Deterministic broadcasting in ad hoc radio networks

Bogdan S. Chlebus * Leszek Gąsieniec † Alan Gibbons † Andrzej Pelc ‡
Wojciech Rytter *†

Abstract

We consider the problem of distributed deterministic broadcasting in radio networks of unknown topology and size. The network is synchronous. If a node u can be reached from two nodes which send messages in the same round, none of the messages is received by u. Such messages block each other and node u either hears the noise of interference of messages, enabling it to detect a *collision*, or does not hear anything at all, depending on the model. We assume that nodes know neither the topology nor the size of the network, nor even their immediate neighborhood. The initial knowledge of every node is limited to its own label. Such networks are called *ad hoc multi-hop* networks. We study the time of deterministic broadcasting under this scenario.

For the model without collision detection, we develop a linear-time broadcasting algorithm for symmetric graphs, which is optimal, and an algorithm for arbitrary n-node graphs, working in time $O(n^{11/6})$. Next we show that broadcasting with acknowledgement is not possible in this model at all.

For the model with collision detection, we develop efficient algorithms for broadcasting and for acknowledged broadcasting in strongly connected graphs.

Key words: broadcasting, distributed, deterministic, radio network.

*Instytut Informatyki, Uniwersytet Warszawski, Banacha 2, 02-097 Warszawa, Poland. E-mail: {chlebus,rytter}@mimuw.edu.pl

†Department of Computer Science, The University of Liverpool, Liverpool L69 7ZF, United Kingdom. E-mail: {leszek,A.M.Gibbons,rytter}@csc.liv.ac.uk

‡Département d'Informatique, Université du Québec à Hull, Hull, Québec J8X 3X7, Canada. E-mail: Andrzej_Pelc@uqah.uquebec.ca
Research supported in part by NSERC grant OGP 0008136. This research was done during this author's stay at The University of Liverpool.

A preliminary version of this paper appeared in the Proc. 11th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'2000), 861-870.

1 Introduction

A radio network is a collection of transmitter-receiver devices (referred to as *nodes*). 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. A radio network can thus be conveniently 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, any node u can reach v, if and only if, it can be reached by v, i.e., the reachability graph is symmetric.

The case of symmetric reachability graphs has received most attention in previous research. This case can be equivalently modeled by undirected graphs assuming that communication along edges is bidirectional. In the present paper we consider both symmetric and arbitrary directed graphs, pointing out differences in communication efficiency in both cases. Hence we adopt the more general model of directed graphs, considering symmetric graphs as a special case in this unified model.

Nodes send messages in synchronous *rounds* or *time slots*, measured by a global clock which indicates the current round number. In every round every node acts either as a *transmitter* or as a *receiver*. A node acting as a receiver in a given round gets a message, if and only if, 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 v and v' of 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 *conflict* or a *collision* occurred at u.

One of the fundamental tasks in network communication is *broadcasting*. Its goal is to transmit a message from one node of the network, called the *source*, to all other nodes. Remote nodes get the source message via intermediate nodes, along directed paths in the network. (This is why such networks are often called *multi-hop*.) It is assumed that there exists a directed path from the source to any node. In the sequel we restrict attention to such graphs only.

In this paper we concentrate on one of the most important and widely studied performance parameters of a broadcasting scheme, which is the total time, i.e., the number of rounds it uses to inform all the nodes of the network. There are also other measures of efficiency of broadcasting in radio networks, such as bandwidth consumption [12], or the total number of transmissions [19].

1.1 Previous work

In most of the research on broadcasting in radio networks [1, 4, 11, 21, 27] the network is modeled as an undirected graph, which corresponds to the assumption that the reachability graph is symmetric. A lot of effort has been devoted to finding good upper and lower bounds on the broadcast time in such radio networks, under the assumption that nodes have full knowledge of the network. In [1] the authors proved the existence of a family of n-node networks of radius 2, for which any broadcast requires time $\Omega(\log^2 n)$, while in [21] it was proved that broadcasting can be done in time $O(D + \log^5 n)$, for any n-node network of diameter D. In [32] the authors restricted attention to communication graphs that can arise from actual geometric locations of nodes in the plane. They 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.

Fault-tolerant broadcasting in radio networks was investigated in [26, 29]. In [26] the authors discussed broadcasting in the presence of permanent faults in radio networks arising from geometric locations of nodes on the line and in the plane. [29] was devoted to the description of a broadcasting protocol working in the presence of transient faults.

In all the above papers, the topology of the radio networks was known in advance, and broadcasting algorithms were deterministic and centralized. (Clearly, the lower bound from [1], obtained under the assumption that the topology is known, also holds when the topology is unknown.) On the other hand, in [4] 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 $O(D \log n + \log^2 n)$. In [27] 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.

The above results leave unexplored the scenario of broadcasting protocols which are simultaneously distributed and deterministic, and are to operate correctly in radio networks whose nodes have no knowledge of the network, or even of their immediate neighborhood. Such networks are often called *ad hoc multi-hop* networks. In [4] the authors established a lower bound $\Omega(n)$ on deterministic broadcasting time in such networks, even for the restricted class of symmetric networks. In fact, they constructed a class of symmetric networks of bounded diameter, for which every deterministic broadcasting algorithm uses time $\Omega(n)$. (Later it was shown in [25] that deterministic broadcasting time for this class of networks is the same as for the class of arbitrary symmetric networks and is in fact equal to $n-1$.) In [8] the lower bound $\Omega(D \log n)$ was established for symmetric n-node networks of diameter D, under the additional assumption that only nodes that already got the source message, can transmit. This result easily implies the same lower bound for the class of arbitrary networks, without this extra assumption. More precisely, the symmetric network constructed in [8] can be modified to give a non-symmetric network requiring broadcasting time $\Omega(D \log n)$, even when nodes are allowed to communicate before receiving the source message. (Such a network was constructed, and the lower bound proof was given in a preliminary version of this paper [14]. At this time the authors were unaware of [8].)

To the best of our knowledge, the first paper to deal with deterministic broadcasting protocols under the above scenario of ad hoc multi-hop networks was [9]. Using combinatorial tools involving polynomials over finite fields 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 longest distance from the source to any other node.) This result was further investigated, both theoretically and using simulations, in [10, 7]. On the other hand, a protocol working in time $O(D\Delta \log^{\log \Delta} n)$ was constructed in [6]. Finally, an $O(D\Delta \log^{a} n)$ protocol, for any $a > 2$, was described in [17]. (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. More efficient broadcasting protocols were constructed in [20]. However, radio networks considered there were quite restrictive: the authors considered the situation when nodes are situated on a line, and every node can reach all nodes within a certain distance. Deterministic broadcasting in networks of unknown topology for models other than radio communication have been studied in [3, 23].

1.2 Models

In many applications, nodes of a radio network are mobile, and hence the topology of the network as well as its size may change over time. Hence it is important to design algorithms that do not assume any knowledge that nodes may have about the network, and which nevertheless work efficiently for arbitrary networks.

In this paper we study execution time of broadcasting algorithms that are deterministic and distributed, and work in arbitrary ad hoc radio networks, i.e., radio networks of unknown topology and size. Thus the initial situation is that of complete ignorance concerning the network: the *a priori* knowledge of every node is limited to its own label. For simplicity we assume that labels are distinct integers between 1 and n, for n-node networks, but all the arguments hold if labels are distinct integers between 1 and $Z = O(n)$. (The existence of distinct labels is necessary. If the radio network is anonymous, it is clear that deterministic broadcasting cannot be done even in the 4-cycle.) We study the general case when networks can be arbitrary directed graphs, and also the more restrictive scenario of symmetric graphs, as in [1, 21, 27].

If all the necessary information about the network is available to nodes, time of broadcasting can be known in advance and thus all nodes can be aware of the termination of broadcasting as soon as it is completed. A different situation occurs in our setting. Since even the size of the network is unknown, broadcast can well be finished but no node need be aware of this fact. Consequently, we study the following two communication tasks: In *radio broadcasting* (RB) the goal is simply to communicate the source message to all nodes. In *acknowledged radio broadcasting* (ARB) the goal is to achieve RB and inform the source about it. This may be essential, e.g., when the source has several messages to disseminate, and none of the nodes should learn the next message until all nodes get the previous one.

