound $\Omega(\log n)$ for broadcasting time, holding for some GRN of diameter 2. Together with the bound $\Omega(D)$ this matches the upper bound from Theorem 3.1.

Theorem 3.2. *For any broadcasting algorithm with collision detection and knowledge radius* 0*, there exist n-node symmetric GRN of diameter* 2 *for which this algorithm requires time $\Omega(\log n)$.*

Proof. We consider the following class $\mathcal{H}$ of symmetric GRN. Nodes of a graph in $\mathcal{H}$ are $1, \ldots, n$. Node 1 is the *source*, node n is the *sink*, and the set $\{2, \ldots, n-1\}$ is partitioned into sets X and Y, where $|Y| = 2$. Nodes $1, \ldots, n-1$ form a complete graph, and additionally, nodes from Y are connected to the sink. In order to see that such graphs are, indeed, symmetric GRN, consider the following configurations in which all nodes have range 1. Let C_1 be the circle with radius $1/10$ and center at $(1/4, 0)$ and let C_2 be the circle with radius $1/10$ and center at $(3/4, 0)$. Node 1 has position $(0, 0)$ and node n has position $(3/2, 0)$. Positions of all other nodes are in circles C_1 and C_2. Let f_i, for $i = 1, 2$, be arbitrary one-to-one functions from the set $\{2, \ldots, n-1\}$ to the circle C_i. The position of a node with label $x \in X$ is $f_1(x)$ and the position of a node with label $y \in Y$ is $f_2(y)$. It is clear that GRN corresponding to these configurations are precisely networks from the class $\mathcal{H}$.

The proof is divided into two stages. The first stage is a reduction to a restricted class of algorithms. In the second stage an adversary strategy is presented for algorithms of the restricted class. For every such algorithm the adversary constructs a network $G \in \mathcal{H}$ for which this algorithm requires time $\Omega(\log n)$ for broadcasting. In order to do this, the adversary finds the two-element set Y of the constructed network G using a halving method. This is done so that both nodes of the set Y behave in the same way during logarithmically many steps, thus producing either silence or collision at the sink in each of those steps, and hence preventing it from getting the source message.

We need to make the definition of a broadcasting algorithm more precise. In each round i, every node has an internal *state* which describes the entire information available to the node at this time. The state consists of the label of the node, of its position, of the current round number i, and of all messages heard by the node before round i. (We assume that each message is stamped with the round number in which it is sent and with the label of the sender.) If a collision was detected by the node in round i', the message corresponding to this round is the special message *col*. For a given algorithm, every node makes decisions based exclusively on its current state. Hence a broadcasting algorithm is a function which, for a given state, outputs either the bit 0 (*act as a receiver*) or a couple $(1, m)$, where the bit 1 means *act as a transmitter* and m denotes the message to be transmitted.

Stage 1. Here we show that for any broadcasting algorithm that works in time t for all networks in $\mathcal{H}$, there is another algorithm in a restricted class $\mathcal{R}$ of algorithms, that works in time $3t + 1$. The restricted class $\mathcal{R}$ is the class of broadcasting algorithms in which:

1. the source is the only node to transmit in round 1,
2. during round $3i - 1$, for positive integers i, only nodes in Y can transmit,
3. during round $3i$, for positive integers i, only nodes in X can transmit,
4. during round $3i + 1$, for positive integers i, only the source can transmit.

Denote by Δ the set of states in all broadcasting algorithms, and by $\Delta_{\mathcal{R}}$ the set of states in algorithms from class $\mathcal{R}$. We will define a function $\mathcal{K}: \Delta_{\mathcal{R}} \to \Delta$ which will be used to simulate an arbitrary algorithm A by an algorithm B from class $\mathcal{R}$. $\mathcal{K}(\delta)$, for $\delta \in \Delta_{\mathcal{R}}$, is the state of a node v in a run of algorithm A simulated by a run of B in which v is in state δ.

