On the Complexity of Distributed Broadcasting and MDS Construction in Radio Networks*

Tomasz Jurdzinski[1] and Dariusz R. Kowalski[2]

[1] Institute of Computer Science, University of Wrocław, Poland
[2] Department of Computer Science, University of Liverpool, United Kingdom

Abstract. We study two fundamental problems in the model of undirected radio networks: broadcasting and construction of a Minimal Dominating Set (MDS). The network is ad hoc, in the sense that initially nodes know only their own ID and the IDs of their neighbors. For both problems, we provide deterministic distributed algorithms working in $O(D\sqrt{n}\log^6 n)$ communication rounds, and complement them by a close lower bound $\Omega(\sqrt{Dn\log(n/D)})$, where n is the number of nodes and D is the radius of the radio network. Our work provides several novel algorithmic methods for overcoming the impact of collisions in radio networks, and shrinks the gap between the lower and the upper bounds for the considered problems from polynomial to polylogarithmic, for networks with small (polylogarithmic) radius.

Keywords: radio networks, broadcasting, minimal dominating set, distributed algorithms.

1 Introduction

Radio Networks model a communication environment where simultaneous message transmissions in a close proximity result in signal interference, and no message is successfully delivered. This model has been successfully used since early 80s in the context of Local Access Networks, wireless networks, multi-bus and multi-core topologies (c.f., [4,9]), for obtaining and analyzing many algorithmically non-trivial and applicable solutions. Even though some of them have been later analyzed in more complex models, radio networks are still widely used for their simplicity and suitability for design and (preliminary) analysis of communication algorithms.

In the radio network model, c.f., [4], the core assumption is that a transmitted message reaches all neighbors of the transmitting node v, however it could be successfully heard by a neighbor w only if w is not transmitting and v is the only transmitting neighbor of w at a time. We consider the setting *without collision detection*, i.e., the case when no neighbor transmits is indistinguishable from the case when at least two neighbors transmit. We use notation n for the number

* This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/G023018/1].

R. Baldoni, P. Flocchini, and R. Binoy (Eds.): OPODIS 2012, LNCS 7702, pp. 209–223, 2012.

of nodes in the network, D for the radius of the network (with respect to some distinguished node, called a source), and $N = \mathrm{poly}(n)$ for the range of node ids. We consider two fundamental problems: Broadcasting and construction of Minimal Dominating Set (MDS). We seek for time-efficient deterministic distributed solutions for these problems.

Previous results. Bar-Yehuda et al. [1] claimed that the time complexity of deterministic broadcasting in ad hoc radio networks is $\Omega(n)$ even for networks of radius 2. Kowalski and Pelc [10] proved that it is not the case: they showed a deterministic algorithm that accomplishes broadcast in $O(n^{2/3} \log n)$ rounds in any network of radius 2, and another algorithm that completes broadcast in $o(n)$ rounds in networks of radius $o(\log \log n)$. On the other hand, a lower bound $\Omega(Dn^{1/4})$ was proved in [10] for broadcasting in networks with radius D, which proves an exponential gap between the overhead in this model and the model with randomization, see the next paragraph. Brito and Vaya [3] improved this bound to $\Omega(n^{1/2})$, still leaving the gap between the lower bound and the best known upper bound of magnitude $n^{1/6} \log n$.

The first efficient randomized solution in the ad hoc radio model, working in expected time $O(D \log n + \log^2 n)$, was presented by Bar-Yehuda et al. [1]. A lower bound $\Omega(D \log(n/D) + \log^2 n)$ on expected time of randomized broadcast was given by Kushilevitz and Mansour [12], and the matching algorithm was developed by Kowalski and Pelc [11] and by Czumaj and Rytter [8].

The problem of constructing a Minimal Dominating Set (MDS) is closely related to Broadcasting in the model of radio networks, and many of the developed techniques and results for broadcasting also hold for MDS. In particular, we are not aware of any separate result on the time complexity of MDS in radio networks that would not be obtained in the context of broadcasting.

Our results. We strengthen the lower bound $\Omega(\sqrt{n})$ on deterministic distributed broadcasting for networks of radius 2 to $\Omega(\sqrt{n \log n})$, which justifies that the complexity of the problem is asymptotically larger than $\sqrt{n}$. For D-hop networks, the lower bound takes the form of $\Omega(\sqrt{Dn \log(n/D)})$. These bounds are easily extended to the problem of deterministic distributed construction of a MDS. We also provide two broadcasting algorithms: one for networks with radius 2, which works in $O(\sqrt{n} \log^6 n)$ communication rounds, and the other for networks of radius D, working in $O(D\sqrt{n} \log^6 n)$ rounds. The former algorithm improves over the best known $O(n^{2/3} \log n)$ time broadcasting algorithm, and thus shrinks the gap between the lower and upper bounds from polynomial to polylogarithmic for networks of radius 2. The latter algorithm extends the range of diameters admitting sublinear $o(n)$-rounds algorithms from $o(\log \log n)$ to polynomial, which is a double exponential improvement, c.f., [10]. It also shrinks the gap between upper and lower bounds from polynomial to polylogarithmic for networks of polylogarithmic radius. Finally, we show how to adapt these algorithms for constructing a MDS in asymptotically same round complexity.

Previous sublinear time deterministic algorithms for broadcasting propagated messages layer-by-layer in such a way that each node followed its own schedule, sometimes coordinated by the source. These method incurred a substantial communication overhead on each hop. We introduce more complex clustering mechanism of bipartite graphs, which allows to form collaborative groups of nodes, with the goal to inform their neighbors, c.f., Phase 3 of algorithm $\mathcal{A}_1$. We show how to efficiently build such clusters and simultaneously maintain short intra- and inter-cluster communication schedules, all in a deterministic distributed way. This clustering combined with a greedy schedule of selecting nodes with certain properties and with the centralized schedule of Chlamtac-Weinstein [5], results in substantial improvement of time complexity, especially for shallow networks (i.e., networks of small diameter). An example of novel algorithmic techniques used for efficient clustering is a new way of constructing transmission schedules, by taking a product of selectors and adaptively maintained minimum ID of cluster nodes, which results in a large portion of inter-cluster point-to-point successful communication.