We assume that the algorithm starts in round 1 and the current round number is indicated by the global clock. (A discussion of differences between this scenario and the situation when nodes have local clocks that tick at the same rate but may start in different rounds, can be found in [24]). An algorithm accomplishes RB in t rounds, if all nodes know the source message after round t, and no messages are sent after round t. An algorithm accomplishes ARB in t rounds, if it accomplishes

RB in t rounds, and if, after round t, the source knows that all nodes know the source message. It should be noted that if ARB is accomplished after $O(t)$ rounds then a more demanding goal of additionally informing all nodes that RB is accomplished can also be achieved in $O(t)$ rounds: the source can simply broadcast the information *done* after learning it.

Since nodes are initially ignorant of the network, it is not surprising that efficiency of broadcasting depends on how fast they can acquire some knowledge of it. Our algorithms are *adaptive*, i.e., a node can schedule its future actions on the basis of its local history. A node can obviously gain knowledge from previously obtained messages. There is, however, another potential way of learning useful information. The availability of this method depends on what exactly happens during a conflict, 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. These two scenarios are often referred to as the absence (resp. availability) of *collision detection* (cf., e.g., [4]). Which of the two scenarios occurs in a particular situation may depend on technological characteristics of the transmitter/receiver devices used by the nodes. A discussion justifying both scenarios can be found in [22, 4]. We will see that efficiency and even feasibility of a particular communication task are significantly influenced by the choice between these scenarios.

We briefly discuss how to relate these considerations to the OSI layers ([33]). Our algorithms are mainly relevant to what is done at the Network Layer. No routing of packets is necessary which makes implementation of the developed algorithms especially simple. Nodes just need to decide whether to transmit or not at a given step. Transmissions of packets are handled by the Data Link Layer. We do not consider the possibility of transmission failures, hence no dialog among senders and receivers to schedule retransmissions is carried out at this layer. A collision with some other packet is the only possible reason that a packet cannot be received. If such collisions can be physically detected then this is managed by the Data Link Layer. It is the Network Layer which reacts to collisions because it is solely responsible for scheduling of transmissions.

1.3 Overview of results

Model without collision detection.

Our main results are in this model. We first give an algorithm which performs radio broadcasting for all symmetric graphs in time $O(n)$, where n is the number of nodes in the graph. While it is not difficult to design such an algorithm assuming that nodes know their neighborhood, it should be recalled that we do not assume this knowledge and we do not even assume the knowledge of any bound on network size. In coping with this total ignorance of the network, the main problem becomes assuring that every node has learned its complete neighborhood. Since the lower bound $\Omega(n)$ from [4] holds for the same model, our algorithm is asymptotically optimal.

Our linear-time algorithm should be contrasted with the lower bound $\Omega(D \log n)$ from [8]. For $D \in \Theta(n)$ this gives lower bound $\Omega(n \log n)$. However, this lower bound is obtained assuming that only nodes that already got the source message can transmit, while our algorithm makes heavy use of preprocessing in which uninformed nodes transmit control messages. On the other hand, as mentioned in Section 1.2, this lower bound can be modified to hold for general algorithms but then the network requiring long broadcasting time is found in the class of arbitrary directed networks, and not in the restricted class of symmetric networks for which our algorithm is designed.

For arbitrary directed graphs we give a broadcasting algorithm running in time $O(n^{11/6})$. It is based on a combinatorial idea using Vishkin's deterministic sample [34]. It should be noted that after publication of the preliminary version of this paper [14], a series of faster broadcasting algorithms have been proposed, including one constructive algorithm with execution time $O(n^{3/2})$ [15], and three non-constructive algorithms based on probabilistic methods with execution times $O(n^{5/3} \log^{1/3} n)$ [18], $O(n^{3/2}\sqrt{\log n})$ [31] and $O(n \log^2 n)$ [16]. The algorithm from [15] used the theory of finite fields, similarly as [9], while the above non-constructive algorithms used the concept of a *selective family* that is introduced in this paper, see section 2.2.

All the above algorithms (including the $O(n^{11/6})$ algorithm presented in this paper) should be contrasted with algorithms cited in Section 1.1 whose efficiency depends on the eccentricity of the source and on maximum degree of the graph. The previously mentioned algorithms were designed in a layer-by-layer fashion, guaranteeing a given upper bound on time spent between informing layer i (the set of nodes at distance i from the source) and layer $i+1$. As opposed to this strategy, our algorithm (and its later improvements cited above) guarantees a certain rate of the number of nodes learning the source message per time unit. Such algorithms give good time guarantees (in terms of n) for any n-node network but these guarantees are not necessarily improved for networks of small source eccentricity or small maximum degree. On the other hand, the other algorithms, efficient for such networks, give poor performance guarantees for n-node networks in the worst case. Thus the qualities and drawbacks of both groups of algorithms are, in a sense, orthogonal, and the choice of a suitable algorithm should depend on the expected application.

Our last result in the model without collision detection is an impossibility result. We show that no deterministic algorithm can perform acknowledged radio broadcasting in this model, even for symmetric graphs.

Model with collision detection.

For the class of strongly connected graphs we show a radio broadcasting algorithm working in time $O(n \cdot ecc)$, where ecc is the eccentricity of the source in the graph. We also observe how the collision detection capability can be used to code messages. We show a simple scheme that broadcasts a message of size l in time $O(l \cdot ecc)$, hence it is an asymptotically optimal algorithm to broadcast messages of size $O(1)$, in arbitrary graphs.

The impact of collision detection is even stronger for acknowledged radio broadcasting. While this is an impossible task without this capability, availability of it permits to perform ARB rather fast. For symmetric graphs we show an algorithm working in time $O(n)$ for n-node graphs. If the graph is non-symmetric, in order to make ARB possible, it must be at least strongly connected. For such graphs we show an algorithm for acknowledged radio broadcasting working in time $O(n \cdot ecc)$.

While we show an asymptotically optimal broadcasting algorithm for ad hoc symmetric networks in the absence of collision detection, the main open problem yielded by our results is that of finding such an algorithm for arbitrary ad hoc networks, with or without collision detection.

The paper is organized as follows. In section 2 we present our results assuming no collision detection and in section 3 we show how the complexity of broadcasting changes when this capability is added.

2 The model without collision detection

In this section we consider broadcasting in the model without collision detection, similarly as, e.g., in [4]. Thus, if at least two neighbors of u transmit in a given round, the effect at u is identical as if none of its neighbors transmitted.

2.1 Radio broadcasting in symmetric graphs

Algorithm Explore-and-Expand

The algorithm works for every symmetric graph in time $O(n)$, where n is the number of nodes. In every round, at most one node acts as a transmitter and all other nodes act as receivers, hence collisions are avoided. The algorithm works in phases, numbered by consecutive positive integers. Every phase starts in the round following the end of the previous phase. Phase k lasts $7 \cdot 2^{k-1}$ rounds divided into 3 stages. Stage A consists of 2^{k-1} rounds, stage B consists of 2^k rounds, and stage C consists of 2^{k+1} rounds. Let L_k be the set of nodes with labels 1,..., 2^k, and let G_k be the connected component of the network containing the source and induced by L_k, if the label of the source is at most 2^k. Otherwise, let G_k be the empty graph. The following invariant *Inv* will be maintained after phase k, for any positive integer k.

- A spanning tree T_k of G_k and an Eulerian cycle C_k on this tree are constructed.
- Nodes on the cycle are labeled by consecutive positive integers and each node knows the number(s) assigned to it in this enumeration. These numbers are called *k-tags* of the node.
- Every node in G_k knows the labels of all its neighbors in G_k.
- Every node in G_k knows the source message.

Assume that the invariant is maintained after phase $k-1$ of Algorithm Explore-and-Expand. We describe phase k and show that the invariant is maintained after it as well.

