Impact of Information on the Complexity of Asynchronous Radio Broadcasting

Tiziana Calamoneri * Emanuele G. Fusco * Andrzej Pelc †

Abstract

We consider asynchronous deterministic broadcasting in radio networks. An execution of a broadcasting protocol is a series of events, each of which consists of simultaneous transmitting or delivering of messages. The aim is to transmit the source message to all nodes of the network. If two messages are delivered simultaneously to a node, a collision occurs and this node does not hear anything. An asynchronous adversary may delay message deliveries, so as to create unwanted collisions and interfere with message dissemination. The total number of message transmissions executed by a protocol in the worst case is called the *work* of the protocol, and is used as the measure of its complexity. The aim of this paper is to study how various types of information available to nodes influence the optimal work of an asynchronous broadcasting protocol. This information may concern past events possibly affecting the behavior of nodes (adaptive vs. oblivious protocols), or may concern the topology of the network or some of its parameters. We show that decreasing the knowledge available to nodes may cause exponential increase of work of an asynchronous broadcasting protocol, and in some cases may even make broadcasting impossible.

keywords: algorithm, asynchronous adversary, deterministic broadcasting, unit disc graph, radio network.

1 Introduction

Radio networks and asynchronous adversaries. A *radio network* consists of stations with transmitting and receiving capabililities. The network is modeled as a directed graph with a distinguished node called the *source*. Each node has a distinct identity (label) which is a positive integer. If there is a directed edge from u to v, node v is called an *out-neighbor* of u and u is called an *in-neighbor* of v. At some time t a node may send a message to all of its out-neighbors. It is assumed that this message is delivered to all the out-neighbors simultaneously at some time $t' > t$ decided by an adversary that models unpredictable asynchronous behavior of the network. The only constraint (cf. [8, 20]) is that the adversary cannot collapse messages coming from the same node, i.e., two distinct messages sent by the same node have to be delivered at different times. We consider two types of asynchronous adversaries. The *strong* adversary, called the *node adversary* in [8], may choose an arbitrary delay $t' - t$ between sending and delivery, possibly different for every message. The *weak* adversary chooses an arbitrary delay for a given node (possibly different

*Computer Science Department, University of Rome "Sapienza", Roma, Italy. E-mail: calamo@di.uniroma1.it, fusco@di.uniroma1.it

†Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mail: pelc@uqo.ca

delays for different nodes), but must use this delay for all messages sent by this node during the protocol execution. The motivation for both adversaries is similar and follows the one given in [8]. Nodes of a radio network execute a communication protocol while concurrently performing other computation tasks. When a message arrives at a node, it is stored (prepared for transmission) and subsequently transmitted by it, the (unknown) delay between these actions being decided by the adversary; storing for transmission corresponds to sending and actual transmission corresponds to simultaneous delivery to all out-neighbors (at short distances between nodes the travel time of the message is negligible). The delay between storing and transmitting (in our terminology, between sending and delivery) depends on how busy the node is with other concurrently performed tasks. The strong adversary models the situation when the task load of nodes may vary during the execution of a broadcast protocol, and thus delay may vary from message to message even for the same node. The weak adversary models the assumption of a constant occupation load of each node during the communication process: some nodes may be more busy than others but the delay for a given node is constant.

At time t', a message is *heard*, i.e., received successfully by a node, if and only if, a message from exactly one of its in-neighbors is delivered at this time. If messages from two in-neighbors v and v' of u are delivered simultaneously at time t', we say that a *collision* occurs at u. Similarly as in most of the literature concerning algorithmic aspects of radio communication, we assume that in this case u does not hear anything at time t', i.e., we assume that a node cannot distinguish collision from silence.

While in general the network is modeled as an arbitrary directed graph, we also consider two natural smaller classes of networks. The first is modeled by *symmetric* directed graphs, or equivalently by undirected graphs. The second, still smaller class of networks is modeled by *unit disk graphs* (UDG) whose nodes are the stations. These nodes are represented as points in the plane. In the case of UDG networks, each node knows its Euclidean coordinates in the plane. These coordinates also play the role of the label (similarly as, e.g., in [14], nodes in UDG networks are not equipped with integer identities). Two nodes are joined by an (undirected) edge if their Euclidean distance is at most 1. Such nodes are called *neighbors*. It is assumed that transmitters of all stations have equal power which enables them to transmit at Euclidean distance 1, and that communication proceeds in a flat terrain without large obstacles. Hence the existence of an edge between two nodes indicates that transmissions of one of them can reach the other, i.e., these nodes can communicate directly. By contrast, arbitrary directed graphs are an appropriate model for radio networks deployed in a terrain with large obstacles and possibly varying power of transmitting devices.

Centralized vs. ad hoc broadcasting. We consider *broadcasting*, which is one of the basic communication primitives. In the beginning, one distinguished node, called the *source*, has a message which has to be transmitted to all other nodes. Remote nodes get the source message via intermediate nodes, along paths in the network. We assume that only stations that have already received the source message can send messages, hence broadcasting is equivalent to a process of waking up the network, when at the beginning only the source is awake. In order for the broadcasting to be feasible, we assume that there is a directed path from the source to any other node. For symmetric networks this is equivalent to connectivity. In this paper we consider only deterministic broadcasting algorithms.

Two alternative assumptions are made in the literature concerning broadcasting algorithms. It is either assumed that the topology of the underlying graph is known to all nodes, in which case nodes can simulate the behavior of a central monitor scheduling transmissions (*centralized broadcasting*), or it is assumed that the network topology is unknown to nodes (*ad hoc broadcasting*). Moreover, in the latter case, some crucial parameters of the network, such as the number n of nodes, may be known or unknown to nodes. In the case of UDG radio networks, an important parameter is the *density* d of the network, i.e., the smallest Euclidean distance between any two stations.

We will see how information about the topology of the network and knowledge of its parameters influence the efficiency of broadcasting protocols. In particular, for UDG networks, optimal work of broadcasting protocols may depend on the *granularity* g of the network defined as the inverse of its density.