Due to space limit, the missing proofs are deferred to the full version of the paper.

2 Preliminaries

We consider radio networks defined as an undirected connected reachability graph $G(V, E)$ whose nodes have distinct labels belonging to the set $[N] = \{1, \ldots, N\}$, where N is polynomially large with respect to the number of stations $n = |V|$; both n and N are known to all stations prior the computation. In the broadcasting problem, a distinguished node with label 1 is called a *source*. We define the radius D of a network as the largest distance from the source to any node of the network, where distance between nodes denotes the length of the shortest path connecting them. Initially each node has no knowledge about the topology of the underlying network, except of the information about IDs of its neighbors — we call it *local knowledge*.

It is assumed that time is divided into discrete time steps, called *rounds*, all nodes start simultaneously, they have access to the central clock, and work in rounds. A message sent at round t by a node u is sent to all its neighbors. However, a neighbor v of u receives this message if u is its only neighbor transmitting in round t. If v does not receive any message at time t, then either none of its neighbors has transmitted at round t, or at least two have. However, v is not able to distinguish between these two events; such model characteristic is typically called a model with *no collision detection*.

Communication protocol. A communication protocol specifies — for each node $v \in [N]$, the set of neighbors of node v, each round t and all messages received by node v before round t — whether node v transmits a message at round t, and if yes, what is the content of this message. The goal of any *broadcast protocol* is to deliver a message originally stored in the source, also called the *broadcast message* or the *source message*, to all nodes of the network, by transmitting and successful receptions of this message along the underlying radio network. We say that a station is *informed* at time t of an execution of a broadcasting protocol if that station received the broadcast message until round t, and it is *uninformed* otherwise. We consider a *non-spontaneous* model, i.e., a node (except the source) may act as a transmitter only if it has received a message earlier. We assume that each time a station sends a message, it encloses its ID and information containing its whole history of communication (from which one can deduce its knowledge about the network). Our algorithms, however, will use only at most polynomial, in n, number of bits, in addition to the source message.

Graph-based notation. Throughout this paper, N denotes the range of identifiers of nodes, n is the actual size of the graph of the network. Each time we refer to a symmetric graph $G(V,E)$, we mean the graph with unique identifiers in the range $[N]$ of its nodes. Given a symmetric graph $G(V,E)$, $\Gamma_G(v)$ denotes the set of neighbors of v in G, and $d_G(v) = |\Gamma_G(v)|$ (the subscript G is omitted when it is clear from the context). For a graph $G(V,E)$ with distinguished source node s, $L_i \subset V$ denotes the set of nodes in distance i from s (thus, in particular, $L_0 = \{s\}$ and L_1 is equal to the set of neighbors of s). Moreover, we denote $n_i = |L_i|$ for each $i \geq 0$. A *dominating set* in graph G is a set of nodes such that every node in the network is in this set or has a neighbor in this set. A dominating set is *minimal* if after removing any node from it the resulting set would not be dominating.

Selectors. We use combinatorial structures, called selectors, which play crucial role in many deterministic communication algorithms for radio networks. We say that a family $\mathcal{F} = (F_1, \ldots, F_f)$ of sets *hits* a set X if $|F_i \cap X| = 1$ for some $i \in [f]$. Moreover $\mathcal{F}$ *hits* X at x if $F_i \cap X = \{x\}$ for some $i \in [f]$.

Definition 1. *A family $\mathcal{F} = (F_1, \ldots, F_f)$ of subsets of $[N]$ is a (N,k,r)*-selector *if for any set $X \subseteq [N]$ of size k there is $X' \subseteq X$ of size $\min\{r+1,k\}$ such that $\mathcal{F}$ hits X at each element of X'.*

We say that f is the *size* of a family $\mathcal{F} = (F_1, \ldots, F_f)$. Several (almost) tight bounds on the size of optimal selectors have been established for various parameters, c.f., [7,2,6]. For our lower bound arguments, we need the following result.

Theorem 1. *[7] Let $\mathcal{F}$ be a $(N,k,1)$-selector, where $N > 2$ and $2 \leq k \leq n/64$. Then, $|\mathcal{F}| \geq \frac{k}{24} \log \frac{N}{k}$.*

On the other hand, we apply the following upper bound in our algorithm(s) for broadcasting in radio networks.

Theorem 2. *[2] For any integers $N \geq k \geq r \geq 1$, there exists a (N,k,r)-selector of size $O(\min(N, \frac{k^2}{k-r+1} \log \frac{N}{k}))$.*

Though the above result is only existential, efficient algorithms constructing (N,k,r)-selectors of size $O(\min(N, \frac{k^2}{k-r+1}\mathrm{polylog}(N)))$ are known as well [6].

Corollary 1. *For any integers $N \geq k \geq 1$ and a real constant $\varepsilon > 0$, there exists a $(N,k,(1-\varepsilon)k)$-selector of size $O(\min(N, k \log \frac{N}{k}))$.*

For our purposes, we need a bit stronger property defined below.

Definition 2. *A family* $\mathcal{F} = (F_1, \ldots, F_f)$ *of subsets of* $[N]$ *is a* linear $(N, k, 1-\varepsilon)$-selector *if for any set* $X \subseteq [N]$ *such that* $k/2 < |X| \le k$*, there is* $X' \subseteq X$ *of size at least* $\min(|X|, (1-\varepsilon)|X| + 1)$ *such that* $\mathcal{F}$ *hits* X *at each element of* X'*.*

Thus, on one hand, definition of linear selectors concerns only the case where $r = (1-\varepsilon)k$ in general selectors. On the other hand, we require that the property of being hit by $\mathcal{F}$ at many elements holds not only for sets of size k but for all sets of size in the range $(k/2, k]$. Using Corollary 1, one can easily prove the following statement.