Stage A. Rounds in stage A of phase k are numbered by integers $2^{k-1}+1$,..., $2^{k-1}+2^{k-1}$. In round number j of stage A the node with label j acts as a transmitter and sends a message containing its label.

Stage B. Rounds of this stage are numbered by integers 1,..., 2^k. A node in the cycle C_{k-1} is called *active* in a given round of stage B if it has heard a message in stage A or in an earlier round of stage B. An active node acts as a transmitter in every round of stage B whose number corresponds to its k-tag. In such a round the node sends a *contact message* which is a fixed single-bit signal.

Stage C. If the source becomes active during stage B, it initiates stage C. At the beginning of this stage every node v knows its neighborhood N_v in G_k. In this stage a token initially situated in the source explores G_k in a DFS manner, and constructs a spanning tree T_k of this subgraph and an Eulerian cycle C_k on this tree. The token carries the source message and a counter, initially set to 1, which is incremented by one after each move. We use an idea similar to that from [2] in order to limit the token's moves only to edges of the spanning tree.

Every node v maintains a list Q_v containing the set of its neighbors in G_k which were not yet visited by the token. Q_v is initialized to N_v. Whenever v gets the message `visited` from some neighbor w, or whenever v gets the token from w, node v removes w from the list Q_v.

When v gets the token with counter c then:

If $Q_v = \emptyset$ then

- v assigns itself k-tag $c+1$;
- v sends the message $\langle$ `label(v), visited` $\rangle$;
- v sends the token with counter $c+1$ to the node w from which it got it for the first time (by sending a message $\langle$ `token, c+1, label(w)`$\rangle$) .

If $Q_v \neq \emptyset$ then

- v assigns itself k-tag $c+1$;
- v sends the message $\langle$ `label(v), visited` $\rangle$;

- v sends the token with counter $c+1$ to the node w with the smallest label in the list Q_v.

In both cases both messages are concatenated and are sent in a single round.

This concludes the description of stage C and of phase k of Algorithm Explore-and-Expand.

Lemma 2.1 *After phase k of Algorithm* Explore-and-Expand *the invariant Inv is maintained, for any positive integer k.*

Proof: After stage A every node in G_k knows its neighborhood in G_k. Indeed, it knows its neighborhood in G_{k-1} after phase $k-1$, and learns about the remaining neighbors in G_k during stage A of phase k.

If there are nodes in G_k outside of G_{k-1}, some node in the cycle C_{k-1} is active in the beginning of stage B. The length of this cycle is at most $2(2^{k-1}-1)$ hence at the end of stage B every node in the cycle, in particular the source (if G_k is not empty), is active. In stage C the moves of the token form the eulerian cycle C_k of a spanning tree T_k of the subgraph G_k. Since all nodes of G_k are visited by the token, they all learn the source message. All nodes get their k-tags in the process. □

Theorem 2.1 *Algorithm* Explore-and-Expand *performs radio broadcasting in time $O(n)$, in any n-node graph.*

Proof: Let k be such that $2^{k-1} < n \leq 2^k$. It is enough to show that

1. After phase k no messages are sent;
2. After phase k all nodes of the network get the source message.

Indeed, (1) guarantees that the algorithm stops and its duration is at most

$$\sum_{i=1}^{k} 7 \cdot 2^{i-1} \leq 7 \cdot 2^k \leq 14n,$$

while (2) guarantees that radio broadcasting is accomplished.

In order to prove (1) notice that $G_r = G_{r+1}$, for all $r \geq k$. Consider any phase r, with $r > k$. There are no nodes sending messages in stage A. Hence no messages are sent in stage B. Consequently the source does not become active in this stage and no messages are sent in stage C.

In order to prove (2) consider phase k (disregarding everything that happened previously). Since G_k is the entire network, (2) follows from Lemma 2.1. □

It should be noted that although Algorithm EXPLORE-AND-EXPAND accomplishes radio broacasting after $O(n)$ rounds, none of the nodes is aware that the process is finished, i.e., acknowledged radio broadcasting is not achieved. This is not accidental: we will prove later that this task is impossible to achieve in the absence of collision detection.

2.2 Radio broadcasting in arbitrary graphs

We start with a simple algorithm, working in time $O(n^2)$ for any n-node graph, and then present a more subtle algorithm with complexity $O(n^{11/6})$.

Algorithm SIMPLE-SEQUENCING

The algorithm works in phases numbered by positive integers. Phase k consists of 2^k segments, each consisting of 2^k rounds. Rounds in each segment are numbered by integers $1, ..., 2^k$. (This number should not be confused with the global number of the round.) The first segment of phase 1 starts in round 1, every segment of a phase begins in the round following the end of the previous segment, and phase $k+1$ begins in the round following the end of phase k. Hence every node knows to which segment of which phase the current round belongs and what is the number of this round in this segment.

Every node acts as a transmitter in exactly one round (after getting the source message), in all other rounds it acts as a receiver. A node acting as a transmitter sends the source message. In every round at most one node acts as a transmitter, hence collisions are avoided. The following schedule is used. Consider a node with label i that got the source message in round r. The node waits till the first round $r' > r$ whose number in its segment is i and acts as a transmitter in this round.

Theorem 2.2 *Algorithm* SIMPLE-SEQUENCING *performs radio broadcasting in time $O(n^2)$, in any n-node graph.*

Proof: Let k be such that $2^{k-1} < n \leq 2^k$. Since phase k lasts 4^k rounds, it is enough to show that

1. After phase k no messages are sent;
2. After phase k all nodes of the network know the source message.

Assertion (2) follows from the fact that a node at distance d from the source gets the source message in a round of the dth segment of the kth phase, at the latest. (This is easily proved by induction on d.) Since $d \leq n-1 \leq 2^k - 1$, it follows that every node will act as a transmitter in a round of the last segment of phase k, at the latest. This implies assertion (1). □

Our next algorithm improves on the time complexity of Algorithm SIMPLE-SEQUENCING by using two essential tools: one is the notion of *k-selective* families, and the other is that of *deterministic sample*, due to Vishkin [34]. Using these tools we will show that radio broadcasting can be done in time $O(n^{11/6})$, for arbitrary n-node graphs.

Let $[a] = \{1, 2, \ldots, a\}$. A family $\mathcal{F}$ of subsets of U is said to be *k-selective* for the set U iff for any $X \subseteq U$, $|X| \leq k$, there is a set $Y \in \mathcal{F}$ satisfying $|X \cap Y| = 1$.

We will construct a small k_m-selective family for the set $[2^m]$, with large k_m. The crucial tool is that of a *deterministic sample*, originally introduced by Vishkin [34] and used successfully in parallel string-matching. Intuitively speaking, a deterministic sample allows to identify succinctly a single object from a given subset. Denote by $\mathcal{P}_r(m)$ the family of all subsets of cardinality r of $[m]$. A deterministic sample with parameters r, m is a member of the set

$$DetSamples(r, m) \;=\; \{(D, \alpha) \;:\; D \in \mathcal{P}_r(m),\; \alpha \in \{0,1\}^r\}.$$

For $\Delta = (D, \alpha) \in DetSamples(r, m)$ define $STRINGS(\Delta)$ as the set of all binary strings of length m whose sequence of bits on positions in D, read in increasing order of indices, equals α.

Lemma 2.2 *[**deterministic-sample lemma**], [34]*
The family $\{STRINGS(\Delta)\} : \Delta \in DetSamples(r, m)\}$ is 2^r-selective for the set $\{0,1\}^m$.

Proof: Let X be a set of at most 2^r binary strings of length m for $r \geq 1$. A deterministic sample (D, α) satisfying $|STRINGS(D, \alpha) \cap X| = 1$ is constructed by the following *halving strategy.*

Initially set $X' = X,\ D = \emptyset,\ \alpha = \textit{empty-string}.$
Then repeat the following 4 operations until $|X'| = 1$:

1. find the first $i \in [m]$ such that two strings in X' differ on the position i;
2. take $b \in \{0, 1\}$ which minimizes $|X' \cap STRINGS(\{i\}, b)|$;
3. $D := D \cup \{i\}$; $\alpha := \alpha \cdot \langle b \rangle$; (dot means concatenation)
4. $X' := X' \cap STRINGS(\{i\}, b)$.