Adaptive vs. oblivious protocols. We consider two kinds of broadcasting protocols: oblivious and adaptive. In an oblivious protocol every node has to send all its messages as soon as it is woken up by the source message. More precisely, a node has to commit to a non-negative integer representing the number of messages it will send during the broadcasting process, prior to the execution of the protocol. This number may depend only on the label of the node or on its position in the case of UDG networks. (In [8] only oblivious protocols were considered.) By contrast, an adaptive protocol is more powerful, as a node can decide on the number and content of messages it sends, depending on its history, i.e., depending on the sequence of messages received so far. Hence, while the total number of messages sent by an oblivious protocol is the same for each of its executions, for an adaptive protocol this number may differ depending on the behavior of the adversary.

We define the *work* of a broadcasting protocol as the worst-case total number of messages sent until all nodes are informed. The worst case is taken over all possible behaviors of an asynchronous adversary under consideration. Work is a natural measure of complexity of an asynchronous radio broadcast protocol. It was introduced in [8] for oblivious protocols. We will see that in some cases the rigidity of oblivious protocols may cause exponential increase of their work as compared to adaptive ones.

Our results. In the first part of the paper (Sections 3-5) we present our results on optimal work of asynchronous broadcasting against the strong adversary (i.e., the node adversary from [8]), see Table 1.

For UDG networks with known topology we get a tight result: the optimal work is $\Theta(\tau)$, where τ is the number of blocks containing at least one node. (Blocks form a partition of the plane into disjoint squares of side $1/\sqrt{2}$ – see Section 3 for a precise definition.) The result holds both for adaptive and for oblivious algorithms. Our upper bound is constructive: we show an oblivious broadcasting algorithm with work $O(\tau)$. For UDG networks with unknown topology the results significantly change and they depend on whether (a lower bound on) the density d of the network is known or not. If it is known, then optimal work depends on the number τ of occupied blocks and on the granularity $g = 1/d$. We show an oblivious broadcasting algorithm with work $O(\tau\alpha^{g^2})$, for some constant $\alpha > 1$. On the other hand, we show that any broadcasting algorithm, even adaptive, must use work $\Omega(\tau\beta^{g^2})$, for some constant $\beta > 1$. If d is unknown, we show that broadcasting against the strong adversary is impossible in UDG networks.

We now summarize our results for networks modeled by graphs that need not come from configurations of points in the plane. (For such networks we assume that all nodes have distinct positive integer labels and each node knows its label.) Symmetric radio networks with known topology are those in which optimal work of asynchronous broadcasting significantly depends on the adaptivity of the algorithm. Indeed, we prove that for adaptive algorithms the optimal work is $\Theta(n)$, where n is the number of nodes in the network. The upper bound is again constructive: we show an adaptive broadcasting algorithm with work $O(n)$ working for any n-node symmetric network of known topology. By contrast, using techniques from [8], it can be proved that any oblivious algorithm uses work $\Omega(c^n)$, for some constant $c > 1$, on some symmetric n-node network, and that there exists an oblivious algorithm working for any symmetric n-node network of known topology, using work $O(2^n)$. Hence we prove an exponential gap between optimal work required by adaptive and by oblivious broadcasting in symmetric networks of known topology. It should be noted that for arbitrary (not necessarily symmetric) networks, broadcasting with linear or even polynomial work is not always possible, even for adaptive algorithms. Indeed, it follows from [8] that exponential work (in the number n of nodes) is needed for some networks, even when the

<table>
<tr><th></th><th>UDG networks</th><th>Symmetric Networks</th><th>Arbitrary Networks</th></tr>
<tr><td>known topology</td><td>adaptive or oblivious:
$\Theta(\tau)$</td><td>adaptive: $\Theta(n)$
oblivious [8]:
$O(2^n)$
$\Omega(c^n)$, for some $c > 1$</td><td>adaptive or oblivious [8]:
$O(2^n)$
$\Omega(c^n)$, for some $c > 1$</td></tr>
<tr><td rowspan="2">unknown topology</td><td>known density d
adaptive or oblivious:
$O(\tau\alpha^{g^2})$, for some $\alpha > 1$
$\Omega(\tau\beta^{g^2})$, for some $\beta > 1$</td><td colspan="2" rowspan="2">adaptive or oblivious:
known or unknown N:
$\Theta(2^N)$</td></tr>
<tr><td>unknown density d
adaptive or oblivious:
impossible</td></tr>
</table>

Table 1: Optimal work of broadcasting against the strong asynchronous adversary. τ is the number of non-empty tiles, n is the number of nodes, N is the maximal label and g is the granularity of the UDG network ($g = 1/d$); c, α and β are constants.

topology is known and the algorithm is adaptive. It is also shown in [8] that, for radio networks of known topology, work $O(2^n)$ is always enough.

For networks of unknown topology we have a tight result on optimal work of asynchronous broadcasting. This work is $\Theta(2^N)$, where N is the maximal label of a node, and this result does not depend on whether the networks are symmetric or not, whether the algorithm is adaptive or not, and whether the maximal label N is known to nodes or not. More precisely, we show a lower bound $\Omega(2^N)$ on the required work, even for symmetric networks with known parameter N, and even for adaptive algorithms. On the other hand, we observe that an (oblivious) algorithm described in [8] and working for arbitrary networks without using the knowledge of N has work $O(2^N)$.

In Section 6 we present our results on optimal work of asynchronous broadcasting against the weak adversary. Introducing this adversary was motivated by the following remark in [8]: "It would be interesting to define a weaker, but still natural, model of asynchrony in radio networks, for which polynomial-work protocols always exist." We show that if nodes are equipped with clocks, then oblivious broadcasting algorithms using work $O(n)$ for n-node networks can always be provided in the presence of the weak asynchronous adversary. This is optimal, as witnessed by the example of the line network. Local clocks at nodes need not be synchronized, we only assume that they tick at the same rate. In fact, even this assumption can be removed in most cases: our algorithm works even when the ratio of ticking rates between the fastest and the slowest clock has an upper bound known to all nodes. The exception is the case of UDG networks of unknown density (for which broadcasting against the strong adversary was proved impossible). In this special case, our algorithm against the weak adversary assumes the same ticking rate of all clocks and relies on the availability of an object obtained non-constructively: if this object is given to nodes, they can perform oblivious broadcasting with work $O(n)$.

