

Deterministic Local Problems in Radio Networks: On the Impact of Local Domination and a Bit of Advice

Pawel Garncarek 

Institute of Computer Science, University of Wrocław, Poland

Tomasz Jurdzinski 

Institute of Computer Science, University of Wrocław, Poland

Dariusz R. Kowalski 

School of Computer and Cyber Sciences, Augusta University, GA, USA

Shay Kutten 

Department of Data and Decision Sciences and the Grand Technion Energy Program, Technion,
Haifa, Israel

Miguel A. Mosteiro 

Computer Science Department, Pace University, New York, NY, USA

Abstract

Radio Networks (RN) is one of the fundamental models for network communication where nodes can broadcast messages locally but their simultaneous transmissions can interfere with each other at their shared neighbors. This work focuses on performing the very fundamental primitive of Local Broadcast, in spite of the interferences.

We investigate to what extent local knowledge, called *advice*, relating to the 2-local domination number γ_2 may speed up Local Broadcast. Specifically for each node and some dominating set, knowledge about some neighboring dominating node and the local number among the neighbors of that dominating node. We show that such advice is sufficient to build an efficient oblivious transmission schedule. Along those lines, we present three algorithms trading the level of adaptiveness (from oblivious to adaptive) for bits of advice per node (from $O(\log(\Delta\gamma_2))$ to 1). All our algorithms complete Local Broadcast in $\tilde{O}(\Delta\gamma_2^2)$ rounds, where Δ is the maximum degree of the network.

On the side of lower bounds, we show that, for each quasi-adaptive deterministic Local Broadcast algorithm, there is some RN that requires $\Omega(\min\{(\min\{\Delta, \gamma_2\}/\log n)^2, n\})$ communication rounds, where n is the number of network nodes. In quasi-adaptive protocols nodes may stop executing once its computational task is completed. To the best of our knowledge, this is the first (nearly) quadratic Local Broadcast (same message for all neighbors) lower bound in the RN model. Our lower bound is stronger than previous works in multiple ways: *i*) it is nearly quadratically better than the best known general lower bound for this class of algorithms, *ii*) it applies to a wider class of algorithms than previous work for fully oblivious, *iii*) it achieves similar time lower bound than previous work proved for a much more demanding Local Broadcast where each node sends a *possibly different* message to each neighbor, and *iv*) it takes into account the local domination parameter γ_2 .

2012 ACM Subject Classification Theory of computation $\rightarrow$ Distributed algorithms

Keywords and phrases Radio Networks, Local Broadcast, Distributed Deterministic Algorithms, Lower Bounds, Graph algorithms, Advice, Labeling Schemes, Local Domination

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.34

Funding *Pawel Garncarek*: Supported by the National Science Center, Poland (NCN), grant 35 2020/39/B/ST6/03288.

Tomasz Jurdzinski: Supported by the National Science Center, Poland (NCN), grant 35 2020/39/B/ST6/03288.



© Pawel Garncarek, Tomasz Jurdzinski, Dariusz R. Kowalski, Shay Kutten, and Miguel A. Mosteiro;
licensed under Creative Commons License CC-BY 4.0

36th International Symposium on Algorithms and Computation (ISAAC 2025).

Editors: Ho-Lin Chen, Wing-Kai Hon, and Meng-Tsung Tsai; Article No. 34; pp. 34:1–34:20

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Shay Kutten: The work of this author was funded in part by grant 1346/22 from the Israeli Science 37 Foundation. His research was carried out within the framework of the Grand Technion Energy 38 Program (GTEP). A part of this work was done while with the Fraunhofer Darmstadt.

Miguel A. Mosteiro: Partially supported by Pace SRC grant and Kenan Fund.