Let A be an arbitrary broadcasting algorithm working in t rounds for all networks in $\mathcal{H}$. We construct an algorithm B from class $\mathcal{R}$ as follows. In the first round, the source transmits the source message. All further rounds are divided into three-round *steps* numbered $1, 2, \ldots, t$. Rounds of step j are denoted $j(1)$, $j(2)$ and $j(3)$. Consider a fixed step j. (Notice that knowing the configurations yielding class $\mathcal{H}$ and knowing its own position, every node from $X \cup Y$ knows if it is in X or in Y.)

Assume that $\delta \in \Delta_{\mathcal{R}}$ is the current state of node v. If $v \in Y$, then the action of v in $j(1)$ is identical to the action of algorithm A for state $\mathcal{K}(\delta)$. If $v \in X$, then the action of v in $j(2)$ is identical to the action of algorithm A for state

$\mathcal{K}(\delta)$. If v is the source then, in $j(3)$, v transmits a concatenation of the messages received during $j(1)$ and $j(2)$ (one or both of these messages can be *col*) together with the message transmitted (if any) in algorithm A for a node in state $\mathcal{K}(\delta)$. In all other cases the node v acts as a receiver.

We now define the function $\mathcal{K}: \Delta_{\mathcal{R}} \to \Delta$. Let $\delta \in \Delta_{\mathcal{R}}$. $\mathcal{K}(\delta)$ is defined by induction on the round number i contained in δ. Assume that $\mathcal{K}(\delta')$ is already defined for all δ' corresponding to rounds $i' < i$, and that δ corresponds to round i.

The label and the position in $\mathcal{K}(\delta)$ are as in δ. The round number j in $\mathcal{K}(\delta)$ corresponds to step number j in δ. To specify the message corresponding to every round $j' < j$ in $\mathcal{K}(\delta)$, each type of node is considered separately.

Let δ' be the state of node v corresponding to a round of step j'. If node v is the sink and the action of algorithm A for state $\mathcal{K}(\delta')$ is to receive, the message corresponding to j' is *col* if a collision was detected in $j'(1)$. Otherwise the message is empty.

If node v is the source, and either the action of algorithm A for state $\mathcal{K}(\delta')$ is to transmit or if the source heard nothing in both $j'(1)$ and $j'(2)$, the message corresponding to j' is empty. Suppose that the action of algorithm A for state $\mathcal{K}(\delta')$ is to receive. If the node heard a message m in one of $j'(1)$ or $j'(2)$ and nothing in the other, the message corresponding to j' is m. Otherwise the message corresponding to j' is *col*.

If $v \in X$, and either the node acts as a transmitter in round $j'(2)$ or receives nothing in $j'(3)$, the message corresponding to j' is empty. Suppose that v acts as a receiver in round $j'(2)$. If the message received in $j'(3)$ indicates a collision then the message corresponding to j' is *col*. Otherwise, the message corresponding to j' is the single message received in $j'(3)$.

If $v \in Y$, in addition to the above the message corresponding to j' depends on the action of algorithm A for state $\mathcal{K}(\delta'')$, where δ'' is the state of the sink in step j'. If v acts as a transmitter in round $j'(1)$, the message corresponding to j' is empty. Suppose that v acts as a receiver in round $j'(1)$. If nothing is received in $j'(3)$ and the action of algorithm A for state $\mathcal{K}(\delta'')$ is to receive, the message corresponding to j' is empty. If nothing is received in $j'(3)$ and the action of algorithm A for state $\mathcal{K}(\delta'')$ is to transmit message m, the message corresponding to j' is m. If the action of algorithm A for state $\mathcal{K}(\delta'')$ is to receive and a single message m' is received in $j'(3)$, the message corresponding to j' is m'. In all other cases, the message corresponding to j' is *col*.

As the messages received by the sink can be determined knowing the messages transmitted by the source, the state δ'', and hence $\mathcal{K}(\delta'')$ is known to all nodes.