Due to lack of space, the proofs of several results were moved to the appendix.

Related work. Algorithmic aspects of radio communication were mostly studied under the assumption that communication is synchronous and using time as a complexity measure of the algorithms. These results can be partitioned into two subareas. The first deals with centralized communication, in which nodes have complete knowledge of the network topology and hence can simulate a central monitor (cf. [1, 4, 5, 17]). The second subarea assumes only limited (often local) knowledge of the topology, available to nodes of the network, and studies distributed communication in such networks with incomplete information.

The first paper to study deterministic centralized broadcasting in radio networks, assuming complete knowledge of the topology, was [4]. The authors also defined the graph model of radio network subsequently used in many other papers. In [5], an $O(D \log^2 n)$-time broadcasting algorithm was proposed for all n-node networks of diameter D. This time complexity was then gradually improved in several papers until the optimal time $O(D + \log^2 n)$ was obtained in [17]. On the other hand, in [1] the authors proved the existence of a family of n-node networks of constant diameter, for which any broadcast requires time $\Omega(\log^2 n)$.

Investigation of deterministic distributed broadcasting in radio networks whose nodes have only local knowledge of the topology was initiated in [2]. The authors assumed that each node knows only its own label and labels of its neighbors. Several authors [3, 6, 7, 9, 10, 11, 15] studied deterministic distributed broadcasting in radio networks under an even weaker assumption that nodes know only their own label (but not labels of their neighbors).

In [6, 7, 9, 11] the model of directed graphs was used. The aim of these papers was to construct broadcasting algorithms working as fast as possible in arbitrary (directed) radio networks without knowing their topology. The currently fastest deterministic broadcasting algorithms for such networks have running times $O(n \log^2 D)$ [11] and $O(n \log n \log \log n)$ [12]. On the other hand, in [10] an $\Omega(n \log D)$ lower bound on broadcasting time was proved for directed n-node networks of radius D.

Randomized broadcasting algorithms in radio networks were studied, e.g., in [2, 11, 19, 16]. The authors do not assume that nodes know the topology of the network or that they have distinct labels.

Another model of radio networks is based on geometry. Stations are represented as points in the plane and the graph modeling the network is no more arbitrary. It may be a unit disk graph, or one of its generalizations, where radii of disks representing areas that can be reached by the transmitter of a node may differ from node to node [13], or reachability areas may be of shapes different than a disk [18]. Broadcasting in such geometric radio networks and some of their variations was considered, e.g., in [13, 14, 18, 21]. The first paper to study deterministic broadcasting in arbitrary geometric radio networks with restricted knowledge of topology was [13]. In [14] the authors considered broadcasting in radio networks modeled by unit disk graphs.

Asynchronous radio broadcasting was considered, e.g., in [8, 20]. In [8] the authors studied three asynchronous adversaries (one of which is the same as our strong adversary), and investigated centralized oblivious broadcasting protocols working in their presence. They concentrated on finding broadcast protocols and verifying correctness of such protocols, as well as on providing lower bounds on their work. In [20] attention was focused on anonymous radio networks. In such networks not all nodes can be reached by a source message. It was proved that no asynchronous algorithm unaware of network topology can broadcast to all reachable nodes in all networks.

2 Terminology and preliminaries

A set S of positive integers is *dominated* if, for any finite subset T of S, there exists $t \in T$ such that t is larger than the sum of all $t' \neq t$ in T.

Lemma 2.1 *Let S be a finite dominated set and let k be its size. Then there exists $x \in S$ such that $x \geq 2^{k-1}$.*

Any oblivious broadcasting algorithm is fully determined by the number of messages sent by each node of the network. This non-negative integer is called the *send number* of the node. For any execution of a broadcasting algorithm, a *transmitter* is a node that sends at least one message in this execution. Hence, for an oblivious algorithm, a transmitter is a node with positive send number. The following lemma is a consequence of Lemma 1 from [8].

Lemma 2.2 *Consider any oblivious broadcasting algorithm $\mathcal{A}$. Let u be a node in the network. Let T be the set of transmitters in the in-neighborhood of u. If at least one element in T is informed by $\mathcal{A}$ and the set of send numbers of T is dominated, then u is eventually informed by $\mathcal{A}$.*

3 UDG radio networks

We recall the tilings of the plane defined in [14] by means of three different grids. Each of the three grids is composed of atomic squares with generic name *boxes*. The first grid is composed of boxes called *tiles*, of side length $d/\sqrt{2}$, the second of boxes called *blocks*, of side length $1/\sqrt{2}$, and the third one of boxes called *5-blocks*, of side length $5/\sqrt{2}$. All grids are aligned with the coordinate axes, each box includes its left side without the top endpoint and its bottom side without the right endpoint. Each grid has a box with the bottom left point with coordinates $(0, 0)$. Let τ be the number of non-empty blocks (i.e., blocks which contain at least one node).

Tiles are small enough to ensure that only one node can belong to a tile. Blocks are squares with diameter 1, i.e., the largest possible squares such that each pair of nodes in a square are able to communicate. 5-blocks are used to avoid collisions during communication: messages originating from central blocks of disjoint 5-blocks cannot cause collisions.

Every 5-block contains 25 blocks, while every block contains $\Theta\left(g^2\right)$ tiles. Blocks inside a 5-block and tiles inside a block are numbered with consecutive integers (starting from 0) left to right, top to bottom. Hence every tile is assigned a pair of integers (i, j) where i is the block number in the 5-block and j is the tile number in the block. (Tiles lying in more than one block are assigned more than one such pair. This is the case when $\sqrt{2}/n \neq d$ for all n.)

We say that two (distinct) blocks are *potentially reachable* from each other if they contain points at distance ≤ 1. Two blocks are *reachable* from each other if they contain nodes at distance ≤ 1. There are exactly 20 blocks that are potentially reachable from any given block.

3.1 Known topology

The following algorithm is oblivious, as it consists in an assignment of send numbers to nodes.
Algorithm `UDG1`.
For any pair of blocks (B, B') that are reachable from each other, Algorithm `UDG1` elects a pair of transmitters (b, b') s.t. $b \in B$, $b' \in B'$, and b is at distance at most 1 from b'. Any fixed strategy (e.g., taking the smallest such pair in lexicographic order of positions) is suitable to perform the election. Notice that at most 20 transmitters can be elected in every block.