1 Introduction

The Radio Networks (RN) model was introduced about 40 years ago by Chlamtac and Kutten [9] as a multi-hop generalization of a multiple-access channel [1, 41]. The model attracted a lot of attention over many years. A somewhat arbitrary list of examples includes [2, 3, 39, 12, 43, 25, 15, 23] among many others. One major difference between RN and, e.g., the CONGEST model of point-to-point communication [42], is that if a node v transmits when v 's two-hop neighborhood does not, then *all* of v 's neighbors receive copies of v 's message, rather than just one of them. (The model is defined later.) Hence, the *Local Broadcast* task of sending messages to all the neighbors arises as a natural building block for RN, both local and global, see e.g., [40, 11]. For example, one can view some known algorithms for network-wide broadcast as using multiple applications of this building block of Local Broadcast. That is, in many works (e.g., [3, 13]) the broadcast proceeds by having nodes that already have received the broadcast message forward it by performing Local Broadcast.

The main challenge in RN is that simultaneous transmissions in the two-hop neighborhood of v may “collide” with v 's message and prevent it from being received by some or all of v 's neighbors. In this paper, we study the scheduling of nodes' transmissions that allows all the nodes to perform their Local Broadcast task in any network, overcoming their collisions. The nodes *do not know the network topology*, but may receive a few bits of advice, as discussed later.

Parameterizing RN graph topologies to measure time performance more accurately

Recently, Davies [15] suggested that the *independence number* of a graph (i.e., the size of a maximum independent set) can accurately parameterize time complexity of *global* problems, such as broadcast and leader election. More specifically, randomized algorithms in [15] have low time complexity as a function of the independence number of the network graph.

In the current paper, we address *local* tasks. Specifically, we consider *distributed deterministic algorithms* for the natural task of *Local Broadcast* and parameterize their time performance by a local version of the domination number of the graph, called *2-local domination number* and denoted γ_2 .¹ For a given radio network with topology graph $G = (V, E)$ and a set $B \subseteq V$, the Local Broadcast problem is to deliver a message from v to all neighbors of v , for each $v \in B$. Parameter γ_2 spans from constant to $O(\Delta^2)$, and in many popular classes, e.g., Unit Disc or Bounded Growth graphs, it is constant.

Preliminary intuitions about time complexity of deterministic Local Broadcast

Before stating our contributions formally, the following intuition may be interesting. Consider a node u that has Δ neighbors $v_1, v_2, \dots, v_\Delta$. In Local Broadcast, each such neighbor v_i needs to receive a message from each of its $O(\Delta)$ neighbors (other than u), and it has to be at some rounds r when u does *not* transmit in order to avoid simultaneous conflicting transmissions. Hence, only $O(\Delta^2)$ rounds are “forbidden” for u to transmit during Local Broadcast. The latter suggests that if u knew which rounds are “forbidden” for itself, it could perform Local Broadcast in $O(\Delta^2)$ time. On the other hand, Local Broadcast can be performed much faster

¹ Local domination is defined analogously to how local independence relates to independence, see [21].

■ **Table 1** Our deterministic distributed RN Local Broadcast results in comparison with closest previous work. n is the number of network nodes, Δ is the maximum degree and γ_2 is the 2-local dominating number.

Section # or reference	Communication rounds	Protocol class	Bits of advice
[8]:Thm 13	$O(\Delta^2 \min\{\log n, \log_\Delta^2 n\})$	oblivious, $\gamma_2 \geq 1$	none
[12]:Thm 1.6	$\Omega(\min\{\Delta^2 \log_\Delta n, n\})$	oblivious, $\gamma_2 \geq 1$	none
3.2	$O(\Delta \gamma_2^2 \log \gamma_2)$	oblivious	$O(\log(\Delta \gamma_2))$
3.3	$O(\Delta \gamma_2^2 \log n)$	quasi-adaptive	$O(\log \gamma_2)$
4.2	$\Omega\left(\min\left\{\left(\frac{\min\{\Delta, \gamma_2\}}{\log n}\right)^2, n\right\}\right)$	quasi-adaptive	none
3.4	$O(\Delta \gamma_2^2 \log^2 n)$	adaptive	1
[14]:Cor 11	$\Omega(\Delta \log n)$	any, $\gamma_2 \geq 1$	none

by a randomized algorithm [3, 14], with high probability. This suggests that for every network, there exists a transmission schedule in which some of the rounds forbidden for u because of neighbor v_i are the same as those forbidden because of some other neighbor v_j (for $j \neq i$). We can further infer then that a better deterministic Local Broadcast algorithm can be obtained if each node learns the whole topology (and is not computationally bounded). However, in this paper, we show that such a saving is not possible (up to a polylogarithmic factor) for oblivious and even *quasi-adaptive* algorithms (i.e., algorithms with predefined transmission schedules, but allowing nodes to stop based on received messages and channel feedback).