Corollary 2. *For any integers* $N \ge k \ge 1$ *and a real constant* $1 > \varepsilon > 0$*, there exists a linear* $(N, k, 1-\varepsilon)$*-selector of size* $O(\min(N, k \log(N/k)))$*.*

(A, B)-broadcast protocol under known topology of graph $G(A \cup B, E)$. Let A and B be disjoint subsets of V such that all nodes in A have the same message M. Then a protocol which makes message M known to all nodes $v \in B$ having a neighbor in A is called *(A, B)-broadcast protocol.*

Theorem 3. *[5] Let a radio network be modeled by a graph* $G(V, E)$*, where IDs of stations belong to* $[N]$*, and let* $A, B \subset V$ *be such that* $A \cap B = \emptyset$*, all nodes in* A *have the same message* M *and they* know *the topology of the subgraph of* G *spanned on* $A \cup B$ *(i.e., the graph* $G(A \cup B, E \cap (A \cup B)^2)$*). Then the elements of* A *can compute* (A, B)*-broadcast protocol that informs all nodes in* B *in time* $O(\log^2 N)$*.*

Communication schedules. An (oblivious) *communication schedule* of length f is a family of sets $\mathcal{S} = (S_1, \ldots, S_f)$, where $S_i \subseteq [N]$ for every $i \in [f]$. The length of such communication schedule is denoted by $|\mathcal{S}| = f$. An *execution* of the communication schedule $\mathcal{S}$ is a protocol in which station v transmits in round j iff $v \in S_j$. An *execution* of the communication schedule $\mathcal{S} = (S_1, \ldots, S_f)$ for r rounds is a communication protocol in which station v transmits in round $j \in [r]$ iff $v \in S_{1+(j-1) \mod f}$, i.e., we apply the communication schedule which consists of consecutive repetitions of $\mathcal{S}$. An *execution* of the communication schedule $\mathcal{S}$ *on the set* X (for r rounds) is a protocol in which station v transmits in round $j \in [|S|]$ (resp., $j \in [r]$) iff $v \in X \cap S_j$ (resp., $v \in X \cap S_{1+(j-1 \mod |S|)}$).

3 Lower Bound

In this section we prove a lower bound $\Omega(\sqrt{n \log n})$ for deterministic broadcasting with local knowledge on networks with radius 2, and its generalized version $\Omega(\sqrt{Dn \log(n/D)})$ for network of radius D.

Theorem 4. *Every deterministic broadcasting protocol for networks of radius 2 works in time* $\Omega(\sqrt{n \log n})$*.*

The idea of the proof is as follows.[1] Consider a class of networks of radius 2, in which nodes in the middle layer are not connected among themselves and are conceptually partitioned into groups. Some groups are connected to a single node in the last layer, some do not have any neighbor in the last layer. Assume that the size of a single group is around k, for some $k \leq n$. In order to choose a successful transmitter from a random group (to inform their unique neighbor in the last layer) without help of the source, a lower bound $\Omega(k \log(N/k))$ applies, c.f., [7]. On the other hand, there are $\Theta(n/k)$ groups, and intuitively the source could not help all the groups (by speeding-up the process of obtaining a successful transmission) in time asymptotically smaller than n/k, provided it can help one group at a time. We show formally that no other faster scenario could happen except the combination of the two described above. Therefore, all nodes in the last layer obtain the source message in time asymptotically not smaller than $\min_{k \leq n} \max\{k \log(N/k), n/k\}$, which is $\Omega(\sqrt{n \log n})$ for $k = \sqrt{n/\log n}$. One could concatenate the above construction and repeat the arguments $\Theta(D)$ times, by putting the source node of the next radius 2 component in the last informed node of the previously built part of the network. Here, network layers have size $\Theta(n/D)$, and optimal parameter k should be set to $k = \sqrt{n/(D \log(n/D))}$, in order to get broadcasting time of $\Omega(D\sqrt{(n/D)\log(n/D)}) = \Omega(\sqrt{nD \log(n/D)})$.

Corollary 3. *Every deterministic protocol requires $\Omega(\sqrt{Dn \log(n/D)})$ rounds to accomplish broadcast on networks with local knowledge and radius D.*

4 Broadcasting Algorithm in Networks of Radius 2

In this section we develop algorithm $\mathcal{A}_1$, whose complexity differs from the lower bound by only a polylogarithmic multiplicative factor. It will also be a subroutine for the protocol broadcasting in networks of any radius, c.f., Section 5.

4.1 Description of Algorithm $\mathcal{A}_1$

Testing and election subroutines. First we define two auxiliary problems for a radio network $G(V, E)$ with distinguished source s, where each station knows its neighbors. Recall that L_k denotes the set of nodes at distance k from the source. Assume that a set of stations $A \subseteq L_k$ is defined such that each station has a unique key in range $[R]$, for some R such that $\log R = O(\log n)$, and it knows whether it belongs to A. However, no station knows which other stations belong to A. The *k-layer emptiness testing* problem is to learn whether A is empty, that is, all nodes in $\bigcup_{i=0}^{k} L_i$ should know at the end of the protocol whether $A = \emptyset$. The *k-layer election* problem is to decide whether A is empty and, if $A \neq \emptyset$, to choose the element in A with the largest value of the key. That is, all nodes in $\bigcup_{i=0}^{k} L_i$ should know at the end of the protocol either that $A = \emptyset$ or the ID of the element of A with the largest key.

[1] Although the general framework of the proof is similar to the one in [3], we analyze slightly different class of networks to obtain an additional factor $\sqrt{\log n}$ in the lower bound formula.