Each elected transmitter in a 5-block is assigned a distinct label from the set $\mathcal{L} = \{0, 1, \ldots, 499\}$. This is done by partitioning the set $\mathcal{L}$ into 25 sets L_i of 20 labels each (in an arbitrary but fixed manner). Transmitters in the i-th block of any 5-block are assigned labels from set L_i. Labels in each block are assigned to transmitters in increasing order according to lexicographic order of their positions.

Assignment of send numbers is done as follows: each elected transmitter with label i is assigned send number 2^i. If the source has not been elected, it is assigned send number 1. All other nodes are assigned send number 0. ■

Lemma 3.1 *Algorithm* `UDG1` *successfully performs broadcast in any UDG radio network of known topology, with work in $O(\tau)$.*

Proof: We first prove the correctness of the algorithm. As the network is connected, either $\tau = 1$ or, for any non-empty block B, there must exist a sequence of block pairs $\langle(S, X_1), (X_1, X_2), \ldots,$

$(X_{k-1}, X_k), (X_k, B)\rangle$ such that S is the block containing the source and blocks in each pair are reachable from each other. If $\tau = 1$, all nodes in the unique non-empty block will be informed as soon as the message transmitted by the source is delivered, and algorithm `UDG1` successfully completes broadcasting with work 1. If $\tau > 1$, any non-empty block has at least one transmitter, and thus any node has a transmitter in its neighborhood. Moreover, every transmitter is connected to a transmitter located in S by a path containing only transmitters.

Consider an arbitrary node v and its block B, and consider the 5-block that has B in its center (this 5-block is not necessarily part of the 5-block grid). All neighbors of v are inside this 5-block. Blocks in this 5-block are assigned distinct numbers, and thus the set of send numbers assigned to transmitters in the neighborhood of v is dominated. It follows from Lemma 2.2 that node v will eventually receive the source message provided that at least one of the transmitters in its neighborhood will receive it. Hence it is enough to show that all transmitters receive the source message. This follows by induction on the length of a shortest path, in the subgraph induced by transmitters, between a transmitter in the block S and a transmitter in the neighborhood of v.

In order to estimate the work of the algorithm, notice that only a constant number of nodes in each block have a positive send number, and each send number is bounded by a constant. It follows that the total work is linear in the number τ of non-empty blocks. □

Lemma 3.2 *The work required to complete broadcast in any UDG radio network is in* $\Omega(\tau)$.

Lemma 3.1 and Lemma 3.2 imply the following theorem.

Theorem 3.1 *The optimal work required to complete broadcast in any UDG radio network of known topology is* $\Theta(\tau)$.

3.2 Unknown topology

When the topology of the network is unknown, elections of transmitters cannot be performed without message exchanges. Here the scenario is different depending on whether (a lower bound on) the density d of the network is known or not.

The following algorithm assumes that each node is provided with the value of d. Similarly as Algorithm `UDG1` it is oblivious.

Algorithm `UDG2`.

The algorithm is based on the tilings from [14] defined in the beginning of Section 3, and works in a similar manner as Algorithm `UDG1`. The set $\mathcal{L}$ of labels is now composed of integers from the interval $\left[0, \ldots, 25 \cdot \left(\lceil\sqrt{2}/d\rceil + 1\right)^2 - 1\right]$, and it is partitioned in 25 sets L_i, each of size $\left(\lceil\sqrt{2}/d\rceil + 1\right)^2$. All nodes in the network are transmitters, and each node in a 5-block gets a distinct label according to the numbering of the tile and the block it belongs to. More precisely, a node in the tile that is assigned the pair of integers (i, j) gets the label that is the jth element of L_i. Recall that there can be tiles which are partially contained in more than one block. In any case, the only node which can be contained in the tile belongs to only one block and thus its label is uniquely determined.

The send number of each node with label i is set to 2^i. ■

Proposition 3.1 *Algorithm* `UDG2` *successfully performs broadcast in any UDG radio network of unknown topology and known density* d *with work in* $O\left(\tau\alpha^{g^2}\right)$*, for some constant* $\alpha > 1$.

We now turn attention to the lower bound on the work of a broadcasting algorithm.

Theorem 3.2 *The work required to complete broadcast in any UDG radio network of unknown topology and known density* d *is in* $\Omega\left(\tau\beta^{g^2}\right)$*, for some constant* $\beta > 1$.

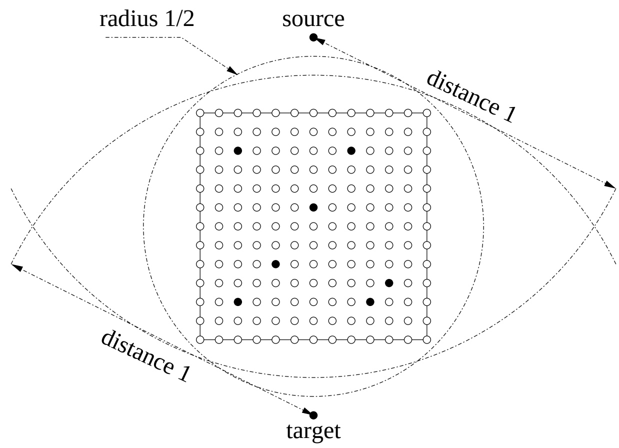

Figure 1: A network of the class $\mathcal{N}$ used in the proof of Theorem 3.2 .

Proof: Consider the class $\mathcal{N}$ of networks depicted in Figure 1. The source occupies position $(0, 1.2)$ and the target occupies position $(0, 0)$. Nodes in the central part of the network are situated in an arbitrary subset of vertices of the largest regular square grid of side length d, contained in the intersection of the circles of radius 1 centered in the source and in the target, and of the circle of radius $1/2$ centered in $(0, 0.6)$. Notice that there are $\Theta\left(g^2\right)$ vertices in the grid.

The set Q of nodes situated in the grid forms a clique, and each node in Q is within distance 1 from the source and from the target. It follows that a network in $\mathcal{N}$ is connected if and only if Q is nonempty.