What topology knowledge, or advice, is sufficient to beat the Δ^2 bound?

The general question of how much of topology knowledge may help nodes in completing a distributed task faster, was heavily investigated in various forms, for example, in [17, 16, 18, 33, 23]. Inspired by [15], who parameterized time complexity of global broadcast by the size of a Maximal Independent Set, and by our lower bound involving the minimum local size of the Dominating Set (DS), we investigate to what extent some local knowledge about some DS may speed up Local Broadcast. In particular, the extra knowledge (also called advice, or, sometimes, a label) about some nearby node in the DS and the local number among the neighbors of that node is sufficient to build an efficient oblivious transmission schedule. Note that such advice consists of a logarithmic (in Δ, γ_2) number of bits per node, and could be further decreased down to 1 when adding some adaptiveness to the protocol.

1.1 Our Contributions

Our results are detailed and compared to previous work in Table 1.

Upper bounds. In Section 3 we present three Local Broadcast algorithms with advice, which are efficient (that is, $o(\Delta^2)$) for graphs with small 2-local domination number γ_2 (defined later). More precisely, all our algorithms complete Local Broadcast in $\tilde{O}(\Delta \gamma_2^2)$ rounds (where notation $\tilde{O}$ hides polylogarithmic factors) trading the size of advice for adaptivity.

The first algorithm (Section 3.2) needs $\log(\Delta \gamma_2^2) \in O(\log(\Delta \gamma_2))$ bits of advice, but it does not need any adaptivity (i.e., the algorithm is fully oblivious). In view of the best known lower bound $\Omega(\min\{\Delta^2 \log_\Delta n, n\})$ for oblivious algorithms without advice in [12], which works for any γ_2 , it demonstrates that a logarithmic (in terms of Δ and γ_2) advice allows to beat this lower bound for networks with $\gamma_2 \in O(\sqrt{\Delta})$.

The second algorithm (Section 3.3) uses less advice of only $O(\log \gamma_2)$ bits, but needs a bit of adaptiveness in deciding to stop (i.e., the algorithm is quasi-adaptive). It is only one factor away from our new lower bound on quasi-adaptive algorithms without advice, described later.

The third algorithm (Section 3.4) requires only 1 bit of advice, but is fully adaptive. All our algorithms *do not require* collision detection or acknowledgments built-in the model. They are also close to the obvious lower bound Δ (as a node can receive at most one message in a round, because of avoiding collisions), as long as γ_2 is small.

Lower bound. In Section 4 we present a lower bound of $\Omega\left(\min\left\{\left(\min\{\Delta, \gamma_2\}/\log n\right)^2, n\right\}\right)$ for quasi-adaptive deterministic algorithms accomplishing Local Broadcast in any n -node radio networks of node degree at most Δ and 2-local domination number γ_2 without any topology knowledge or any other kind of advice. For settings where either Δ or γ_2 are not restricted, our lower bound yields either $\Omega\left(\min\left\{\left(\gamma_2/\log n\right)^2, n\right\}\right)$ or $\Omega\left(\min\left\{\left(\Delta/\log n\right)^2, n\right\}\right)$ respectively. To the best of our knowledge, this is the first (nearly) quadratic Local Broadcast (same message for all neighbors) lower bound in the RN model. To prove it, we design and analyze a **hitting game on graphs**, in which one player tries to “hit” a matching hidden in a graph designed by other player, by asking queries (each query is a subset of nodes of the graph). See Section 4.1 and Theorem 17 for more details.