If $|D| < r$ then we add to D aribitrarily $r - |D|$ extra positions and add to α the symbols corresponding to these extra positions in the only string of X'.

There are at most r iterations, since each time we halve the set X'. It follows from the construction that $|STRINGS(D, \alpha) \cap X| = 1$. □

Lemma 2.3 *$|DetSamples(\lceil m/6 \rceil, m)|$ is $O(2^{5m/6})$.*

Proof: We estimate the number of deterministic samples $\Delta = (D, \alpha)$. There are $2^{\lceil m/6 \rceil}$ possible strings α and $\binom{m}{\lceil m/6 \rceil}$ possible subsets D. By the Stirling formula we have, for some constant c:

$$\binom{m}{\lceil m/6 \rceil} \leq c \cdot \frac{m^m}{(\frac{m}{6})^{m/6} \cdot (m(1 - \frac{1}{6}))^{m(1-\frac{1}{6})}} = c \cdot (6^{\frac{1}{6}} \cdot (\frac{6}{6-1})^{\frac{6-1}{6}})^m.$$

Since $6^{\frac{1}{6}} \cdot (\frac{6}{5})^{\frac{5}{6}} \leq 2^{2/3}$, we get $\binom{m}{\lceil m/6 \rceil} = O(2^{\frac{2}{3}m})$, which proves the lemma. □

Lemma 2.2 for $r = \lceil m/6 \rceil$ and Lemma 2.3 yield the construction of a $2^{\lceil m/6 \rceil}$-selective family of size $O(2^{5m/6})$ on the set $[2^m]$, for any integer m, in view of the natural bijection between $\{0,1\}^m$ and $[2^m]$. For any positive integer m, fix such a family $\mathcal{F}_m$ and let $f_m = |\mathcal{F}_m|$ and $k_m = 2^{\lceil m/6 \rceil}$.

During the execution of the broadcasting algorithm every node of the network is in one of the following states: *uninformed* (a node has not received the source message yet), *active* (a node has the source message and acts as a transmitter in rounds prescribed by the algorithm), and *switched-off* (a node has the source message, but it remains silent till the end of the algorithm). A node that is in state *uninformed* (*active*,*switched-off*) is simply called uninformed (active, switched-off). All nodes (apart from the source that is initially active) are initially uninformed. When an uninformed node gets the source message it becomes active. Some time after all neighbors of an active node learn about the source message, the active node becomes switched-off. The exact time when an active node becomes switched-off is defined by the following procedure.

Procedure m-SEGMENT

Let $F_1, ..., F_{f_m}$ be an arbitrary enumeration of $\mathcal{F}_m$. The procedure works in $k_m f_m + 2^m$ rounds and consists of two parts: the *head* composed of k_m consecutive sequences, called *windows*, each of f_m rounds corresponding to the appropriate sets F_i, followed by the *tail* composed of 2^m rounds. In any round of the head, corresponding to set F_i, every active node with label $l \in F_i$, acts as a transmitter and sends the source message; all other nodes act as receivers. In the lth round of the tail, the active node v with label l acts as a transmitter, sends the source message and becomes switched-off; all other nodes act as receivers.

Algorithm SELECTIVE-BROADCASTING

The algorithm works in consecutive phases numbered by non-negative integers. At the beginning of phase 0 all nodes (except the source which is active) are uninformed. Phase m consists of $\lceil 2^m/k_m \rceil$ consecutive calls of procedure m-SEGMENT described above.

A node v is called an m-node if there exists a directed path from the source s to v, whose all nodes, including s and v, have labels at most 2^m.

Lemma 2.4 *After phase m of Algorithm* Selective-Broadcasting *all m-nodes are switched-off.*

Proof: We prove the lemma by induction on m. When $m = 0$ the lemma is true, since either the set of 0-nodes is empty (the source s has a label greater than 1) or the source s has label 1, and it becomes switched-off just after the execution of the tail of procedure 0-Segment. Assume the lemma is true for all $0 \leq m' < m$. By the inductive hypothesis, all $(m-1)$-nodes are switched-off at the beginning of phase m. Consider a fixed call S of procedure m-Segment in phase m. We show that the number of active m-nodes at the beginning of the tail of S is at least $\min(k_m, x)$, where x is the total number of m-nodes that are not switched-off yet. (This implies that after $\lceil 2^m/k_m \rceil$ calls of the procedure all m-nodes become switched-off.) Consider two cases. If at the beginning of some window in S the number of active m-nodes is larger than k_m, all these nodes will be also active at the beginning of the tail of S. Otherwise, we show that in each window the number of active nodes increases, in view of k_m-selectivity of $\mathcal{F}_m$. Indeed, fix a window, and let A be the set of those active m-nodes at the beginning of this window that are adjacent to uninformed m-nodes. Hence $|A| \leq k_m$. If A is empty then there are no uninformed m-nodes left. Assume $A \neq \emptyset$. Consequently, in some round t of the window there is exactly one node $x \in A$ which acts as a transmitter and sends the source message. Consider a node v which was an uninformed neighbor of x at the beginning of the window. Three situations are possible: (1) v became active before round t: in this case the number of active nodes increased, (2) v was uninformed before round t and x is the only node transmitting in round t: then v becomes active in round t and the number of active nodes increased, (3) some other node y transmits in round t: this node must have become active in some round of the window before round t, and hence the number of active nodes increased.

Hence in each window the number of active m-nodes increases, and consequently at the beginning of the tail of S there are at least $\min(k_m, x)$ active m-nodes. As observed above, this concludes the proof.

□

Theorem 2.3 *Algorithm* Selective-Broadcasting *performs radio broadcasting in time $O(n^{11/6})$, in any n-node graph.*

Proof: Let m^* be such that $2^{m^*-1} < n \leq 2^{m^*}$. By Lemma 2.4 all nodes know the source message after phase m^* and no messages are sent after this phase. Phase m lasts $(k_m \cdot |\mathcal{F}_m| + 2^m) \cdot \lceil 2^m/k_m \rceil$ rounds. Since $|\mathcal{F}_m| = O(2^{5m/6})$, $k_m = O(2^{m/6})$ and $O(n) = O(2^{m^*})$, this implies that the total time of execution of phases 0 to m^* is $O(n^{11/6})$. □

Similarly as in the case of Algorithm Explore-and-Expand for symmetric graphs, in the case of Algorithms Simple-Sequencing and Selective-Broadcasting, none of the nodes is aware that the process is finished, i.e., acknowledged radio broadcasting is not achieved. In what follows we will show that ARB is in fact impossible to achieve without collision detection.

2.3 Infeasibility of acknowledged radio broadcasting

We conclude this section by proving that in the absence of collision detection acknowledged radio broadcasting cannot be accomplished, even for symmetric graphs. In order to do so we need to define precisely an ARB-protocol. We want to capture the intuition that the behavior of any node in any given round depends only on its local history, this history consisting of the node's label and all messages previously received by it. Moreover, at the termination of an ARB-protocol, the source must be convinced that all nodes got the source message, and this conviction must be correct.

We first define an *input* in round t. This is either a sequence $(t, (t_1, m_1), ..., (t_k, m_k))$, or a sequence $(t, \sigma, (t_1, m_1), ..., (t_k, m_k))$ where $t_1 < \cdots < t_k < t$ are positive integers, σ is an arbitrary finite binary string and m_i are arbitrary messages (coded as finite binary strings). The intuitive meaning of an input is the following. The input of a given node $v \neq s$ in round t is the entire information v obtained from its neighbors before this round, during the execution of the protocol, together with the number of the current round, read on the global clock. Integers t_i are rounds preceding t in which exactly one of v's neighbors acted as a transmitter (in other rounds v could not hear anything), and m_i is the message transmitted by this unique neighbor in the respective round. The input of the source (in any round) contains, moreover, the source message σ. Hence the total knowledge of a node in a given round consists of its input in this round, and of its label.