Theorem 5. *[10] Consider a symmetric radio network with distinguished source node s and with no collision detection where each station knows its neighbors. Then,*

1. *there exists a protocol of time $O(1)$ solving 1-layer emptiness testing;*
2. *there exists a protocol of time $O(\log n)$ solving 1-layer election problem.*

Introduction to algorithm $\mathcal{A}_1$. Below, we present an algorithm $\mathcal{A}_1$ broadcasting in networks of radius 2. It consists of four *Phases.* Each time we check in algorithm $\mathcal{A}_1$ whether a subset of L_1 is empty or we choose an element of this subset, the appropriate protocol for 1-layer emptiness or 1-layer election from Theorem 5 is applied. Notice that, when one node v from L_1 is chosen, it can pass any message M to all elements of $L_1 \cup \{s\}$ in two rounds: first v sends this message to the source s, then s sends M to all elements of L_1.

During execution of algorithm $\mathcal{A}_1$ we conceptually *delete*, or *remove*, some nodes from the network, which means that these nodes are switched off (i.e., become idle) in the following parts of the algorithm. Therefore, all references to the network graph, layers L_1, L_2 and to the sets of neighbors of nodes (i.e., to $\Gamma(v)$ and $d(v)$, for a station v) in the following description of the algorithm will be made with respect to the values of these parameters after removal of *deleted* nodes and edges adjacent to them from the network reachability graph. Each time we will remove nodes from the network during Phases 1 and 4 of algorithm $\mathcal{A}_1$, all nodes in L_1, as well as the source s, will be aware of this fact and will send this information in their messages. However, in general, it is sufficient that non-removed neighbors of a removed node v know about the deletion of v (this issue becomes nontrivial in Phase 3).

High-level description of $\mathcal{A}_1$. The idea of the algorithm is as follows. We gradually decrease the size of the network graph by removing some nodes from it, i.e., by deciding that some nodes remain idle and do not participate in the further part of the algorithm; each station is aware whether it is removed or not. However, an invariant will be maintained that a node from L_1 can be removed only when all its neighbors in L_2 are informed, and a node from L_2 can be removed only when it is informed already. Next we describe Phases 1-4.

Phase 1. Using 1-layer election we first eliminate all nodes from L_1 that have at least $\sqrt{n}$ neighbors in L_2. More precisely, we delete some nodes from L_1, together with their neighbors in L_2, such that in the resulted graph (i.e., after these deletions), no node in L_1 has more than $\sqrt{n}$ neighbors in L_2. Since each such node eliminates at least $\sqrt{n}$ nodes from the graph, and since it can be chosen in $O(\log n)$ rounds (see Theorem 5), Phase 1 requires $O(\sqrt{n}\log n)$ rounds. Moreover, thanks to connection to the source, all stations from L_1 are aware of the deleted nodes, and therefore they know their neighborhood in the remaining network graph.

Phase 2. When there are no more nodes in L_1 with at least $\sqrt{n}$ (remaining) neighbors in L_2, we cannot continue choosing the remaining nodes in L_1 sequentially (to inform their neighbors in L_2), since this might require $\omega(\sqrt{n})$ rounds.

Instead, some nodes in L_2 can be informed in parallel. To this aim, we execute a sequence of linear $(N, 2^i, 1/2)$-selectors, for consecutive $i = 0, 1, \ldots, (1/2)\log n$, on nodes in L_1, which ensures that all stations from L_2 of degree at most $\sqrt{n}$ are informed (Phase 2), c.f., Corollary 2. Indeed, if X is a set of neighbors of a node v and $2^{i-1} < |X| \leq 2^i$, then at least half of neighbors of v will be heard by v during the execution of $(N, 2^i, \frac{1}{2})$-selector. Hence, the degrees of all stations from L_2 which were not informed are larger than $\sqrt{n}$ after Phase 2.

Phase 3. If stations from L_1 knew which of their neighbors are *not informed*, we could choose sequentially (as we will do later in Phase 4) stations from L_1 with the largest number of uninformed neighbors in L_2 and remove them from the graph together with their neighbors. Such a process would inform all stations in $O(\sqrt{n}\log n)$ rounds, since we can benefit from the fact that removed stations from L_2 "eliminate" many edges of the graph (recall that their degrees are larger than $\sqrt{n}$).

Unfortunately, we do not know whether the task of acquiring such a knowledge by the considered stations in L_1 is feasible in $O(\sqrt{n}$ polylog$(n))$ rounds. However, in Phase 3 we design a protocol which achieves similar goal with slightly relaxed knowledge requirements. Namely, we require that in the sub-network remaining after Phase 3, the nodes in $L_1 \cup L_2$ with degree smaller than $\sqrt{n}$ constitute only small isolated connected components (here by small we understand $O(\sqrt{n})$) and each station knows its whole component. This gives stations a knowledge about uninformed neighbors in L_2 and will allow informing all uninformed nodes in L_2 (i.e., those with degrees at least $\sqrt{n}$) in $O(\sqrt{n}\log n)$ rounds later in Phase 4, by using a greedy process similar to the one in Phase 1.

In order to trim the network graph to obtain the desired property at the end of Phase 3, we keep building a specific clustering allowing efficient propagation of knowledge inside each cluster, and simultaneously we uncover nodes that gather large information about its surrounding (i.e., information about $\Omega(\sqrt{n})$ remaining nodes that are reachable through the intra-cluster communication in $O(\sqrt{n}$ polylog$(n))$ rounds). The uncovered node delivers the information about its surrounding to all nodes in L_1 via the source, and thus the nodes in this surrounding also become uncovered. In the process of building the clustering, we keep joining clusters in a way guarantying fast intra-cluster communication, until they become big (and then uncovered) or isolated. Then, at the end of Phase 3, a short $O(\log^2 n)$ broadcasting schedule is designed locally for all nodes in L_1 uncovered in Phase 3, so that they can successfully inform all their neighbors in L_2, among which some may be still not informed. (This follows from the fact that some nodes in L_1 are uncovered by another member of their clusters, as a part of its surrounding, so they might not have had an opportunity to transmit successfully.) The details of Phase 3 include several novel algorithmic techniques and synchronization between them, and therefore they are deferred to the full version of the paper. Below we describe a high-level idea of how the clusters are joined and how uncovering is done.