Our lower bound is stronger than previous results in three ways. It is nearly quadratically (in Δ) better than the best known general lower bound for this class of algorithms, mainly, $\Omega(\Delta \log n)$ [14].² Second, comparing to the $\Omega(\min\{\Delta^2 \log_\Delta n, n\})$ lower bound in [12] for fully oblivious algorithms, ours works for much wider class of quasi-adaptive algorithms and takes into account parameter γ_2 , which as we described earlier, conveniently parametrizes time complexity of algorithms from the upper bound perspective. Third, our lower bound is also stronger than another similar lower bound of $\Omega(\Delta^2 \log n)$ in [14], which was proved for much more demanding communication task (under the name of Local Broadcast) – each node could send a *possibly different* message to each neighbor.

Other related work is overviewed in Appendix 1.1. Interesting extensions and open directions are discussed in Section 5.

A variety of *global* computational problems that are fundamental for distributed computing in communication networks has been studied under the RN model. *Local* broadcast on the other hand has been studied for randomized algorithms [3] and especially for particular cases of RN where, for instance, some (known) links are assumed to be unreliable and/or the solutions given are randomized (see [40, 26, 14] and references therein), or the topology is restricted e.g., to a complete graph. However, to the best of our knowledge, there is no previous work on *deterministic* Local Broadcast under the *general multi-hop* RN model. A non-comprehensive overview of related works, as well as lower bounds for other models that nevertheless applies to our context, are included below.

Historically, the problem of Local Broadcast was first considered in complete graphs, also called multiple-access channels or shared channels, where only a subset of nodes was active and ready to broadcast. Capetanakis [7], Hayes [31], and Tsybakov and Mikhailov [44] independently proposed a deterministic adaptive tree-like algorithm in the model with collision detection, working in $O(\Delta \log(n/\Delta))$ communication rounds, where Δ is the number of active

² The lower bound $\Omega(\Delta \log n)$ in [14] works also for adaptive, and even randomized, protocols, however it does not take into account parameter γ_2 .

nodes. The same time complexity can be achieved even by quasi-adaptive algorithms [36, 38] in a channel with acknowledgments but without collision detection. Clementi, Monti, and Silvestri [12] showed a matching $\Omega(\Delta \log(n/\Delta))$ lower bound for such a setting, which also holds for adaptive algorithms. As for the setting with collision detection, the best lower bound $\Omega(\Delta \log n / \log \Delta)$ was given by Greenberg and Winograd [30], leaving the logarithmic gap open for 40 years now.

In arbitrary RN, the best previously known deterministic Local Broadcast algorithm required time $O(\Delta^2 \min\{\log n, \log_{\Delta}^2 n\})$ by Cheraghchi and Nakos [8]. It (roughly) logarithmically improved upon the algorithm by Clementi, Monti and Silvestri [12], based on superimposed codes introduced by Kautz and Singleton [35]. Both these algorithms are oblivious, and as such do not need collision detection or acknowledgments. The complexity of randomized Local Broadcast in arbitrary networks of node degree Δ in the related Beeping Networks models matches the general lower bound and is $\Theta(\Delta \log n)$, as shown in [14].

There is a rich literature about distributed algorithms and even non-distributed online algorithms with advice and/or with informative labels, e.g., [22, 24, 34, 45, 37, 5].

2 Model, Notation, Problems, and Protocols

We use a standard description of the Radio Networks (RN) model, formulated 40 years ago [9]. Consider a communication network formed by n devices with communication and computation capabilities, called **nodes**. Each node has a unique **ID** from the range $[1, n^c]$, for some constant $c \geq 1$.³ Nodes communicate by sending **messages** among them. A message may be lost in cases specific to the nature of radio networks, defined later. A message is composed of a binary sequence containing the source node ID, the destination node ID (if applicable), and the specific information to be sent.

Network topology and communication. If a message from the source node is not lost, and it is received by the destination node, we say that the message was **delivered**. Each pair of nodes that are able to communicate directly (i.e., without relaying communication through other nodes) are said to be connected by a communication **link** and are called **neighbors**. We assume that links are **symmetric**, i.e., messages can be sent in both directions. For each node $v \in V$, the set of neighbors of v is called its **neighborhood**, denoted as $N(v)$. The network topology defined by the communication links is modeled by an undirected graph $G = (V, E)$ where V is the set of nodes/vertices and E is the set of links/edges. The network $G = (V, E)$ is **connected** if E is such that for every pair of nodes $u, v \in V$, there is a path of links connecting u and v . The maximum node degree in G is denoted by $\Delta = \max_{v \in V} |N(v)|$; we assume that nodes know a linear upper estimate of Δ .