It can be proved by induction on j that if a node v is in state δ after the jth step of algorithm B then v is in state $\mathcal{K}(\delta)$ after the jth round of algorithm A. Since after the tth round of A, all nodes are informed, it follows that there is a unique node from Y transmitting in round t of algorithm A. Hence this node is the unique node from Y transmitting in algorithm B in step t, and consequently all nodes are informed after step t in algorithm B.

Stage 2. Here we describe an adversary strategy that, for any algorithm in the class $\mathcal{R}$, selects a network in $\mathcal{H}$ in which the sink does not receive the source message in the first $t = \lfloor \frac{\log n}{2} \rfloor$ steps. From this it follows that there is an adversary strategy for any broadcasting algorithm that prevents the sink from receiving the source message in the first $t = \lfloor \frac{\log n}{2} \rfloor$ rounds.

Fix an algorithm B in class $\mathcal{R}$. We build a set X' of labels that the adversary will later assign to nodes in set X of the network $G \in \mathcal{H}$. Initially the set X' is empty. Round 1 is independent of the choice of network G, and the adversary does not make any decision with respect to this round.

For round 2, we consider the set of states $\Delta_2 = \{\delta_{2,2}, \ldots, \delta_{2,n-1}\}$ such that $\delta_{2,l}$ is the state of the node with label l, positioned in Y, in round 2, with the message corresponding to round 1 being the message transmitted by the source in that round. If the set of labels l such that the action of algorithm B for state $\delta_{2,l}$ is to transmit, is larger than the set of labels l such that the action of algorithm B for state $\delta_{2,l}$ is to receive, then the adversary declares that a collision is heard by all receiving nodes, and puts all labels from the latter set in X'. If the states for which the algorithm B decides on transmitting are outnumbered by the states which result in receiving, the adversary declares that nothing is heard, and the former set is put into X'.

In round 3, we consider the set of states $\Delta_3 = \{\delta_{3,2}, \ldots, \delta_{3,n-1}\}$ such that $\delta_{3,l}$ is the state of the node with label l, positioned in X, in round 3, with the message corresponding to round 1 being the message transmitted by the source in round 1, and the message corresponding to round 2 being *col* or empty, as declared above by the adversary. The decisions of the adversary depend on the number z of states in Δ_3 for which the algorithm B decides on transmitting. If $z = 0$, the adversary declares silence for the round. If $z = 1$ and the message transmitted is m, the adversary declares that m is received by all receiving nodes apart from the sink, and adds the label corresponding to the single state to

X' if it is not yet there. If z is 2 or 3, the adversary declares that a collision is detected by all receiving nodes apart from the sink, and adds to X' the labels corresponding to the states resulting in transmissions. If $z \geqslant 4$, the adversary declares that a collision is detected by all receiving nodes apart from the sink, and adds nothing to X'.

In round 4, the action of the source is determined by the algorithm B for the state $\delta_{4,1}$ which includes the messages declared by the adversary in the previous rounds.

This completes the actions of the adversary for the initial round and the first step. The steps 2 to t are treated similarly, except for the sets Δ_{3j-1} considered in the first round of every step, which only include states corresponding to labels that are not in X'.

The set $\{2, \ldots, n-1\} \setminus X'$ of labels remaining after t steps is of size at least $\frac{n-2}{2^t} - 3t$, which is $\geqslant 2$, for sufficiently large n. Pick any two nodes of this set. The adversary network G is now a network of class $\mathcal{H}$ in which Y consists of these two nodes.

It remains to prove that if algorithm B is run on network G then the sink does not receive the source message during the first t steps. This is done by showing that the two nodes in the selected set Y either both transmit or both receive during each of the first t steps of algorithm B.

It can be verified case-by-case that the messages received by nodes of G during the run of algorithm B are precisely those that were declared by the adversary. Consider the case when algorithm B decided on transmitting for at least 4 states ($z \geqslant 4$). Then no labels were added to the set X', and hence two of the nodes corresponding to the states resulting in transmitting can be in Y. Even in this situation, at least two of these transmitting nodes remain in X, and this still results in a collision, as declared by the adversary. The remaining cases are easier and we omit the verification. The state of each node in Y, when B is run on G, is therefore identical to the state constructed by the adversary for the first round of every step. Since in each of those steps, algorithm B decided that both nodes from Y act as transmitters, or both act as receivers, this will be their behavior when B is run on G.