Initially each node participating in Phase 3 constitutes a single cluster. Suppose we are given a partition of participating nodes into connected clusters, each of them is not big and provides intra-cluster communication schedule that allows exchanging point-to-point messages between any two nodes v, w in the cluster in $O(\sum_{i \le k} d(v_i)\ \mathrm{polylog}(n))$ rounds, where $v = v_1, \ldots, v_k = w$ is a path between v and w in the cluster. It can be argued that any two nodes in the cluster can therefore communicate in $O(\sqrt{n}\ \mathrm{polylog}(n))$ rounds. Consider a single node in a cluster. It learns the minimum ID of nodes in its cluster in $O(\sqrt{n}\ \mathrm{polylog}(n))$ rounds, and then it locally computes the product of its selector schedule and the minimum ID. More precisely, the local transmission schedule of a node is defined as follows: whenever the node belongs to the currently considered set in the selector family, it performs a sequence of silences/transmissions corresponding to the 0-1 representation of the hold minimum ID; otherwise it stays idle for $\log N$ rounds. It can be shown that when using the obtained schedules, several clusters exchange messages and join into bigger clusters, in $O(\sqrt{n}\ \mathrm{polylog}(n))$ rounds. This is because selectors combined with the minimum IDs of the clusters (to which nodes belong) assure that a constant fraction of *inter-cluster* edges will propagate a message successfully. After joining into bigger clusters, nodes interleave their previous intra-cluster schedules with the newly computed ones, which, as we will show, preserves the required property of fast intra-cluster communication with respect to the new clusters. This invariant assures that every such joining operation lasts $O(\sqrt{n}\ \mathrm{polylog}(n))$ rounds. Because after each of them a constant fraction of inter-cluster edges become intra-cluster edges, this process can be continued no more than $\log m = O(\log n)$ times, where m is the number of edges in the graph. This gives $O(\sqrt{n}\ \mathrm{polylog}(n))$ bound on the length of joining processes in Phase 3.

The above joining process can be applied only to small clusters. Therefore, once a surrounding of a node in L_1 becomes big (i.e., the cluster itself has become big after the last merge), it participates in the process of electing nodes in its cluster such that each of them will cover $\Omega(\sqrt{n})$ remaining nodes in the network (we say that a node v *covers* other uncovered node if v has knowledge that this node belongs to the network and it knows some edge adjacent to it). This is done through the source by using election procedure, c.f., Theorem 5. After that the uncovered parts of the network (which, as we will show, cover all newly created big clusters), are conceptually removed from the graph of participating nodes, and the joining process described above can be resumed with respect to the remaining small clusters. The process of uncovering components takes $O(\sqrt{n}\ \mathrm{polylog}(n))$ rounds in total, by arguments similar to the one used for Phase 1.

At the end of Phase 3, the remaining nodes switch to Phase 4, while the nodes in L_1 that have been uncovered (together with their neighbors) in Phase 3 compute a short $O(\log^2 n)$ broadcast schedule to inform all their neighbors. For this purpose, a centralized algorithm from [5] is applied, as all nodes in L_1 share the same knowledge about uncovered nodes. All together: joining clusters, uncovering components and final broadcast schedule, take $O(\sqrt{n}\ \mathrm{polylog}(n))$ rounds.

Phase 4. The source sequentially elects elements of L_1 with largest remaining neighborhoods.

The structure of Algorithm $\mathcal{A}_1$ is as follows:

Algorithm $\mathcal{A}_1$

Phase 1
While the set $X = \{v \mid v \in L_1 \text{ and } d(v) \geq \sqrt{n}\}$ is not empty:
1. choose $v \in X$ with the largest ID, using the protocol for 1-layer election;
2. v transmits a message and informs L_1 about $\Gamma(v)$ via the source;
3. remove $(\Gamma(v) \cap L_2) \cup \{v\}$ from the graph.

Phase 2
Execute the sequence of linear $(N, 2^i, \frac{1}{2})$-selectors, for consecutive $i = 0, 1, \ldots, \frac{\log n}{2}$ on nodes of L_1.

Phase 3
This phase removes some number of nodes from L_1 and L_2. As the result, we obtain the network with properties (a)–(d) specified in Lemma 1.

Phase 4
While $X = \{v \mid v \in L_1 \text{ and } \Gamma(v) \cap L_2 \neq \emptyset\}$ is not empty:
1. choose $v \in \{x \in L_1 \mid \ |\Gamma(x) \cap L_2| = \max_{w \in L_1} |\Gamma(w) \cap L_2|\}$ with the largest ID, using the protocol for 1-layer election and IDs $(|\Gamma(x) \cap L_2|, x)$ with lexicographic ordering;
2. v transmits a message and informs L_1 about $\Gamma(v)$ via the source;
3. remove $(\Gamma(v) \cap L_2) \cup \{v\}$ from the graph.

4.2 Analysis of Algorithm $\mathcal{A}_1$

Properties of Phases 1 and 2 are quite straightforward, therefore we will state them later in the proof of the final theorem. Now we focus on the properties of Phase 3, and based on them we analyze the complexity of Phase 4.

Lemma 1. *Time complexity of Phase 3 is $O(\sqrt{n} \log^6 n)$. Moreover, the graph $G(V, E)$ corresponding to the network at the end of Phase 3 satisfies:*

(a) $\Gamma_G(v) \leq \sqrt{n}$ for each $v \in L_1$;
(b) $\Gamma_G(v) > \sqrt{n}$ for each $v \in L_2$;
(c) each station $v \in L_1$ knows IDs of its neighbors from L_2 in G;
(d) each station deleted from the network is informed.