We assume that time is slotted in **rounds** of communication such that if there is a link between nodes u and v and a message is sent by u in some round, the message is either delivered to v in the same round or it is lost (in the cases detailed later). In both cases, we say that v **transmits** in that round. If u does not transmit in some round, we say that it **listens** in that round, meaning that it waits for the reception of the transmission of a neighbor, an event that may or may not happen. All the nodes start running protocols simultaneously, i.e., the network is **synchronous**. We assume that computations take negligible time with respect to communication. Thus, we measure algorithm performance in rounds.

³ The availability of identifiers is essential in order to break symmetry in deterministic RN protocols, as pointed out in previous works, e.g., [28].

Transmission result. In a round when a node v listens, it may either receive a message - this is the case that exactly one of its neighbors transmits in that round. Or, it may also hear **silence** - this is the case that either none of its neighbors transmits or some set $\{u_1, u_2, \dots, u_k\}$ of neighbors of v , for $k > 1$ does. In the second case, we say that the messages of $\{u_1, u_2, \dots, u_k\}$ **collided** at v . In a round, a node v that transmits any message does not receive any message sent by any of its neighbors $u \in N(v)$. We say that u 's message collided with v 's message at v . If a node u transmits and its neighbor v does not receive the message (in the cases defined above), we say that u 's message is lost on the way to v . This does not mean that the other neighbors of u do not receive the message from the transmission of u .

Networks with collision detection (CD). We may or may not assume a third feedback a listening node v may receive (instead of either receiving a message or hearing silence). In a radio network equipped with CD, if more than one of v 's neighbors transmit in some round, v hears **noise** rather than silence. Obviously, an algorithm that does not use CD can function with the same complexities in networks with CD.

Advice. Some of the results presented in this paper are in the framework of algorithms with **advice** or, alternatively, with **informative labels**. In this paradigm that has recently got growing attention, an oracle knowing the network provides binary strings to nodes before the beginning of a computation. A distributed algorithm uses this advice to solve the particular problem. The required size of advice (the maximum length of the strings) can be considered as a measure of the difficulty of the problem (in terms of "how much topology knowledge is needed"). Two variations are studied in the literature: either the binary string given to nodes is the same for all of them [27] or, as we do, different strings may be given to different nodes (as in e.g. [19, 28, 6, 29]).

Preliminaries and Specification of Local Broadcast. The following notations are used for the topology graph $G = (V, E)$ of a radio network. A path between two nodes $u, v \in V$ is a sequence of adjacent links without cycles connecting u and v . If the shortest path from u to v contains x links, we say that v is within x **hops** from u and vice versa, and we also say that the **distance** between u and v is x . Let k be a natural number. The **k -neighborhood** of a node v is the set of all nodes within k hops from v , including v itself.

A **distance k coloring** of $S \subseteq V$ is a color assignment such that every node $v \in S$ receives a color such that any two nodes $u, w \in S$ that received the same color cannot be within k distance of each other. E.g., the standard coloring is distance 1 coloring of V .

A (distance-1) **dominating set** of a graph $G = (V, E)$ is a set $S \subseteq V$ such that every node of $V \setminus S$ is at distance 1 from some node in S . The (global) **domination number**, $\gamma(G)$, of G is the minimum cardinality of any dominating set of G [32]. We will use the following local version of the (distance-1) domination number. Our definition is consistent with the local version of independence characterizations [21]. Let $\mathcal{S}$ be the set of dominating sets of G . For any integer $k \geq 1$ and set $S \in \mathcal{S}$, the **k -local domination number $\gamma_k(G, v, S)$ of node $v \in V$** is the number of nodes in S within distance k of v in G . We also define a **k -local domination number $\gamma_k(G, V', S) = \max_{v \in V'} \gamma_k(G, v, S)$ of a set of nodes $V' \subseteq V$** , and the **k -local domination number $\gamma_k(G) = \min_{S \in \mathcal{S}} \gamma_k(G, S)$ of the graph G** , where $\gamma_k(G, S) = \gamma_k(G, V, S)$.