Let $\mathcal{I}$ denote the set of all inputs and $\mathcal{N}$ the set of all positive integers. Let ρ, τ, θ, ϕ and ω be special symbols whose meaning will be specified below. Let M be the set of couples (τ, m) and let K be the set of couples (ϕ, m), where m is an arbitrary finite binary string.

An ARB-protocol is a function

$$\mathcal{P} : \mathcal{I} \times \mathcal{N} \to (\{\rho\} \cup M) \times (\{\omega\} \cup \{\theta\} \cup K).$$

$\mathcal{P}(I, l)$ is the output of the protocol $\mathcal{P}$ in round t at a node with label l, if the input of this node in round t is I. This output is a couple $(\mathcal{P}_1(I, l), \mathcal{P}_2(I, l))$, where $\mathcal{P}_1(I, l) \in \{\rho\} \cup M$ and $\mathcal{P}_2(I, l) \in \{\omega\} \cup \{\theta\} \cup K$. $\mathcal{P}_1(I, l)$ represents the decision of the node how to act: it is either ρ (act as a receiver) or (τ, m) (act as a transmitter and transmit message m). $\mathcal{P}_2(I, l)$ represents the state of knowledge of the node. For any node, it is either θ (I am yet uninformed), or (ϕ, m) (I know the source message and I think it is m). Additionally, if s is the label of the source, the value $\mathcal{P}_2(I, s)$ can be ω: if I is the input of the source in round t, then $\mathcal{P}_2(I, s) = \omega$ means that the source considers RB to be finished by round t.

The protocol $\mathcal{P}$ achieves ARB by round t, if the following two conditions are satisfied:

- $\mathcal{P}_2(I_s, s) = \omega$, where s is the label of the source and I_s is the input of the source in round t.
- for any round t', if $\mathcal{P}_2(J_s, s) = \omega$, where J_s is the input of the source in round t', then $\mathcal{P}_2(J_v, v) = (\phi, \sigma)$, where J_v is the input of the node with label v in round t', for all labels v of nodes other than the source, where σ is the source message (included in the input of the source).

The above formalism captures the property that ARB is accomplished by round t if, in round t, the source considers RB to be accomplished, and if, in any round, such a conviction of the source implies that RB is indeed accomplished by this round.

The following result states the infeasibility of ARB, even in the class of symmetric graphs.

Theorem 2.4 *For any ARB-protocol $\mathcal{P}$ there exists a symmetric graph on which $\mathcal{P}$ is not correct.*

Proof: Fix an ARB-protocol $\mathcal{P}$ and fix two distinct source messages σ' and σ''. Let t' (resp. t'') be the first round in which $\mathcal{P}$ completes ARB on the 1-node graph consisting only of the source s, if the source message is σ' (resp. σ''). Let $t = \max(t', t'')$. Thus t is the time after which the source realizes that it is the only node of the network, if the source message is σ' or σ''. The idea of the proof is to construct a labeled graph G with at least two nodes, for which

- some nodes do not get the source message after t rounds;
- the input of the source during the first t rounds is the same as in the 1-node graph consisting of the source, i.e., the source does not get any messages in the first t rounds,

assuming that the source message is σ' or σ''.

Then, the protocol $\mathcal{P}$ working on the graph G will induce the source to falsely conclude that ARB is accomplished after t rounds.

The graph G has $2^{2t} + 4t + 2$ nodes and is defined as follows. The set of nodes is $V = \{a, b\} \cup X$, where X has size $2^{2t} + 4t$ and contains the source s. The set of directed edges is $E = \{(a, x), (x, a), (b, x), (x, b) : x \in X\}$, i.e., the graph is symmetric. Choose any node $s \in X$ as the source. Labeling will be defined below and depends on the protocol $\mathcal{P}$. Intuitively, we want to label nodes in such a way that a and b, the only neighbors of s, either both act as transmitters or both act as receivers in the first t rounds of protocol execution. This is the reason for constructing a graph with so many nodes (we find the appropriate labels for a and b by a halving strategy) and it is the main difficulty of the proof. This difficulty is aggravated by the fact that nodes a and b have to behave identically in spite of the possibility that nodes in X, including the source, may spontaneously send some messages that may be received by nodes a and b, in these early rounds.

Assign label 1 to s. Let L denote the set of integers $\{2, ..., 2^{2t} + 4t + 2\}$, let T^* denote the set of integers $\{1, ..., t\}$, and let ϵ denote the empty sequence. For any round $i \in T^*$, let I_i denote the input $I(i, \epsilon)$ (of nodes other than the source), and let $J_{i,\sigma}$ denote the input $I(i, \sigma, \epsilon)$ (of the source), where σ is a source message. Hence I_i (resp. $J_{i,\sigma}$) are inputs in round i corresponding to the situation when a node other than the source (resp. the source, given source message σ) did not get any messages until round i.

Let T' be the set of those rounds $r \leq t$ for which $\mathcal{P}_1(J_{r,\sigma'}, 1) \in M$, i.e., those rounds in which the source acts as a transmitter if it has not received any message yet, and if the source message is σ'. Let T_1' be the set of those rounds $r \in T'$ for which there exists a label $c_r' \in L$ such that $\mathcal{P}_1(I_r, c_r') \in M$. I.e., T_1' is the set of those rounds in which the source and some other node whose label is in L act as transmitters if they have not received any message yet, and if the source message is σ'. Let $A' = \{c_r' : r \in T_1'\}$.

Let T_2' be the set of those rounds $r \in T^* \setminus T'$ for which there exists a unique label $d_r' \in L$ such that $\mathcal{P}_1(I_r, d_r') \in M$. I.e., T_2' is the set of those rounds in which a unique node acts as a transmitter if it has not received any message before, and if the source message is σ'. Let $B' = \{d_r' : r \in T_2'\}$.

Finally, let T_3' be the set of those rounds $r \in T^* \setminus T'$ for which there exist at least two labels $x_r, y_r \in L$ such that $\mathcal{P}_1(I_r, x_r) \in M$ and $\mathcal{P}_1(I_r, y_r) \in M$. I.e., T_3' is the set of those rounds outside of T' in which at least two nodes with labels in L act as transmitters if they have not received any message before. Let $C' = \{x_r, y_r : r \in T_3'\}$.

The set $Y' = A' \cup B' \cup C'$ has size at most $2t$. Assign labels from Y' to arbitrarily chosen nodes from $X \setminus \{s\}$. Assume that the source message is σ'. This assignment implies that, regardless of how other labels from the set L are assigned to other nodes from X, the only rounds in which nodes a and b can get a message by round t are the rounds from the set $T' \setminus T_1' \cup T_2'$.

Next, define sets T'', T_1'', T_2'', T_3'', A'', B'', C'', and Y'', similarly as T', T_1', T_2', T_3', A', B', C', and Y', with the exception that source message σ' is replaced by source message σ''. Again Y'' has size at most $2t$. Assign labels from Y'' to arbitrarily chosen, yet unlabeled nodes from $X \setminus \{s\}$. Assume that the source message is σ''. This assignment implies that, regardless of how other labels from the set L are assigned to other nodes from X, the only rounds in which nodes a and b can get a message by round t are the rounds from the set $T'' \setminus T_1'' \cup T_2''$.

Let r be any round in $T' \setminus T_1'$ and let $\mathcal{P}_1(J_{r,\sigma'}, 1) = (\tau, m_r')$. Let u be any round in T_2' and let $\mathcal{P}_1(I_u, d_u') = (\tau, m_u')$. Hence, if the source message is σ', the only messages that nodes a and b can get by round t are messages m_r', for $r \in T' \setminus T_1'$, and m_u', for $u \in T_2'$. More precisely, let $r_1', ..., r_q'$ be an increasing ordering of rounds in $T' \setminus T_1' \cup T_2'$. For any round $p \leq t$ let $j(p)'$ be the largest index

j such that $r'_j < p$. Hence, if the source message is σ', the inputs of nodes a and b in round p are identical and equal to $\Gamma'_p = (p, (r'_1, m'_{r'_1}), ..., (r'_{j(p)'}, m'_{r'_{j(p)'}}))$.