All nodes in Q become informed as soon as the first message sent by the source is delivered. When the first message from an informed node in Q is delivered without colliding with any delivery from other nodes in Q, broadcasting is completed successfully.

It follows that, until the completion of broadcasting, the only events that are perceived by nodes in Q are determined by deliveries of messages sent by the source. The source and the target will not receive any message until the completion of broadcasting.

Consider an arbitrary adaptive algorithm $\mathcal{A}$. $\mathcal{A}$ is forced to provide a send number for the source, and it is not able to modify this number until the end of the execution (no event is perceived by the source). The adversary delays all deliveries of nodes in Q until all messages from the source have been delivered, thus guaranteeing that no node in Q can perceive an event between the first delivery of one of its messages and the end of broadcasting.

This allows us to treat $\mathcal{A}$ as an oblivious algorithm, which is obliged to provide send numbers to all nodes in the network once and forever. In fact we can assume that the algorithm assigns send numbers to vertices in the grid (a node occupying vertex p is assigned the respective send number).

Now consider a vertex p of the grid. If algorithm $\mathcal{A}$ assigns send number 0 to p, then $\mathcal{A}$ is unsuccessful in the network $N \in \mathcal{N}$ where the set Q contains only the node in vertex p. It follows that all vertices in the grid have to be assigned positive send numbers.

If the set of send numbers, assigned by $\mathcal{A}$ to vertices of the grid, is not dominated, then there exists a set T of vertices for which the largest send number x, corresponding to vertex p_0, is at most equal to the sum of all others. The adversary can make $\mathcal{A}$ unsuccessful on the network $N \in \mathcal{N}$ in which nodes in Q occupy exactly vertices from T, by letting all deliveries collide. This can be done as follows. The deliveries of messages from the node in vertex p_0 are done at times $t_1 < t_2 < \ldots < t_x$. Every other message can be delivered at one of those time points, so that at each time point t_i at least two messages are delivered.

This contradiction shows that the set of send numbers, assigned by $\mathcal{A}$ to vertices of the grid must be dominated. As the set of vertices in the grid is of size $\Theta\left(g^2\right)$ and, by Lemma 2.1, any dominated set on k elements contains a number $\geq 2^{k-1}$, it follows that any algorithm working correctly on all networks in $\mathcal{N}$ requires work in $\Omega\left(\beta^{g^2}\right)$, for some constant $\beta > 1$. By arranging networks of class $\mathcal{N}$ in a chain of length τ, we get a lower bound on work in $\Omega\left(\tau\beta^{g^2}\right)$ □

All results of this subsection remain valid if, instead of density d of the network, only a lower bound d' on d is known to nodes. In this case, in the formulae for the upper and lower bounds on the work, the parameter $g = 1/d$ should be replaced by $g' = 1/d'$. If nothing is known about d, however, broadcasting in UDG radio networks turns out to be impossible, as shown in the following theorem.

Theorem 3.3 *Broadcast in UDG radio networks of unknown topology and unknown density is impossible.*

4 Symmetric networks of known topology

In symmetric networks of known topology we prove an exponential gap between the work of adaptive and oblivious algorithms. Indeed, while an adaptive algorithm can complete broadcasting on n-node symmetric networks with work in $O(n)$, an oblivious algorithm requires work in $\Omega\left(c^n\right)$, for some constant $c > 1$ (cf. [8]).

4.1 Adaptive broadcast

The following algorithm is adaptive. Each node decides if it sends a message, after each perceived event.

Algorithm `SYM`.

Knowing the topology of the network, all nodes compute the same spanning tree T, rooted at the source. Notice that, even assuming that the source is unknown to other nodes in the network, this information can be appended to the source message and thus it can be made available to each node when it is woken up by the first received message.

All internal nodes of the spanning tree T are then explored in a depth first search manner, using token-based communication in order to avoid collisions. A message is sent only after the previous message has been delivered. Algorithm `SYM` ends when the token is sent back to the source by its last internal child. ■

Lemma 4.1 *Algorithm* `SYM` *successfully performs broadcast in any n-node symmetric radio network of known topology with work in $O(n)$.*

As the optimal work to perform broadcasting on the n-node line is $n-1$, we have the following theorem.

Theorem 4.1 *The optimal work required to complete broadcasting in all n-node symmetric radio networks of known topology is $\Theta(n)$.*

4.2 Oblivious broadcast

An oblivious algorithm, performing broadcasting in *any* n-node connected radio network of known topology (not necessarily symmetric) can be obtained by arranging nodes in increasing order of labels, and assigning send number 2^{i-1} to the ith node. Such an algorithm can be proved to be correct by induction on the length of a shortest path connecting the source to an arbitrary node v, using Lemma 2.2. The work required to complete broadcasting by this algorithm is in $O(2^n)$.

In [8], the following network class has been introduced in order to prove that oblivious broadcasting algorithms against a more powerful adversary require work in $\Omega(c^n)$, for some constant $c > 1$.

Networks in the above mentioned class contain $\binom{k}{3} + k + 1$ nodes, for integers $k > 0$. Nodes are partitioned in three layers: the first layer contains the source, the central layer contains k nodes, while the third layer contains the remaining $\binom{k}{3}$ nodes. Edges in these networks connect the source to all nodes in the second layer, while each node in the third layer is connected to a distinct subset of 3 nodes choosen among those in the second layer. Even though edges were oriented away from the source in [8], the same proof remains valid for oblivious algorithms even if the network is made symmetric, and even against our strong adversary (which was called the node adversary in [8]).

Since the upper bound $O(2^n)$ holds for arbitrary networks and the lower bound $\Omega(c^n)$ holds even for symmetric networks, we have the following theorem.

Theorem 4.2 *The optimal work of an oblivious algorithm, which completes broadcasting in radio networks of known topology, is in $O(2^n)$ and in $\Omega(c^n)$, for some constant $c > 1$, both for symmetric and for arbitrary networks.*

5 Networks of unknown topology

For networks of unknown topology we prove matching upper and lower bounds on the optimal work of broadcasting algorithms. The upper bound we show is based on the oblivious algorithm described below, which works correctly on any network (not necessarily symmetric) containing a directed path from the source to every node. The lower bound, on the other hand, holds even on symmetric networks and for all algorithms, including the adaptive ones.