It follows that in each of these steps, the sink can only hear either a collision or nothing. Hence, if the source message is sufficiently long, more precisely, if its Kolmogoroff complexity is larger than t, the sink cannot get it in the first t steps. This implies the lower bound $\Omega(\log n)$ on broadcasting time. □

3.2. *Model without collision detection*

In this subsection we consider broadcasting in symmetric GRN under the model without collision detection. Our aim is to maintain complexity $O(D + \log n)$ of broadcasting. However, we need a stronger assumption concerning knowledge radius: it is no longer 0, but positive, although arbitrarily small.

Let $s > 0$ be an arbitrary positive knowledge radius. We start with the description of a broadcasting algorithm working in time $O(D + \log n)$. The algorithm is a modification of Algorithm Elect-and-Broadcast from the previous subsection. The ingredient to be changed in the algorithm is Procedure Elect-Couple. We will show how to make it work (still in time $O(\log n)$), in spite of unavailability of collision detection, using positive knowledge radius s. More precisely, for $z = \min(s, r_1)/\sqrt{2}$, consider two distinct tiles T_1 and T_2 whose side is z. By definition of z, all nodes in such a tile are in each other's range, and for any pair of nodes in a tile, one node is within distance s from the other. Hence, every node knows all other nodes in its tile. We will describe Procedure Modified-Elect-Couple which works for tiles T_1 and T_2, and accomplishes the same goal as Procedure Elect-Couple but does not need collision detection.

The idea of the modification is to simulate the capability of collision detection (*a priori* unavailable under our current scenario), using Procedure Echo from [24]. Let u be a node in tile T, and U the set of all nodes of this tile, except u. Some set U' of nodes (in or out of tile T but not containing u) transmit in a particular round, and we want all nodes in U to know if the round would be LOUD or SILENT for them (cf. the description of Procedure Elect-Couple in the previous subsection), if collision detection was available. This can be done simulating this round by the following two-round procedure (we recall the formulation and short analysis from [24] to make the paper self-contained):

Procedure Echo (u, U')**.**

1. All nodes in U' transmit their label.
2. All nodes in $U' \cup \{u\}$ transmit their label.

Of course, any node in U' knows that the simulated round would be LOUD for it, as it transmits in this round. Consider any node v in $U \setminus U'$. As mentioned in [24], three cases can occur (recall that we are in the model *without* collision detection):

Case 1. v heard a message in round 1 of Procedure Echo. Then v knows that the simulated round would be LOUD.

Case 2. v heard nothing in round 1 of Procedure Echo and heard a message (from u) in round 2. Then v knows that the simulated round would be SILENT.

Case 3. v heard nothing in either round of Procedure Echo. Then v knows that the simulated round would be LOUD (more than one node from U', within whose range v is located, transmitted in round 1).

Procedure Modified-Elect-Couple, for distinct tiles T_1 and T_2, can be now described as follows. Let a be the lowest label node in T_1 and b the lowest label node in T_2. Due to the size of tiles, all nodes in T_1 know a and all nodes in T_2 know b. Let A denote the set of nodes from T_1, other than a, and let B denote the set of nodes from T_2, other than b.

The procedure is divided into 4 parts.

Part 1

The aim of this part is to verify if a and b are in each other's range, and if so, to elect this pair. This is done in 3 rounds.

1. a transmits its label.
2. If b heard a in round 1 then b transmits its label and the message *done*.
3. If a heard b in round 2 then a transmits the message *done*.

If both a and b transmit message *done*, all nodes in T_1 and T_2 know that the pair a and b are in each other's range. These nodes are elected, and Procedure Modified-Elect-Couple for tiles T_1 and T_2 is concluded. Otherwise, Part 2 is executed.