We define a decreasing sequence of sets L'_i, for $i = 0, ..., t$, by induction. Let $L'_0 = L \setminus Y' \setminus Y''$.

$^{2t} + 1$. Suppose that L'_{p-1} is already defined. Consider the following sets

$$H_1 = \{l \in L'_{p-1} : \mathcal{P}_1(\Gamma'_p, l) = \rho\},$$

$$H_2 = \{l \in L'_{p-1} : \mathcal{P}_1(\Gamma'_p, l) \in M\}.$$

ne set of those labels in L'_{p-1} which cause a node to act as a receiver iven input Γ'_p in round p, if the source message is σ'. Let H be the set I_1 and H_2. We define $L'_p = H$. It follows that L'_t has size at least $2^t + 1$.

ssages $m''_{r''_i}$, and input Γ''_p similarly as r'_i, $m'_{r'_i}$, and Γ'_p but assuming that nstead of σ'. Define a decreasing sequence of sets L''_i, for $i = 0, ..., t$, by L'_t. Sets L''_i, for $i > 0$, are defined similarly as L'_i, replacing input Γ'_p by source message is σ''. It follows that L''_t has size at least 2.

bels in L''_t. We assign those labels to nodes a and b, respectively, and we ining labels in L'_0 to yet unlabeled nodes in the set X.

l_2, nodes a and b either both act as receivers or both act as transmitters ardless of whether the source message is σ' or σ''. It follows that in any ne source is $J_{i,\sigma'}$, if the source message is σ', and it is $J_{i,\sigma''}$, if the source put of any other node in X is I_i, in both cases. However, since for the 1- plished by round t, it means that $\mathcal{P}_2(J_{t,\sigma'}, 1) = \mathcal{P}_2(J_{t,\sigma''}, 1) = \omega$. Suppose he graph G. By definition this implies that $\mathcal{P}_2(I_t, l) = (\phi, \sigma') = (\phi, \sigma'')$, $X \setminus \{s\}$. This implies $\sigma' = \sigma''$, contradiction. □

3 The model with collision detection

In this section we show how our results change when collision detection is available. Clearly, all RB algorithms described in the previous section remain valid in the present setting. However, we will present a new RB algorithm working more efficiently for a large class of graphs, and another very simple algorithm which enables to broadcast short messages in arbitrary graphs very fast. We will also show that (as opposed to the scenario without collision detection) acknowledged radio broadcasting is now possible for all strongly connected graphs, and can in fact be performed efficiently.

We will use the following terminology. A node v acting as a receiver in a given round *hears signal* μ if at least one of its neighbors acts as a transmitter in this round, i.e., if v hears any message or if there is a conflict between its neighbors in this round. Otherwise (if none of its neighbors acts as a transmitter in this round), v *hears nothing* (background noise). A *contact message* is a fixed one-bit signal.

A graph is *strongly connected* if there exists a directed path between any pair of nodes.

3.1 Radio broadcasting in strongly connected graphs

We first show an algorithm accomplishing RB in time $O(n \cdot ecc)$ in any strongly connected n-node graph for which ecc is the eccentricity of the source.

Algorithm BOUND-AND-BROADCAST

The algorithm works in two stages. The aim of the first stage is that all nodes find an upper bound on n within a factor of 2. In the second stage the actual broadcasting is performed.

Stage 1 is divided into phases numbered by consecutive positive integers. The kth phase consists of $2^k + 1$ rounds. The first phase starts in round 1, and each phase starts in the round following the end of the previous phase. Rounds of the kth phase are numbered by consecutive integers 1,...,$2^k + 1$. A node is said to be *active* after the jth phase, $j > 1$, if it acted as a transmitter in some round of this phase. All nodes are active in the first phase. We will show (cf. Lemma 3.1) that after each phase either all nodes are active or all are not active. In the first round of the kth phase all active nodes whose labels are larger than 2^k act as transmitters, sending a contact message, and all others act as receivers. In every round $i > 1$ of the kth phase all active nodes that heard signal μ in round $i - 1$ of this phase and have not heard it in any previous round of this phase, act as transmitters, sending a contact message, and all others act as receivers. The first phase after which all nodes are not active, is the last phase of the first stage. Every node can identify this phase as the first phase after which it is not active. Suppose that the number of this phase is k_0. We will prove (cf. Lemma 3.2) that $2^{k_0-1} < n \leq 2^{k_0}$.

Stage 2 starts in the round following the end of Stage 1. It is divided into phases consisting of 2^{k_0} rounds, numbered by consecutive positive integers. In each phase rounds are numbered by consecutive integers 1,...,2^{k_0}. In each round, at most one node acts as a transmitter, thus collision is avoided. Every node acts as a transmitter in exactly one round, after receiving the source message. Let l_s denote the label of the source. In all rounds $i < l_s$ of the first phase all nodes act as receivers. In round l_s of the first phase the source acts as a transmitter sending the source message. Every other node with label l that got the source message in phase j waits till the first round whose number in its phase is l (this can happen in phase j or in phase $j+1$ at the latest) and acts as a transmitter, sending the source message.

This completes the description of Algorithm BOUND-AND-BROADCAST. In order to prove its correctness and analyze its efficiency, we need the following lemmas.

Lemma 3.1 *If $n > 2^k$ then all nodes are active after the kth phase of Stage 1 of Algorithm* BOUND-AND-BROADCAST. *Otherwise, all nodes are not active after the kth phase of Stage 1.*

Proof: Suppose the lemma is true for all phases $k' < k$ of Stage 1. If all nodes were not active after some phase $k' < k$ then all of them are not active after the kth phase. Suppose that all nodes were active in all phases $k' < k$.

If $n > 2^k$ then in the first round of phase k all nodes whose labels are larger than 2^k act as a transmitter. For every node v with label $l \leq 2^k$ there exists a node w with label $l' > 2^k$ such that there is a directed path of length at most 2^k from w to v. Hence node v is guaranteed to hear signal

μ in round 2^k of the kth phase at the latest, and hence it acts as a transmitter in the kth phase. Consequently, all nodes act as transmitters in this phase.

If $n \leq 2^k$ then in the first round of phase k no node acts as a transmitter, and consequently no node hears signal μ or acts as a transmitter in this phase. □

Lemma 3.2 *Let k_0 be the first phase of Stage 1 after which all nodes are not active. Then $2^{k_0-1} < n \leq 2^{k_0}$.*

Proof: In view of Lemma 3.1, we have $2^{k_0-1} < n$, since all nodes were active after phase $k_0 - 1$. If $n > 2^{k_0}$, all nodes would be active after phase k_0. □

Theorem 3.1 *Algorithm* Bound-and-Broadcast *performs radio broadcasting in any strongly connected graph. It works in time $O(n \cdot ecc)$ in any strongly connected n-node graph for which ecc is the eccentricity of the source.*

Proof: Let k_0 be the first phase of Stage 1 after which all nodes are not active. By Lemma 3.2, Stage 1 lasts at most

$$\sum_{i=1}^{k_0} 2^i < 2^{k_0+1} \leq 4n$$

rounds. Every phase of Stage 2 lasts $2^{k_0} \leq 2n$ rounds. Hence, in order to prove the theorem, it is enough to show that:

1. after phase ecc of Stage 2 all nodes got the source message;
2. no messages are sent after phase $ecc + 1$ of Stage 2.

Both 1. and 2. follow from the fact that if $(s, v_1, ..., v_d = v)$ is a directed path of length $d \leq ecc$ from the source to a node v then node v_i hears the source message in phase at most i, and hence it acts as a transmitter in phase at most $i + 1$. □

It should be noted that although Algorithm Bound-and-Broadcast accomplishes radio broadcasting in time $O(n \cdot ecc)$, no node is aware of this, i.e., acknowledged radio broadcasting is not accomplished. However, we will show later that a refinement of this algorithm achieves ARB also in time $O(n \cdot ecc)$.