Using the properties stated in Lemma 1 we can analyze time complexity of Phase 4. Let $\mathcal{E}_1(n)$ be maximum of time complexities of 1-layer emptiness testing and 1-layer election problem. Although $\mathcal{E}_1(n) = O(\log n)$ according to Theorem 5, we present complexity analysis of $\mathcal{A}_1$ explicitly specifying the number of executions of election and emptiness testing, since we will apply this result for broadcasting in networks with larger diameter.

Proposition 1. *All elements of L_2 become informed after at most $(2\frac{n_1}{\sqrt{n}} + 1)\log n$ executions of steps $1-3$ of Phase 4, where $n_1 = |L_1|$. That is, time complexity of Phase 4 is $O((\frac{n_1}{\sqrt{n}} + 1) \cdot \mathcal{E}_1(n) \log n)$.*

Theorem 6. *The algorithm $\mathcal{A}_1$ performs broadcasting in radio networks of radius 2 in time $O(\sqrt{n}\log^6 n + \mathcal{E}_1(n) \cdot \frac{n_1+n_2}{\sqrt{n}} \cdot \log n) = O(\sqrt{n}\log^6 n)$.*

Proof. (Sketch) Since the above claimed time complexity of $\mathcal{A}_1$ corresponds to the time complexity of Phase 3 stated in Lemma 1, it remains to analyze Phases 1, 2 and 4. Time of Phase 2 is $O(\sum_{i=1}^{(\log n)/2} 2^i \log n) = O(\sqrt{n}\log n)$, according to Corollary 2. Phase 1 consists of at most $1+n_2/\sqrt{n}$ calls of the election procedure, where $n_2 = |L_2|$, since each execution of the election (but the last one) deletes at least $\sqrt{n}$ stations from L_2. Finally, time complexity of Phase 4 is $O(\mathcal{E}_1(n) \cdot \log n \cdot \frac{n_1}{\sqrt{n}})$, as stated in Proposition 1.

As for correctness of Algorithm $\mathcal{A}_1$, it follows from Lemma 1 and the fact that a node $v \in L_1$ is deleted in Phase 1 or Phase 4 only when all its neighbors are informed, while a node $v \in L_2$ is deleted only when it is informed. □

Finally, we make an additional observation, which will be useful for designing an extension of protocol $\mathcal{A}_1$ to multi-hop networks.

Corollary 4. *After execution of $\mathcal{A}_1$, the stations from L_1 can build an (L_1, L_2)-broadcast protocol working in time $O(\log^2 n)$.*

Proof. (Sketch) All nodes from L_1 can compute an (L_1', L_2')-broadcast protocol S_1 of required size, where $L_1' \subseteq L_1$ and $L_2' \subseteq L_2$ are the nodes uncovered in Phases 1, 3 and 4 (c.f., Theorem 3). The graph spanned on all remaining nodes can be partitioned into connected components such that there are no edges between these connected components in the original network, and each node v knows its whole connected component $G(v)$; it follows from the structure of Phase 3, that only small components that cannot merge into bigger ones remain at the end of this phase. Therefore, each node v can compute a $(L_1 \cap G(v), L_2 \cap G(v))$-broadcast protocol. Since there are no edges between the components, the schedules for all components can be executed simultaneously without causing additional collisions, forming a new protocol S_2. Concatenation of S_1 and S_2 gives a (L_1, L_2)-broadcast protocol working in time $O(\log^2 n)$. □

5 Broadcasting in Networks with Any Radius $1 \le D \le n$

In this section we describe a deterministic algorithm accomplishing broadcast in time $O(D\sqrt{n}\log^6 n)$ on any network of radius D. The algorithm work in stages. After the kth stage of the algorithm, for $k \in [D]$, where D is the radius of the network, the following properties will be satisfied:

(P1) All nodes from $\bigcup_{i=0}^{k} L_i$ are informed and each node $v \in L_i$, for $i \le k$, knows its layer i.

(P2) For each $i \in [k-1]$, the protocol $\textsc{Send}_i$ is constructed, which performs (L_{i-1}, L_i)-broadcast in time $O(\log^2 n)$, i.e., if all nodes in L_i have the same message M, the protocol $\textsc{Send}_i$ makes M known to all nodes of L_{i+1} in time $O(\log^2 n)$.

(P3) For each $i \in [k-1]$, the protocol $\textsc{Test}_i$ is constructed which solves the i-layer emptiness testing problem in time $O(i \log^2 n)$.

The term "protocol is constructed" means here that each node knows its schedule in some protocol solving the appropriate communication problem.

Observe that, after application of Algorithm $\mathcal{A}_1$, the above statements are satisfied for $k = 2$ (i.e., (P1) follows from Theorem 6 and (P2) follows from Corollary 4, and (P3) follows from Theorem 5). Assume that the above properties are satisfied for $k \geq 2$. First, we would like to show how the protocol $\textsc{Test}_k$ can be build without any communication in the network, assuming $\textsc{Send}_i$ and $\textsc{Test}_i$ are known for $i < k$. Below, we assume that $A \subseteq L_k$ is the set of stations for which we test emptiness.

Procedure $\textsc{Test}_k(A)$

1: nodes from L_{k-2} execute protocol $\textsc{Send}_{k-2}$ with the same (arbitrary) message M_1; at the same time, each element of $A \subseteq L_k$ sends a message M_2 in each of $|\textsc{Send}_{k-2}|$ rounds different from M_1, where $|\textsc{Send}_{k-2}|$ denotes the time of $\textsc{Send}_{k-2}$;
2: each station $v \in L_{k-1}$ which could not hear a message M_1 from L_{k-2} in the preceding $|\textsc{Send}_{k-2}|$ rounds belongs to the set A';
3: execute $\textsc{Test}_{k-1}(A')$, let R be the result of this execution known to all elements of L_{k-1};
4: execute $\textsc{Send}_{k-1}$ with the message R.