To help better understand the k -local domination number, we provide Observations 1, 2 and 3 below. Their proofs, including graph examples, will be provided in the full version of this paper.

► **Observation 1.** *For every Δ , there exists a graph G with degree Δ such that the 2-local domination number $\gamma_2(G) \in \Theta(\Delta^2)$.*

► **Observation 2.** *Consider any connected graph $G = (V, E)$ where $V \neq \emptyset$. The 2-local domination number $\gamma_2(G) = 1$ if and only if there exists a dominating set DS on graph G such that $|DS| = 1$.*

Note that a dominating set of size $|DS| = 1$ can only exist in some graphs with diameter $D \leq 2$.

The above lower bound on the 2-local dominating number $\gamma_2(G)$ may be unsatisfactory, since it only applies for small graphs where the dominating set has only one node. However, one can note that the dominating number can be as low as 2 in many large and complex graphs.

► **Observation 3.** *The 2-local domination number $\gamma_2(G)$ can be as low as 2 in some graphs with arbitrarily large n, Δ, D and $|DS|$, where D is the diameter of the graph G and DS is some dominating set.*

The formal definition of the Local Broadcast problem is as follows.

► **Definition 4 (Local Broadcast Problem).** *Given a radio network with topology graph $G = (V, E)$, where each node $v \in B$, for some set $B \subseteq V$, holds a message m_v , the problem is solved once m_v for every $v \in B$ is delivered to every node in $N(v)$.*

It is enough for us to consider only the case that $B = V$. A lower bound for this case is an existential lower bound for a general B . Similarly, it is easy to show that an upper bound for the case that $B = V$ holds for a smaller B too. Hence, in this paper, $B = V$.

Protocol Classes. We give now a precise definition of the classes of algorithms considered in this work. The following will be used. In the context of a communication protocol for radio networks with set of nodes V , let a **schedule of transmissions** be a Boolean matrix $[b_{v,r}]_{v \in V, r \in [1, \lambda]}$, where $b_{v,r}$ indicates whether node v transmits in round r or not. We say that λ is the **length** of the schedule. We consider two classes of deterministic Local Broadcast protocols: oblivious and quasi-adaptive.

An **oblivious protocol** is a schedule of transmissions $[b_{v,r}]_{v \in V, r \in [1, \lambda]}$ to be used by the nodes of the network. The schedule is pre-defined before execution, and it does not change until round λ , when all the nodes stop.

A **quasi-adaptive protocol** is a schedule of transmissions $[b_{v,r}]_{v \in V, r \in [1, \lambda]}$ to be used by the nodes of the network. The schedule is predefined before execution, but any given node may independently stop its execution in any round $1, \dots, \lambda$ after its computational task has been completed.

Quasi-adaptive protocols in multi-hop networks have been considered before, e.g. in [38] (Section 5). In such protocol nodes use channel feedback to stop, hence, they are not strictly oblivious (in the sense of lack of adaptive switch-off, as considered e.g., in [13]). However, some of their components are oblivious (based on fixed selectors). Also, many randomized algorithms are quasi-adaptive, in the sense that they transmit according to fixed distribution of schedules but stop immediately after getting acknowledgment of successful transmission(s) - the reader may find some examples of backoff-based quasi-adaptive algorithms e.g., in [38].

An **adaptive protocol** is a protocol in which each node v determines whether to transmit or listen in a given round r based on the ID of v and on the feedback from the radio channel for all rounds $r' < r$, where the feedback for a given round r' is either silence if v does not receive any message in the rounds r' or the content of the received message otherwise.

3 Algorithms with Advice to Beat the Lower Bound

In this section, we present three algorithms, trading the level of adaptiveness (from oblivious to adaptive) for bits of advice per node (from $O(\log(\Delta\gamma_2))$ to 1). All of them complete in $\tilde{O}(\Delta\gamma_2^2)$ rounds. The assumptions that follow will be used.