An oblivious algorithm performing broadcasting in any connected radio network of unknown topology, is obtained by assigning to node with label i send number 2^{i-1}. The algorithm works in the same manner as the one for known topology networks introduced in the previous section, but its work, instead of depending on the number of nodes of the network, depends on the largest label N appearing in the network. (N need not be known to nodes.) Thus the work of this algorithm is in $O\left(2^N\right)$. This work is proved to be optimal by the following lemma.

Lemma 5.1 *The work required to complete broadcasting in any symmetric radio network of unknown topology is in $\Omega(2^N)$, where N is the largest label that appears in the network.*

6 Broadcasting against the weak adversary

In this section we present our results on the work of asynchronous broadcasting against the weak adversary. Recall that this adversary may delay delivery of messages sent by various nodes by arbitrary and unknown time intervals that may vary between nodes, but are equal for all messages sent by a given node. In this section we assume that nodes are equipped with local clocks. These clocks need not be synchronized. In one algorithm, working for UDG networks with unknown density, we assume that they tick at the same rate, and in the other, working for UDG networks with known (lower bound on) the density and also working for arbitrary networks with distinct positive integer labels, we weaken even this assumption and require only that all nodes know an upper bound on the ratio of ticking rates between the fastest and the slowest clock.

The idea of broadcasting algorithms working against the weak adversary comes from the observation that since delivery delay must be the same for all messages sent by a given node, if a node sends two messages at some time interval t, this interval may only be shifted by the adversary when delivering messages, but its length must be kept intact. Thus, using exponential intervals between just two messages sent by every node (where the exponent depends on the node label), blocking of messages can be prevented similarly as sending an exponential *number* of messages permitted preventing blocking by the strong adversary. (This is a similar work-for-time trade-off as, e.g., that in the Time-Slicing algorithm for leader election on the ring.) Due to the above possibility we can restrict the number of messages sent by every node to just 2, and thus use linear work.

We first describe an oblivious broadcasting algorithm working for networks of unknown topology whose nodes are labeled with distinct positive integers. In this algorithm we make a very weak assumption: not only clocks of nodes need not be synchronized, but they need not tick at the same rate, as long as the upper bound α on the ratio of ticking rates between the fastest and the slowest clock is known to all nodes. Without loss of generality we may assume that $\alpha \geq 2$.

Algorithm `Time-Intervals`

The source sends the message once. Upon receiving the source message, any node with label i, different from the source, sends two messages at time interval $4^{i\alpha}$ on its local clock. ■

Theorem 6.1 *Algorithm* `Time-Intervals` *successfully performs broadcast in an arbitrary n-node network, with work in $O(n)$.*

We now turn attention to broadcasting against the weak adversary in UDG networks. First notice that if the topology of the network is known, then Algorithm `UDG1` clearly works correctly against the weak adversary as well, and it uses the same work $O(\tau)$, which is at most $O(n)$ for n-node networks. Thus we may restrict attention to networks with unknown topology. If a lower bound on the network density is known to all nodes, then we may use the same tiling as in Algorithm `UDG2` to obtain integer labels of all nodes of the network. Subsequently we use Algorithm `Time-Intervals` and the same argument as before proves its correctness and work complexity.

The only remaining case is that of UDG radio networks in which nothing is known about the density. Recall that in this case we proved that broadcasting against the strong adversary is impossible. Somewhat surprisingly, we will show that if the adversary is weak, then broadcasting in n-node UDG networks with unknown density can be performed with work in $O(n)$. Our algorithm, however, is only of theoretical interest: its main goal is to show a situation when broadcasting is impossible against the strong adversary, but can be done using linear work against the weak adversary. The impracticality of the algorithm has two reasons. First, since it works on networks of arbitrarily small density, it requires infinite precision of the perception of Euclidean coordinates by nodes. Second, the algorithm is non-constructive: it relies on the availability of a function whose existence we prove, but which is not constructed. Once this function is given to nodes, they can perform easy broadcasting with linear work. More precisely, our algorithm relies on the following set-theoretic lemma.

Lemma 6.1 *There exists a function $f : \mathbf{R} \times \mathbf{R} \longrightarrow \mathbf{R}^+$ such that any distinct elements $v_1, \ldots, v_k$ and $w_1, \ldots, w_r$ from $\mathbf{R} \times \mathbf{R}$ satisfy the inequality $\pm f(v_1) \pm \cdots \pm f(v_k) \neq \pm f(w_1) \pm \cdots \pm f(w_r)$.*

The broadcasting algorithm for UDG networks with unknown density assumes that all nodes have clocks ticking at the same rate. Given the function f whose existence follows from Lemma 6.1, the algorithm can be formulated as follows.

Algorithm `Non-Constructive`

The source sends the message once. Upon receiving the source message, any node with Euclidean coordinates (x, y), different from the source, sends two messages at time interval $f(x, y)$. ■

Theorem 6.2 *Algorithm* `Non-Constructive` *performs correct broadcasting in an arbitrary n-node UDG network, using work $O(n)$.*

References

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

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

[3] Bruschi, D., Del Pinto, M.: Lower bounds for the broadcast problem in mobile radio networks. *Distributed Computing* **10**, (1997), 129–135.

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

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

[6] Chlebus, B., Gąsieniec, L., Gibbons, A., Pelc, A., Rytter, W.: Deterministic broadcasting in unknown radio networks. *Distributed Computing* **15**, (2002), 27–38.

[7] Chlebus, B., Gąsieniec, L., Östlin, A., Robson, J.M.: Deterministic radio broadcasting. In *Proc. 27th Int. Colloq. on Automata, Languages and Programming* (ICALP 2000), LNCS 1853, 717–728.

[8] Chlebus, B., Rokicki, M.: Centralized asynchronous broadcast in radio networks. *Theoretical Computer Science* 383 (2007), 5–22.

[9] Chrobak, M., Gąsieniec, L., Rytter, W.: Fast broadcasting and gossiping in radio networks. In *Proc. 41st Symp. on Foundations of Computer Science* (FOCS 2000), 575–581.