Part 2

The aim of this part is to verify whether a and some node from B are in each other's range, and if so, to elect such a pair. $2 + 2 \cdot \lceil \log \Gamma \rceil$ rounds are reserved for this part. Fix a binary partition of B, as defined in the description of Procedure Elect Couple. For each member Y of the partition, let Y_1 and Y_2 denote the first and second half of Y, respectively. (If Y is a singleton, there is only the first half of Y, equal to Y.) In the first round of Part 2, node a transmits (as before, we use the word "transmits" to mean "transmits its label"). Let B' be the set of nodes from B that heard a in the first round. The next two rounds are used to perform Procedure Echo (b, B'). Since b and B are in the same tile, notions of a LOUD or SILENT simulated round are the same for all these nodes. In Case 2, Part 2 is concluded without success: no further transmissions are made in this part, and after the allocated time is exhausted, Part 3 starts. In Cases 1 and 3, the set B gets promoted (there is a node c in B such that a and c are in each other's range).

The rest of Part 2 is a binary search for a node in B within the range of a. If a member Y of the binary partition gets promoted, the next two rounds are used to perform Procedure Echo (b, Y_1'), where $Y_1' = Y_1 \cap B'$. In Cases 1 and 3 the set Y_1 gets promoted, while in Case 2 the set Y_2 gets promoted. Procedure Echo is iterated in this way $\lceil \log \Gamma \rceil$ times. By the end of this period (after $1 + 2 \cdot \lceil \log \Gamma \rceil$ rounds of Part 2), a single node from B transmits, i.e., Case 1 occurs in Procedure Echo. Hence all nodes in T_2 and node a know the identity of the node c in B such that a and c are in each other's range. In the final round of Part 2, node a transmits the message *done* to inform all nodes in T_1 that the couple (a, c) has been elected. Procedure Modified-Elect-Couple for tiles T_1 and T_2 is concluded.

Part 3

Part 3 is executed, if Procedure Echo (b, B') in Part 2 resulted in Case 2, i.e., when Part 2 terminates without success. The aim of Part 3 is to verify whether b and some node from A are in each other's range, and if so, to elect such a pair. This part is analogous to Part 2, with the roles of A and B, as well as of a and b, interchanged. If Part 3 is successful, this concludes Procedure Modified-Elect-Couple. Otherwise, Part 4 is executed.

Part 4

The aim of part 4 is to elect a pair of nodes from A and B, which are in each other's range (if such a pair exists). We will simulate the basic 8-round step of Procedure Elect-Couple for sets A and B, described in the previous subsection, using Procedure Echo to distinguish between LOUD and SILENT rounds. Since we need to make this distinction in both tiles, T_1 and T_2, each of the 8 rounds of the basic step is replaced by two calls of Procedure Echo. More precisely, any round in which a set Z of nodes transmits in Procedure Elect-Couple, is replaced by the following calls:

Procedure Echo (a, Z);

Procedure Echo (b, Z).

These two calls enable all nodes from A and from B to tell if the simulated round in Procedure Elect-Couple would be LOUD or SILENT. It is important to note that this distinction can be made only by nodes from A and from B but not necessarily by a and b. This is why we had to treat these nodes separately in Parts 1, 2, and 3, electing a pair containing these nodes, if possible. Now in Part 4, we can restrict search to sets A and B. As previously, this basic modified step (lasting $4 \cdot 8 = 32$ rounds) is iterated $\lceil \log \Gamma \rceil$ times, hence $32\lceil \log \Gamma \rceil$ rounds have to be reserved for Part 4.

This concludes the description of Procedure Modified-Elect-Couple for tiles T_1 and T_2. The modified version of Algorithm Elect-and-Broadcast, called Algorithm Modified-Elect-and-Broadcast, consists of replacing Procedure Elect-Couple by Procedure Modified-Elect-Couple, for the partition into tiles of side $z = \min(s, r_1)/\sqrt{2}$ instead of $r_1/\sqrt{2}$. Since the preprocessing part still takes only $\mathrm{O}(\log n)$ rounds, the analysis from the previous section gives the following result.