Assume that time of $\textsc{Send}_i$ is at most $c_1 \log^2 n$ and time of $\textsc{Test}_i$ is at most $c_2 i \log^2 n$ for each $i < k$ and $c_2 > 2c_1$. Then, time of the above protocol is at most $2c_1 \log^2 n + c_2(k-1)\log^2 n < c_2 k \log^2 n$ which shows that time of $\textsc{Test}_k$ is $O(k \log^2 n)$.

Procedure $\textsc{Elect}_k$. Using the protocol $\textsc{Test}_k$, one can build a protocol $\textsc{Elect}_k$ solving the kth layer election problem, i.e., chooses an element of $A \subseteq L_k$ with the largest key (keys are polynomial wrt n), provided A is not empty. Such a protocol requires $\log n$ execution of $\textsc{Test}_k$, since it gradually decreases A using binary selection. Therefore, the complexity of protocol $\textsc{Elect}_k$ is $O(k \log^3 n)$.

Procedure $\textsc{Inform}_k$. Algorithm $\mathcal{A}_1$ relies on the fact that all elements of L_1 are connected to the source and therefore, once an element $v \in L_1$ is elected, it can pass any message M to all elements of L_1 in two rounds (through the source). We need a counterpart of this possibility in the case when a node $v \in L_k$ for $k > 1$ wants to pass a message M to all other elements of L_k. Such a message can be first sent to the source in $k-1$ rounds in the following way. Assume that each station v stores $prec(v)$, id of the station which informed v. In order to send a message from $v_0 \in L_k$ to s in k rounds, $v_i = prec(v_{i-1})$ sends a message from

L_{k-i+1} to L_{k-i} in the ith round, for $i \in [k]$. Then, the message is transmitted from the source to L_k by the application of $\textsc{Send}_0, \textsc{Send}_1, \ldots, \textsc{Send}_{k-1}$. We call such a protocol $\textsc{Inform}_k$. Note that its time complexity is $O(k \log^2 n)$ by (P2).

Algorithm $\mathcal{A}_k$*.* Equipped with the protocols $\textsc{Send}_k$, $\textsc{Elect}_k$ and $\textsc{Inform}_k$, we are ready to transmit the broadcasted message from L_k to L_{k+1}. Namely, we mimic the algorithm $\mathcal{A}_1$ in the following way:

(a) nodes from L_k work as the elements of L_1 in $\mathcal{A}_1$;
(b) newly informed nodes and stations in $L_k \cup L_{k-1}$ work as the elements of L_2 in $\mathcal{A}_1$ (nodes informed during this execution, which do not belong to L_{k-1}, learn that they belong to L_{k+1});
(c) each time emptiness of some subset of L_1 should be checked in $\mathcal{A}_1$, the procedure $\textsc{Test}_k$ is applied;
(d) each time an element from some subset of L_1 should be chosen in $\mathcal{A}_1$, the procedure $\textsc{Elect}_k$ is applied;
(e) each time a message M from $v \in L_1$ should be transmitted through the source to the whole L_1, the procedure $\textsc{Inform}_k$ is used.

One subtle issue is that our presentation of Algorithm $\mathcal{A}_1$ utilized the fact that nodes in layer L_1 know which of their neighbors are in which layer. A corresponding property may not be true after moving to the next layers. Therefore, in order to apply algorithm $\mathcal{A}_1$, after the adaptation described in the above items (a)–(e), for propagating the broadcast message from L_k to L_{k+1}, a few more subtle technical fixes in Phase 3 are needed (they do not, however, change the general structure of the algorithm and its analysis). Let $\mathcal{A}_k$ denote algorithm $\mathcal{A}_1$ modified as described in (a)–(e).

Procedure $\textsc{Send}_k$. It can be argued that the knowledge about the nodes collected during the execution of $\mathcal{A}_1$ is sufficient for designing a (L_1, L_2)-broadcast protocol of size $O(\log^2 n)$ (c.f., Corollary 4). This property generalizes to $\mathcal{A}_k$, since the information acquired by L_k about L_{k+1} corresponds to the information about L_2 known to L_1 during the execution of $\mathcal{A}_1$. That is, the nodes in L_k can build a (L_k, L_{k+1})-broadcast protocol $\textsc{Send}_k$ of size $O(\log^2 n)$ after the execution of $\mathcal{A}_k$. Thus, (P1)–(P3) are satisfied after the execution of $\mathcal{A}_k$. Based on the constructions of $\mathcal{A}_k, \textsc{Test}_k, \textsc{Send}_k$, and $\textsc{Elect}_k$, we obtain the following broadcast algorithm $\mathcal{B}$:

Algorithm $\mathcal{B}$

```
1: The source sends the broadcasted message.
2: for k = 2, 3, ... do
3:     Execute A_k;
4:     Build TEST_k, SEND_k, and ELECT_k;
5:     Execute TEST_k(L_k) in order to check whether L_k is empty;
6:     If L_k is empty, finish the algorithm.
```

Let us stress here that deletion of nodes in phases 1–4 of $\mathcal{A}_k$ applies only to the execution of $\mathcal{A}_k$ — the deleted nodes are restored after that.

Theorem 7. *Algorithm $\mathcal{B}$ completes broadcasting in time $O(D\sqrt{n}\log^6 n)$ in any n-node radio network of radius D.*

Proof. (Sketch) The above discussion justifies the fact that properties (P1)-(P3) are satisfied in consecutive stages defined by the for loop of algorithm $\mathcal{B}$. Moreover, $\textsc{Elect}_k$ works in time $O(k\log^3 n)$ for each k, as discussed earlier. Therefore, the number of rounds in the kth stage of the algorithm is

$$O\left(\sqrt{n}\log^6 n + (k\log^3 n)\cdot\frac{n_{k-1}+n_k+n_{k+1}}{\sqrt{n}}\right) ,$$

due to Theorem 6 (recall that nodes from L_{k-1} and L_{k+1} play the role of L_2 in the execution of $\mathcal{A}_k$).