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

[11] Czumaj, A., Rytter, W.: Broadcasting algorithms in radio networks with unknown topology. In *Proc. 44th Symp. on Foundations of Computer Science* (FOCS 2003), 492–501.

[12] De Marco, G.: Distributed broadcast in unknown radio networks. In *Proc. 19th ACM-SIAM Symp. on Discrete Algorithms* (SODA 2008).

[13] Dessmark, A., Pelc, A.: Broadcasting in geometric radio networks. *J. of Discrete Algorithms* **5** (2007), 187–201.

[14] Emek, Y., Gasieniec, L., Kantor, E., Pelc, A., Peleg, D., Su, C., Broadcasting time in UDG radio networks with unknown topology. In *Proc. 26th Ann. ACM Symposium on Principles of Distributed Computing* (PODC 2007), 195-204.

[15] Kowalski, D., Pelc, A.: Time complexity of radio broadcasting: adaptiveness vs. obliviousness and randomization vs. determinism. *Theoretical Computer Science* **333**, (2005), 355–371.

[16] Kowalski, D., Pelc, A.: Broadcasting in undirected ad hoc radio networks. *Distributed Computing* **18**, (2005), 43–57.

[17] Kowalski, D., Pelc, A.: Optimal deterministic broadcasting in known topology radio networks. *Distributed Computing* **19**, (2007), 185–195.

[18] Kranakis, E., Krizanc, D., Pelc, A.: Fault-tolerant broadcasting in radio networks, *J. of Algorithms* **39**, (2001), 47–67.

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

[20] A. Pelc, Activating anonymous ad hoc radio networks, Distributed Computing 19 (2007), 361-371.

[21] Sen, A., Huson, M. L.: A New Model for Scheduling Packet Radio Networks, In *Proc. 15th Joint Conf. of the IEEE Computer and Communication Societies* (IEEE INFOCOM 1996), 1116 - 1124.

APPENDIX

Proof of Lemma 2.1

The proof is by induction on the size k of S. If $k = 1$ then $2^0 = 1$ and the basis of induction holds.

If a set is dominated, all its subsets are dominated. By the inductive hypothesis every subset of S of size $i < k$ contains an element $x \geq 2^{i-1}$. It follows that arranging elements in S in increasing order we have $x_i \geq 2^{i-1}$, for $1 \leq i \leq k-1$. Then $\sum_{i=1}^{k-1} x_i \geq \sum_{i=1}^{k-1} 2^{i-1} = 2^{k-1} - 1$. As x_k is the largest element in S and S is dominated, we have $x_k \geq \sum_{i=1}^{k-1} x_i > 2^{k-1} - 1$, which proves the lemma.

Proof of Lemma 3.2

The proof follows from the fact that at least one node in every non empty 5-block has to transmit at least once.

Proof of Proposition 3.1

The correctness of the algorithm follows from Lemma 2.2 by induction on the length of a shortest path from the source to an arbitrary node v.

The work of the algorithm in every block is upper bounded by $2^{\left(\lceil\sqrt{2}/d\rceil+1\right)^2}$. As $\lceil\sqrt{2}/d\rceil \in \Theta(g)$, the lemma follows.

Proof of Theorem 3.3

Consider the class $\mathcal{C}$ of networks depicted in Figure 2. Networks in $\mathcal{C}$ are similar to networks in class $\mathcal{N}$, defined in the proof of Theorem 3.2. In particular, the source and the target are located in the same positions, while the set Q of nodes is an arbitrary finite set of points in the plane, contained in the square S of side $1/2$, centered at $(0, 0.6)$. A network $C \in \mathcal{C}$ is connected if and only if Q is non empty. By following the reasoning of the proof of Lemma 3.2, we can show that any adaptive algorithm $\mathcal{A}$ can be treated as an oblivious one when working on a network in $\mathcal{C}$. Algorithm $\mathcal{A}$ can then be identified with a function $f : S \mapsto \mathbf{N}$ which assigns send numbers to points in the square.

First assume that the range of f is infinite and suppose that broadcasting ends with work T. This leads to a contradiction, as we can always choose a network $C \in \mathcal{C}$ with 2 nodes in Q located in two points of S that are mapped to values larger than T. By scheduling the first T deliveries of messages sent by these two nodes in the same time points, the adversary can delay completion of broadcasting until the overall work of nodes in C is at least $2T + 1$, while we assumed the total work to be exactly T.

Hence the range of f must be finite. If $f(z) = 0$, for some point $z \in S$, then broadcasting is unsuccessful on the network C in which Q contains only one node located in z. It follows that all points of S have to be mapped by f into positive integers. Then there must exist two points, x and y, such that $f(x) = f(y)$. If this is the case, the adversary can make the algorithm unsuccessful on the network C where Q contains two nodes, one in the point x and the other in the point y, by

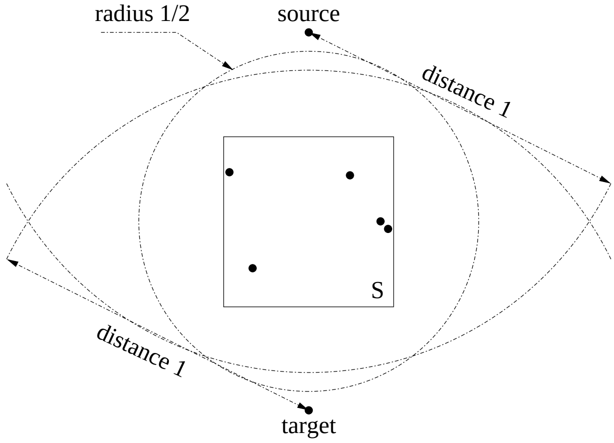

Figure 2: A network of the class $\mathcal{C}$ used in the proof of Theorem 3.3.

delivering messages sent by these two nodes at the same time points.

Proof of Lemma 4.1

We first prove correctness of Algorithm `SYM`. Since any message is sent only after the previous message has been delivered, it follows that no collision can occur during the execution of broadcasting. As all internal nodes in T transmit at least once, and T is a spanning tree of the network, all nodes will eventually receive the source message.

Since the token traverses every edge of T either 0 or 2 times, the total work of the algorithm is smaller than $2n \in O(n)$.