Suppose there is a fixed dominating set (DS), known to all the nodes. Each node not in the DS is assigned to one of its neighbors in this DS . A node in the DS is assigned to itself. Moreover, each node $v \in DS$ has a unique numbering of nodes assigned to itself, $1, \dots, i$, where i is the number of nodes assigned to v . Note that $i \leq \Delta + 1$, because the assignment is of nodes neighboring v plus the assignment of v to itself.

Initially, the above knowledge will come from an oracle (see Section 3.2). In Section 3.4, we show how to find all the above assignments but assuming a given marking of the nodes in the DS , which can be accomplished with a single bit of advice from an oracle.

Our algorithms will also use the combinatorial structures called **selectors**, **avoiding selectors** and **strong selectors**, presented below in Section 3.1. In particular, specific selectors from that section are integrated in our algorithms.

3.1 Preliminaries: Selectors

► **Definition 5** (Selectors). *A family $\mathcal{F}$ of subsets of $[n]$ is called an (n, k) -selector of length $|\mathcal{F}|$, if for every non-empty subset $S \subseteq [n]$ such that $|S| \leq k$, there exists a set $F \in \mathcal{F}$ such that $|F \cap S| = 1$.*

► **Theorem 6** ([13]). *For any $n > k \geq 2$, there exists an (n, k) -selector of length $O(k \log(n/k))$.*

► **Definition 7** (Avoiding selectors). *A family $\mathcal{F}$ of subsets of $[n]$ is called an (n, k, ℓ) -avoiding selector of length $|\mathcal{F}|$, where $1 \leq \ell < k \leq n$, if for every non-empty subset $S \subseteq [n]$ such that $|S| \leq k$ and for any subset $R \subseteq S$ of size at most ℓ , there is an element $a \in S \setminus R$ for which there exists a set $F \in \mathcal{F}$ such that $F \cap S = \{a\}$.*

The following fact follows directly from Definition 7, see also [4, 10].

► **Fact 8**. *Suppose we are given an (n, k, ℓ) -avoiding selector $\mathcal{F}$ and a set S of size at most k . Then, the number of elements in S not “selected” by selector $\mathcal{F}$ (i.e., for which there is no set in the selector that intersects S on such singleton element) is smaller than $k - \ell$.*

► **Theorem 9** ([4, 10]). *There exists an (n, k, ℓ) -avoiding selector of length $O\left(\frac{k^2}{k-\ell} \log n\right)$.*

► **Definition 10** (Strong selectors). *A family $\mathcal{F}$ of subsets of $[n]$ is called a (n, k) -strong selector of length $|\mathcal{F}|$, if for every non-empty subset $S \subseteq [n]$ such that $|S| \leq k$, for every element $a \in S$, there exists a set $F \in \mathcal{F}$ such that $F \cap S = \{a\}$.*

► **Theorem 11** ([20, 35]). *For any $n \geq 3$ and for $k \geq 2$, there exists an (n, k) -strong selector of length $O(\min\{n, k^2 \log n\})$.*

Note that selectors do not depend on the network, so they can be computed in a non-distributed way in advance and integrated into our algorithms.

3.2 An Oblivious Algorithm LB_LAd with $O(\log(\Delta\gamma_2))$ Advice

At the beginning, the oracle determines a dominating set DS which minimizes⁴ the value $\gamma_2 = \gamma_2(G, DS)$. Then, the oracle chooses a distance 4 coloring of DS in G such that the number of colors is $O(\gamma_2^2)$. Note that such a coloring always exists since, by the definition of γ_2 , the number of nodes $u \in DS$ in distance ≤ 2 from a node $v \in V$ is at most γ_2 . I.e., one can greedily color DS nodes using $\gamma_2^2 + 1$ colors. Next, the oracle assigns each $u \in V$ to its neighbor from DS and, for each $v \in DS$, provides unique labels to the nodes assigned to v from the range $[\Delta]$.