Theorem 3.3. *Algorithm Modified-Elect-and-Broadcast performs broadcasting in any n-node symmetric GRN of diameter D in time $\mathrm{O}(D + \log n)$, for any positive knowledge radius.*

4. Conclusion

We studied the impact of knowledge radius s on the time of deterministic broadcasting in geometric radio networks. For arbitrary GRN, under the model without collision detection, we got an optimal algorithm for large knowledge radius. On the other hand, for knowledge radius 0, we got an $\mathrm{O}(n)$-time algorithm. It remains open if this algorithm can be improved for arbitrary GRN (with eccentricity D sublinear in n) without collision detection. It would be also interesting to study trade-offs between the size of knowledge radius and the time of broadcasting, when knowledge radius is positive but small.

For symmetric GRN, we got an optimal algorithm under the model with collision detection and knowledge radius 0, working in time $\mathrm{O}(D + \log n)$. The same complexity of broadcasting can be achieved under the model without collision detection and arbitrarily small positive knowledge radius. It remains open whether this complexity still holds under the model without collision detection, when knowledge radius is 0.

Note. In a preliminary version of this paper [18] we claimed two more results. One of them was an $\mathrm{O}(D(1 + \log(n/D)))$ -time broadcasting algorithm, for arbitrary GRN with arbitrary positive knowledge radius, when collision detection is not available. However, we discovered an error in the analysis of this algorithm, and it remains open whether this result (or more generally, the possibility of $o(n)$-time broadcasting for all GRN with $D \in \mathrm{o}(n)$ and small positive knowledge radius) is true. We also claimed a lower bound $\Omega(n)$ on broadcasting time in the class of symmetric GRN of constant diameter, when knowledge radius is 0 and collision detection is not available. However, the proof of this result was based on the lower bound from [3], which was proved incorrect in [24].

References

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

[2] B. Awerbuch, O. Goldreich, D. Peleg, R. Vainish, A tradeoff between information and communication in broadcast protocols, Journal of the ACM 37 (1990) 238–256.