Proof of Lemma 5.1

To prove the lemma, consider the following class $\mathcal{Z}$ of networks. Networks in the class $\mathcal{Z}$ contain a source, a target and a set R of nodes. Each node in R is connected to the source s and to the target t. The source has label 1. Nodes in $R \cup \{t\}$ are labeled with distinct integers larger than 1, and N is the largest label appearing in $R \cup \{t\}$. R has to be non-empty, as otherwise the network would be disconnected.

The rest of the proof is based on the same idea as the proof of Lemma 3.2. Labels larger than 1 play the role of vertices in the grid.

As soon as a node in R delivers a message to the target without collisions, broadcasting in any network $Z \in \mathcal{Z}$ is completed. Hence, we can treat any adaptive algorithm $\mathcal{A}$ as an oblivious one, when working on networks in $\mathcal{Z}$. It follows that algorithm $\mathcal{A}$ has to assign a send number to any integer larger than 1 (which is a potential label of a node in R).

If there exists a label $\ell > 1$ such that $\mathcal{A}$ assigns send number 0 to ℓ, then $\mathcal{A}$ is unsuccessful on the network $Z \in \mathcal{Z}$ where the only node in R is labeled ℓ. It follows that $\mathcal{A}$ has to assign positive send numbers to all integers larger than 1. (Even if the maximum label N is known to $\mathcal{A}$, there is no guarantee that any particular label is assigned to a node in R, as N can be assigned to the target.) If the set of send numbers is not dominated, the adversary can make the algorithm $\mathcal{A}$ unsuccessful on the network $Z \in \mathcal{Z}$ where the (finite) set of send numbers assigned to nodes in R does not contain an element which is larger than the sum of all others (cf. the proof of Lemma 3.2).

As $R \cup \{t\}$ can contain up to $N-1$ nodes, the lemma follows from Lemma 2.1.

Proof of Theorem 6.1

Since any node sends at most two messages, the work used is in $O(n)$. It remains to prove the correctness of the algorithm.

Fix the slowest ticking rate among all local clocks and call it *universal*. In the rest of the proof we will use only the universal ticking rate. Since α is the ratio of ticking rates between the fastest and the slowest clock, the (universal) time interval used by node with label i is $T_i = \frac{4^{i\alpha}}{\beta}$, where $1 \leq \beta \leq \alpha$. Fix a node u and its in-neighbors $v_1, \ldots, v_k$ that got the source message. Without loss of generality, assume that nodes v_i are ordered in increasing order of interval lengths T_i. The delivery times of messages sent by nodes v_i are $x_i, x_i + T_i$, for $i = 1, \ldots, k$. In order to prove that at least one of these messages will be heard by node u, it is enough to show that $T_k > T_1 + \cdots + T_{k-1}$. Hence it is enough to show that

$$\frac{4^{k\alpha}}{\alpha} > 4^{\alpha} + 4^{2\alpha} + \cdots + 4^{(k-1)\alpha}. \tag{1}$$

We have $\alpha^{1/\alpha} < 3$, hence

$$\frac{4^k}{\alpha^{1/\alpha}} > \frac{4}{3} \cdot 4^{k-1} > 4^1 + \cdots + 4^{k-1},$$

hence

$$\frac{4^{k\alpha}}{\alpha} > (4^1 + \cdots + 4^{k-1})^{\alpha} > 4^{\alpha} + 4^{2\alpha} + \cdots + 4^{(k-1)\alpha},$$

which proves inequality (1) and concludes the proof by induction on the length of the shortest path from the source to a given node.

Proof of Lemma 6.1

Let κ be the cardinal of the continuum. Hence the cardinality of sets $\mathbf{R} \times \mathbf{R}$ and $\mathbf{R}^+$ is κ. Using the axiom of choice (this is the non-constructive ingredient in the definition of the function f), order the set $\mathbf{R} \times \mathbf{R}$ in ordinal type κ. Let $x_\gamma : \gamma < \kappa$ be this ordering. We now define the function f by transfinite induction. Suppose that $f(x_\gamma)$ is already defined, for all $\gamma < \delta$. Consider the set Z of all reals $\pm f(x_{\gamma_1}) \pm \cdots \pm f(x_{\gamma_d})$, for any finite set $\{x_{\gamma_1}, \ldots, x_{\gamma_d}\}$ of elements of $\mathbf{R} \times \mathbf{R}$, such that $\gamma_1, \ldots, \gamma_d < \delta$. The set Z has cardinality equal to the maximum of the cardinality of δ and of $\aleph_0$ (the latter is the cardinality of the set of natural numbers). Hence the cardinality of Z is strictly less than κ, and consequently there exists a number $z \in \mathbf{R}^+ \setminus Z$. We put $f(x_\delta) = z$.

Thus the function f is defined by transfinite induction. It remains to verify that it has the desired property. Suppose by contradiction that some elements $v_1, \ldots, v_k$ and $w_1, \ldots, w_r$ from $\mathbf{R} \times \mathbf{R}$ satisfy the equality $\pm f(v_1) \pm \cdots \pm f(v_k) = \pm f(w_1) \pm \cdots \pm f(w_r)$. Let ξ be the largest index of all these elements in the ordering $x_\gamma : \gamma < \kappa$. It follows that $f(x_\xi) = \pm f(x_{\gamma_1}) \pm \cdots \pm f(x_{\gamma_d})$, for some $\gamma_1, \ldots, \gamma_d < \xi$, which contradicts the definition of $f(x_\xi)$.

Proof of Theorem 6.2

As before, the complexity of the algorithm is straightforward. It remains to prove its correctness. Suppose that there exists a network with a node u that has in-neighbors $v_1, \ldots, v_k$ that got the source message. Suppose that there exist delays such that the adversary can shift time segments of lengths $f(v_1), \ldots, f(v_k)$ between messages sent by these nodes, so that all message deliveries are blocked by collisions. This implies that, for some nodes $w_1, \ldots, w_r, u_1, \ldots u_m \in \{v_1, \ldots, v_k\}$ we must have $f(w_1) + \cdots + f(w_r) = f(u_1) + \cdots + f(u_m)$, which contradicts the property of the function f established in Lemma 6.1. This contradiction shows that all nodes in every UDG network will eventually get the source message.