3.2 Radio broadcasting in arbitrary graphs

When collision detection is available, this feature can be used to code messages. Any node that acts as a receiver in a given round can distinguish between hearing signal μ and hearing nothing (background noise), which can thus serve as bits for the encoding. This observation yields the following simple scheme working for arbitrary graphs.

Algorithm Encoded-Broadcast

The algorithm works as follows. Suppose that σ is the source message and $(a_1, ..., a_r)$ is a binary representation of σ. The algorithm proceeds in phases. The first phase consists of one round, and each other phase contains $2r+4$ rounds. The second phase starts in round 2, and every other phase starts in the round following the end of the previous phase. In each phase some nodes are *active* and all others are *passive*. Passive nodes act as receivers, while nodes active in a given phase transmit the source message during this phase, as described below. In the first phase only the source is active, it acts as a transmitter and sends the message σ. In each subsequent phase k, those nodes that heard signal μ in at least one round of phase $k-1$ and have not heard it in any phase before $k-1$, become active, and all others are passive. We will maintain the invariant that all active nodes know σ. A node active in a given phase divides it into two-round segments $b_0, b_1, ..., b_r, b_{r+1}$. In both rounds of segments b_0 and b_{r+1} every active node acts as a transmitter and sends a contact message. Consider a segment b_i, for $0 < i < r+1$. If $a_i = 0$, active nodes act as a receiver in both rounds of segment b_i. If $a_i = 1$, active nodes act as receivers in the first round of segment b_i, and act as transmitters in the second round of this segment, sending a contact message. After phase k, all nodes that become active in phase $k+1$ determine the limits of phase k as the sequence of rounds between two pairs of consecutive rounds in which they heard signal μ. They divide this sequence of rounds into two-round segments, and reconstruct the message $\sigma = (a_1, ..., a_r)$ using the rule:

$a_i = 0$, if nothing was heard in both rounds of segment b_i; $a_i = 1$, if nothing was heard in the first round of segment b_i, and signal μ was heard in the second round of this segment.

Theorem 3.2 *Algorithm* Encoded-Broadcast *performs radio broadcasting of source message σ in an arbitrary graph, in time $O(|\sigma| \cdot ecc)$, where ecc is the eccentricity of the source and $|\sigma|$ is the number of bits in the message σ.*

Proof: Nodes at distance i from the source are active in phase $i+1$ and passive in all other phases. They decode message σ after phase $i+1$. □

3.3 Acknowledged radio broadcasting in symmetric graphs

As opposed to the scenario without collision detection, acknowledged radio broadcasting is feasible in our present setting. We first show that for symmetric graphs it can in fact be done in time $O(n)$. We show how to modify our Algorithm Explore-and-Expand in order to accomplish ARB, when collision detection is available.

Algorithm Explore-Expand-Check

The algorithm is a modification of Algorithm Explore-and-Expand. As before, it works in phases, each of them divided into stages A, B, and C. Stage A of phase k has one round more than previously, added at the very end of this stage. In this round, all nodes with labels larger than 2^k act as transmitters, sending a contact message, and all nodes with labels at most 2^k act as receivers. Every node with label at most 2^k that heard signal μ in this additional round, is called *warned*.

Stage B proceeds exactly as in Algorithm Explore-and-Expand. Stage C is modified as follows. When a node that is warned sends the token to a neighbor, it appends a *warning message* to the token, and the neighbor getting the token becomes warned. If, at the end of phase k, the source is not warned, it knows that RB has been completed.

Theorem 3.3 *Algorithm* EXPLORE-EXPAND-CHECK *performs acknowledged radio broadcasting in time $O(n)$, in any n-node symmetric graph.*

Proof: We show how to modify the proof of Theorem 2.1. It is enough to prove that if $2^{k-1} < n \leq 2^k$ then then the source is not warned at the end of phase k. Since there are no nodes with labels larger than 2^k, no node acts as a transmitter in the additional round of stage A of phase k. Hence no node is warned in stage C, and in particular, the source is not warned at the end of phase k. □

3.4 Acknowledged radio broadcasting in strongly connected graphs

We now show how to modify Algorithm BOUND-AND-BROADCAST in order to accomplish acknowledged radio broadcasting in time $O(n \cdot ecc)$ in any strongly connected n-node graph for which ecc is the eccentricity of the source. Note that strong connectedness is a necessary condition for feasibility of ARB. Indeed, there must exist a directed path from the source to any other node, for RB to be possible, and there must exist a directed path from any node to the source, in order to inform the source that the node got the message.

Algorithm BOUND-BROADCAST-CHECK

The algorithm is a refinement of Algorithm BOUND-AND-BROADCAST. We first give a high-level overview of the algorithm, pointing out the modifications with respect to Algorithm BOUND-AND-BROADCAST. First nodes learn an upper bound on n within a factor of 2, as before. Then each node learns its distance from the source. In the remaining part of the algorithm broadcasting phases of Algorithm BOUND-AND-BROADCAST are interleaved with acknowledgement phases in which nodes at increasing distances from the source send a contact message which is then propagated. The protocol finishes in the first phase in which no node sends an acknowledgement message. At this point all nodes can also detect that broadcasting is terminated.

We now give a more detailed description of the algorithm. It works in three stages. Stage 1 is identical as in Algorithm BOUND-AND-BROADCAST. Similarly as before, after its completion all nodes know an upper bound on n within a factor of 2. Let k_0 have the same meaning as before. By Lemma 3.2 we have $2^{k_0-1} < n \leq 2^{k_0}$.

Stage 2 lasts 2^{k_0} rounds, numbered 1,..., 2^{k_0}. Round 1 of Stage 2 follows the last round of Stage 1, so all nodes can identify it. Every node acts as a transmitter in at most one round of this stage, and sends a contact message in this round. In round 1 only the source acts as a transmitter. In round $i > 1$ those nodes which received signal μ in round $i-1$ for the first time, act as transmitters. We will show (cf. Lemma 3.3) that every node knows its distance from the source after stage 2.

The aim of Stage 3 is to broadcast the source message to all nodes, and simultaneously learn an upper bound on the eccentricity *ecc* of the source. This will make possible the termination of the algorithm after $O(n \cdot ecc)$ rounds, and informing the source (and in fact all nodes) that RB has been accomplished. Stage 3 starts in the round following the end of stage 2 and is divided into phases numbered by consecutive positive integers. Each phase consists of part 1 and part 2. Each of these parts lasts 2^{k_0} rounds, numbered 1,..., 2^{k_0}, separately for each part. The first round of part 2 follows the last round of part 1, for any phase. The first round of part 1 of the kth phase follows the last round of part 2 of the $(k-1)$th phase.

Part 1 of each phase of Stage 3 is devoted to broadcasting the source message. This is done similarly as in Stage 2 of Algorithm Bound-and-Broadcast. In each round, at most one node acts as a transmitter, thus collision is avoided. Every node acts as a transmitter in exactly one round, after receiving the source message. Let l_s denote the label of the source. In all rounds $i < l_s$ of the first part of the first phase, all nodes act as receivers. In round l_s of the first part of the first phase the source acts as a transmitter, sending the source message. Every other node with label l that got the source message in round r of the first part of phase j waits till round l of the first part of phase j, if $r < l$, and till round l of the first part of phase $j+1$, if $r \geq l$, and acts as a transmitter in this round, sending the source message.

Part 2 of each phase of Stage 3 is devoted to finding an upper bound on the eccentricity *ecc* of the source. This is done similarly as finding an upper bound on n in Stage 1. The difference is that instead of using labels of nodes we use their distances from the source, learned in Stage 2, and that now phases have equal length. A node is said to be *active* after the jth phase of Stage 3, $j > 1$, if it acted as a transmitter in some round of part 2 of this phase. All nodes are active in the first phase of Stage 3. Similarly as in Algorithm Bound-and-Broadcast, after each phase either all nodes are active or all are not active. In the first round of part 2 of the kth phase all active nodes whose distances from the source are larger than k act as transmitters, sending a contact message, and all others act as receivers. In every round $i > 1$ of part 2 of the kth phase all active nodes that heard signal μ in round $i-1$ of part 2 of this phase and did not hear it in any previous round of part 2 of this phase, act as transmitters, sending a contact message, and all others act as receivers. The first phase after which all nodes are not active terminates Stage 3 and also the entire algorithm. Every node can identify this phase as the first phase after which it is not active.