[3] R. Bar-Yehuda, O. Goldreich, 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, I. Chlamtac, A mobility-transparent deterministic broadcast mechanism for ad hoc networks, IEEE/ACM Transactions on Networking 7 (1999) 799–807.
[5] S. Basagni, A.D. Myers, V.R. Syrotiuk, Mobility-independent flooding for real-time multimedia applications in ad hoc networks, in: Proc. 1999 IEEE Emerging Technologies Symposium on Wireless Communications & Systems, Richardson, TX.
[6] D. Bruschi, M. Del Pinto, Lower bounds for the broadcast problem in mobile radio networks, Distributed Computing 10 (1997) 129–135.
[7] I. Chlamtac, A. Faragó, Making transmission schedule immune to topology changes in multi-hop packet radio networks, IEEE/ACM Transactions on Networking 2 (1994) 23–29.
[8] I. Chlamtac, A. Faragó, H. Zhang, Time-spread multiple access (TSMA) protocols for multihop mobile radio networks, IEEE/ACM Transactions on Networking 5 (1997) 804–812.
[9] I. Chlamtac, S. Kutten, On broadcasting in radio networks—problem analysis and protocol design, IEEE Transactions on Communications 33 (1985) 1240–1246.
[10] I. Chlamtac, S. Kutten, Tree based broadcasting in multihop radio networks, IEEE Transactions on Computers 36 (1987) 1209–1223.
[11] I. Chlamtac, O. Weinstein, The wave expansion approach to broadcasting in multihop radio networks, IEEE Transactions on Communications 39 (1991) 426–433.
[12] B.S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, W. Rytter, Deterministic broadcasting in ad hoc radio networks, Distributed Computing 15 (2002) 27–38.
[13] B.S. Chlebus, L. Gasieniec, A. Östlin, J.M. Robson, Deterministic radio broadcasting, in: Proc. 27th Int. Coll. on Automata, Languages and Programming (ICALP'2000), in: Lecture Notes in Computer Science, vol. 1853, Springer, Berlin, 2000, pp. 717–728.
[14] M. Chrobak, L. Gasieniec, W. Rytter, Fast broadcasting and gossiping in radio networks, in: Proc. 41st Symposium on Foundations of Computer Science (FOCS'2000), pp. 575–581.
[15] A.E.F. Clementi, A. Monti, R. Silvestri, Selective families, superimposed codes, and broadcasting on unknown radio networks, in: Proc. 12th Ann. ACM–SIAM Symposium on Discrete Algorithms (SODA'2001), pp. 709–718.
[16] A. Czumaj, W. Rytter, Broadcasting algorithms in radio networks with unknown topology, in: Proc. 44th Symposium on Foundations of Computer Science (FOCS'2003), pp. 492–501.
[17] G. De Marco, A. Pelc, Faster broadcasting in unknown radio networks, Information Processing Letters 79 (2001) 53–56.
[18] A. Dessmark, A. Pelc, Tradeoffs between knowledge and time of communication in geometric radio networks, in: Proc. 13th Ann. ACM Symposium on Parallel Algorithms and Architectures (SPAA'2001), pp. 59–66.
[19] K. Diks, E. Kranakis, D. Krizanc, A. Pelc, The impact of knowledge on broadcasting time in linear radio networks, Theoretical Computer Science 287 (2002) 449–471.
[20] M. Elkin, G. Kortsarz, An improved algorithm for radio broadcast, in: Proc. 16th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'2005).
[21] I. Gaber, Y. Mansour, Broadcast in radio networks, in: Proc. 6th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'1995), pp. 577–585.
[22] R. Gallager, A perspective on multiaccess channels, IEEE Transactions on Information Theory 31 (1985) 124–142.
[23] L. Gasieniec, D. Peleg, Q. Xin, Faster communication in known topology radio networks, in: Proc. 24th Annual ACM Symposium on Principles of Distributed Computing (PODC'2005), pp. 129–137.
[24] D. Kowalski, A. Pelc, Time of deterministic broadcasting in radio networks with local knowledge, SIAM Journal on Computing 33 (2004) 870–891.
[25] D. Kowalski, A. Pelc, Faster deterministic broadcasting in ad hoc radio networks, SIAM Journal on Discrete Mathematics 18 (2004) 332–346.
[26] D. Kowalski, A. Pelc, Broadcasting in undirected ad hoc radio networks, in: Proc. 22nd Ann. ACM Symposium on Principles of Distributed Computing (PODC'2003), pp. 73–82.
[27] D. Kowalski, A. Pelc, Centralized deterministic broadcasting in undirected multi-hop radio networks, in: Proc. 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'2004), in: Lecture Notes in Computer Science, vol. 3122, Springer, Berlin, 2004, pp. 171–182.
[28] D. Kowalski, A. Pelc, Optimal deterministic broadcasting in known topology radio networks, Distributed Computing, in press.
[29] E. Kranakis, D. Krizanc, A. Pelc, Fault-tolerant broadcasting in radio networks, Journal of Algorithms 39 (2001) 47–67.
[30] E. Kushilevitz, Y. Mansour, An Omega(D log (N/D)) lower bound for broadcast in radio networks, SIAM Journal on Computing 27 (1998) 702–712.
[31] A. Pelc, D. Peleg, Feasibility and complexity of broadcasting with random transmission failures, in: Proc. 24th Ann. ACM Symposium on Principles of Distributed Computing (PODC'2005), pp. 334–341.
[32] D. Peleg, Deterministic radio broadcast with no topological knowledge, Manuscript, 2000.
[33] K. Ravishankar, S. Singh, Broadcasting on $[0, L]$, Discrete Applied Mathematics 53 (1994) 299–319.
[34] A. Sen and M.L. Huson, A new model for scheduling packet radio networks, in: Proc. 15th Annual Joint Conference of the IEEE Computer and Communication Societies (IEEE INFOCOM'96), 1996, pp. 1116–1124.