The test of emptiness of L_k in line 6 guarantees that the algorithm finishes its work only after informing all nodes in the Dth layer, where D is the radius of the network (recall that, after execution of $\textsc{Test}_k$ on the set A, all elements of $\bigcup_{i=1}^{k} L_i$ know the result of the test).

Observe that each execution of $\textsc{Inform}_k$ in Algorithm $\mathcal{A}_k$ (e.g., in step 2 of Phase 1 or Phase 4), for $k\in[D]$, is preceded by an execution of $\textsc{Elect}_k$. Hence, the executions of $\textsc{Inform}_k$, for $k\in[D]$, have no impact on the asymptotic complexity of the algorithm (as the complexity of $\textsc{Inform}_k$ is asymptotically smaller than the complexity of $\textsc{Elect}_k$). Thus, the time complexity of algorithm $\mathcal{B}$ is

$$O\left(D\cdot\sqrt{n}\log^6 n + \sum_{k=1}^{D-1} k\log^3 n\frac{n_{k-1}+n_k+n_{k+1}}{\sqrt{n}}\right) = O(D\sqrt{n}\log^6 n) . \qquad \square$$

6 From Broadcasting to Minimal Dominating Set

Observe that the lower bound $\Omega(\sqrt{Dn\log(n/D)})$ on broadcasting can be extended in a natural way to the problem of distributed construction of MDS, since at least one node in the last component of the network used in the proof of the lower bound on broadcasting (c.f., Theorem 4 and Corollary 3) must be reached by the message initiated by the source. Indeed, otherwise all elements of the last component must have decided whether they belong to MDS based merely on the information about their neighbors in a graph, which is insufficient for some network topologies.

Algorithms $\mathcal{A}_k$ and $\mathcal{B}$ could be used as black boxes to obtain MDS in a distributed way in asymptotically the same number of rounds. In the beginning, the broadcasting algorithm $\mathcal{B}$ is run. It is enough to compute sets MDS_k, being the intersection of the final MDS with layer L_k of the network, after (and based on) the execution of $\mathcal{A}_k$, where $k=3i+1$ for non-negative integers i not larger than $(D-2)/3$. Assume that the execution of algorithm $\mathcal{A}_k$ has just finished. First, all nodes that end up Phase 3 in small components without outside neighbors, apply a centralized greedy schedule to select a MDS for the component. Next, nodes that were elected by the source during Phases 1, 3 and 4 check, one after another in the reversed order to the one they were elected in the execution of $\mathcal{A}_k$, whether they have neighbors that have not been dominated yet and whether they have neighbors already selected to the dominating set; both checks are done by using procedure TEST. If the first question is answered affirmative or the second one is answered negative, the node includes itself to the dominating set.

It follows directly from the properties of broadcasting and the above greedy selection made from the broadcasting nodes, that the above algorithm computes a dominating set, and no node can be removed without violating the domination property. In terms of round complexity, the MDS algorithm mimics some operations that occurred in the original execution of the broadcast algorithm $\mathcal{B}$, and therefore its time complexity is (asymptotically) upper-bounded by the time complexity of algorithm $\mathcal{B}$. Thus the following result holds.

Theorem 8. *Every distributed solution building a MDS requires $\Omega(\sqrt{Dn \log \frac{n}{D}})$ rounds on some radio networks of radius D. There exists a distributed algorithm constructing a MDS in $O(D\sqrt{n}\log^6 n)$ on any radio network of radius D.*

References

1. Bar-Yehuda, R., Goldreich, O., Itai, A.: On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. J. Comput. Syst. Sci. 45(1), 104–126 (1992)
2. De Bonis, A., Gąsieniec, L., Vaccaro, U.: Generalized Framework for Selectors With Applications in Optimal Group Testing. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 81–96. Springer, Heidelberg (2003)
3. Brito, C.F., Vaya, S.: Improved lower bound for deterministic broadcasting in radio networks. Theor. Comput. Sci. 412(29), 3568–3578 (2011)
4. Chlamtac, I., Kutten, S.: Tree-based broadcasting in multihop radio networks. IEEE Trans. Computers 36(10), 1209–1223 (1987)
5. Chlamtac, I., Weinstein, O.: The wave expansion approach to broadcasting in multihop radio networks. IEEE Transactions on Communications 39(3), 426–433 (1991)
6. Chlebus, B.S., Kowalski, D.R.: Almost Optimal Explicit Selectors. In: Liśkiewicz, M., Reischuk, R. (eds.) FCT 2005. LNCS, vol. 3623, pp. 270–280. Springer, Heidelberg (2005)
7. Clementi, A.E.F., Monti, A., Silvestri, R.: Selective families, superimposed codes, and broadcasting on unknown radio networks. In: SODA, pp. 709–718 (2001)
8. Czumaj, A., Rytter, W.: Broadcasting algorithms in radio networks with unknown topology. J. Algorithms 60(2), 115–143 (2006)
9. Goldberg, L.A., Jerrum, M., Leighton, F.T., Rao, S.: Doubly logarithmic communication algorithms for optical-communication parallel computers. SIAM J. Comput. 26(4), 1100–1119 (1997)
10. Kowalski, D.R., Pelc, A.: Time of deterministic broadcasting in radio networks with local knowledge. SIAM J. Comput. 33(4), 870–891 (2004)
11. Kowalski, D.R., Pelc, A.: Broadcasting in undirected ad hoc radio networks. Distributed Computing 18(1), 43–57 (2005)
12. Kushilevitz, E., Mansour, Y.: An omega(d log (n/d)) lower bound for broadcast in radio networks. SIAM J. Comput. 27(3), 702–712 (1998)