This completes the description of Algorithm Bound-Broadcast-Check. We will show that upon its completion, RB is accomplished and every node of the network is aware of it. In particular, ARB is accomplished as well. In order to prove correctness and analyze the efficiency of Algorithm Bound-Broadcast-Check we need the following lemmas.

Lemma 3.3 *After Stage 2 of Algorithm* Bound-Broadcast-Check *every node knows its distance from the source.*

Proof: Every node that acts as a transmitter in round $d+1$ of Stage 2 knows that its distance from the source is d. Now the lemma follows from $ecc \leq n \leq 2^{k_0}$. □

Lemma 3.4 *After phase k of Stage 3 all nodes at distance at most k from the source know the source message.*

Proof: Straightforward, by induction on k. □

Lemma 3.5 *Let p be the first phase of Stage 3 of Algorithm* Bound-Broadcast-Check *after which all nodes are not active. Then*

1. $p = ecc$, where ecc is the eccentricity of the source;

2. After phase p all nodes of the network know the source message.

Proof: 1. Since all nodes were active after phase $p-1$, we have $p-1 < ecc$. If $p < ecc$, all nodes would be active after phase p.

2. follows from 1. and from Lemma 3.4. □

Theorem 3.4 *Algorithm* BOUND-BROADCAST-CHECK *performs acknowledged radio broadcasting in any strongly connected graph. It works in time $O(n \cdot ecc)$ in any strongly connected n-node graph for which ecc is the eccentricity of the source.*

Proof: Every node can identify the first phase of Stage 3 after which all nodes are not active (phase p) as the first phase after which it is not active. This phase finishes the algorithm. By Lemma 3.5 all nodes know that RB is accomplished upon completion of the algorithm. In particular, the source has this knowledge, and hence ARB is accomplished. It remains to compute execution time of the algorithm. By Lemma 3.2, Stage 1 lasts at most

$$\sum_{i=1}^{k_0} 2^i < 2^{k_0+1} \leq 4n$$

rounds. Stage 2 lasts $2^{k_0} \leq 2n$ rounds. Stage 3 lasts $2p2^{k_0} = ecc \cdot 2^{k_0+1} \leq 4n \cdot ecc$ rounds. Hence the total execution time of Algorithm BOUND-BROADCAST-CHECK is $O(n \cdot ecc)$. □

References

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

[2] B. Awerbuch, A new distributed depth-first-search algorithm, Information Processing Letters 20, (1985), 147-150.

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

[4] R. Bar-Yehuda, O. Goldreich, and A. Itai, On the time complexity of broadcast in radio networks: an exponential gap between determinism and randomization, Journal of Computer and System Sciences 45 (1992), 104-126.

[5] R. Bar-Yehuda, A. Israeli, and A. Itai, Multiple communication in multihop radio networks, SIAM J. on Computing 22 (1993), 875-887.

[6] S. Basagni, D. Bruschi and I. Chlamtac, A mobility-transparent deterministic broadcast mechanism for ad hoc networks, IEEE/ACM Trans. on Networking 7 (1999), 799-807.

[7] S. Basagni, A.D. Myers and V.R. Syrotiuk, Mobility-independent flooding for real-time multimedia applications in ad hoc networks, Proc. 1999 IEEE Emerging Technologies Symposium on Wireless Communications & Systems, Richardson, TX.

[8] D. Bruschi and M. Del Pinto, Lower bounds for the broadcast problem in mobile radio networks, Distr. Comp. 10 (1997), 129-135.

[9] I. Chlamtac and A. Faragó, Making transmission schedule immune to topology changes in multi-hop packet radio networks, IEEE/ACM Trans. on Networking 2 (1994), 23-29.

[10] I. Chlamtac, A. Faragó and H. Zhang, Time-spread multiple access (TSMA) protocols for multihop mobile radio networks, IEEE/ACM Trans. on Networking 5 (1997), 804-812.

[11] I. Chlamtac and S. Kutten, On broadcasting in radio networks - problem analysis and protocol design, IEEE Transactions on Communications 33 (1985), 1240-1246.

[12] I. Chlamtac and S. Kutten, Tree-based broadcasting in multihop radio networks, IEEE Trans. on Computers 36 (1987), 1209–1223.

[13] I. Chlamtac and O. Weinstein, The wave expansion approach to broadcasting in multihop radio networks, IEEE Trans. on Communications 39 (1991), 426-433.

[14] B.S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc and W. Rytter, Deterministic broadcasting in unknown radio networks, Proc. 11th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'2000), 861-870.

[15] B.S. Chlebus, L. Gąsieniec, A. Östlin and J.M. Robson, Deterministic radio broadcasting, Proc. 27th Int. Coll. on Automata, Languages and Programming, (ICALP'2000), July 2000, Geneva, Switzerland, LNCS 1853, 717-728.

[16] M. Chrobak, L. Gąsieniec and W. Rytter, Fast broadcasting and gossiping in radio networks, Proc. 41st Symposium on Foundations of Computer Science (FOCS 2000), Redondo Beach, California, U.S.A., (2000), 575-581.

[17] A.E.F. Clementi, A. Monti and R. Silvestri, Selective families, superimposed codes, and broadcasting on unknown radio networks, Proc. 12th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'2001), 709-718.

[18] G. De Marco and A. Pelc, Faster broadcasting in unknown radio networks, Information Processing Letters, to appear.

[19] A. Dessmark and A. Pelc, Deterministic radio broadcasting at low cost, Proc. 18th Ann. Symposium on Theoretical Aspects of Computer Science (STACS 2001), February 2001, Dresden, Germany, LNCS 2010, 158-169.

[20] K. Diks, E. Kranakis, D. Krizanc and A. Pelc, The impact of knowledge on broadcasting time in radio networks, Proc. 7th European Symposium on Algorithms, ESA'99, Prague, Czech Republic, July 1999, LNCS 1643, 41-52.

[21] I. Gaber and Y. Mansour, Broadcast in radio networks, Proc. 6th Ann. ACM-SIAM Symp. on Discrete Algorithms, SODA'95, 577-585.

[22] R. Gallager, A perspective on multiaccess channels, IEEE Trans. on Information Theory 31 (1985), 124-142.

[23] L. Gargano, A. Pelc, S. Pérennes and U. Vaccaro, Efficient communication in unknown networks, Networks, to appear.

[24] L. Gąsieniec, A. Pelc and D. Peleg, The wakeup problem in synchronous broadcast systems, SIAM Journal on Discrete Mathematics, to appear.

[25] F.K. Hwang, The time complexity of deterministic broadcast radio networks, Discrete Applied mathematics 60 (1995), 219-222.

[26] E. Kranakis, D. Krizanc and A. Pelc, Fault-tolerant broadcasting in radio networks, Journal of Algorithms, to appear.

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

[28] E. Kushilevitz and Y. Mansour, Computation in noisy radio networks, Proc. 9th Ann. ACM-SIAM Symp. on Discrete Algorithms, SODA'98, 236-243.

[29] E. Pagani and G.P. Rossi, Reliable broadcast in mobile multihop radio networks, Proc. 3rd Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking, (MOBICOM'97), (1997), 34-42.

[30] K. Pahlavan and A. Levesque, Wireless Information Networks, Wiley-Interscience, New York, 1995.

[31] D. Peleg, Deterministic radio broadcast with no topological knowledge, manuscript (2000).

[32] A. Sen and M. L. Huson, A new model for scheduling packet radio networks, Proc. 15th Annual Joint Conference of the IEEE Computer and Communication Societies (IEEE INFOCOM'96) (1996), 1116 - 1124.

[33] A.S. Tanenbaum, Computer Networks, Prentice Hall, 1996.

[34] U. Vishkin, Deterministic sampling - A new technique for fast pattern matching, SIAM J. on Computing 20 (1991), 22-40.