Each node v receives from the oracle three numbers: the color of the DS node, say w , to which v has been assigned by the oracle, the label of v among the nodes assigned to w , and the value of γ_2 . We assume here a DS node is always assigned to itself with the special label 0 indicating that it is a member of DS . It could be easily seen that the length of advice is $O(\log(\Delta\gamma_2))$ in each node.

Let $\mathcal{F}$ be a (γ_2^2, γ_2) -strong selector. We could assume that the number of sets in $\mathcal{F}$ is $O(\gamma_2^2 \log \gamma_2)$, see Theorem 11.

The algorithm is split into phases, each of $\Delta + 1$ rounds. A node v is *active* in phase i if the color of the DS node w to which v is assigned is in the i th set of $\mathcal{F}$; otherwise, v is *passive*. An active node v transmits its message, combined with its ID and the color of the assigned DS node w , in round x of the phase and is silent in all other rounds, where x is the label of v among nodes assigned to w (recall that this number is provided to v at the beginning of the computation as part of the advice). See Algorithm 1 for a pseudo-code.

■ **Algorithm 1** LB_LAd– Local Broadcast algorithm with $O(\log(\Delta\gamma_2))$ Advice for a node v .

```

1: Let  $\mathcal{F} = \{S_1, \dots, S_\ell\}$  be the fixed (shared between all nodes)  $(\gamma_2^2, \gamma_2)$ -strong selector,
   where  $\ell = |\mathcal{F}|$ 
2: Receive (from oracle) color  $c$  of  $DS$  node  $w$ , in some distance 4 coloring, that  $v$  will be
   assigned to
3: Receive from the oracle the number  $x$  assigned to  $v$ 
4: for  $phase = 1, \dots, \ell$  do
5:   for  $round = 1, \dots, \Delta + 1$  do
6:     if  $w \in S_{phase}$  and  $round = x$  then
7:       Transmit the message, ID of  $v$  and the color of  $w$ 
8:     else
9:       Remain silent
10:    end if
11:  end for
12: end for

```

► **Theorem 12.** *Algorithm LB_LAd is a deterministic distributed oblivious protocol performing Local Broadcast in $O(\Delta\gamma_2^2 \log \gamma_2)$ communication rounds. It uses $O(\log(\Delta\gamma_2))$ bits of advice per node, and does not require collision detection or acknowledgments built-in the model.*

Proof. Consider any two neighboring nodes v, u . We prove that v delivers its message successfully to u by the algorithm. Let w be the DS node to which v is assigned and let $DS_2(u)$ be the set of DS nodes of distance at most 2 from u . By definition, $|DS_2(u)| \leq \gamma_2$.

⁴ The algorithm works with any DS , but the round complexity depends on $\gamma_2(G, DS)$.

Note that each neighbor of u is assigned to some node in $DS_2(u)$, and is using the assigned node's color to determine, based of the corresponding set in $\mathcal{F}$, whether to be active or passive in a phase. Since $\mathcal{F}$ is a (γ_2^2, γ_2) -strong selector, there is a set in $\mathcal{F}$ containing w but no other node in $DS_2(u)$. Let it be a set i , which corresponds to being active or not in phase i . It means that only neighbors of u that are assigned to w are active in phase i . In this phase, they perform a round-robin according to their numbering (each of them knows its number, as it has been provided as part of the advice). Hence, there is no interference between these nodes at node u while recalling that all other neighbors of u are passive in phase i . It implies that node v , which is among neighbors of u assigned to w , performs transmission in a unique round of phase i and is successfully received by u .

The number of rounds is obviously upper bounded by the number of sets in the strong selector $\mathcal{F}$, multiplied by the length of each phase $\Delta + 1$. ◀

3.3 A Quasi-adaptive Algorithm with $O(\log \gamma_2)$ Advice

At the beginning, the oracle determines a dominating set DS which minimizes⁵ the value $\gamma_2 = \gamma_2(G, DS)$. Then, it chooses a distance 4 coloring of DS in G such that the number of colors is $O(\gamma_2^2)$. As explained in Section 3.2, such a coloring exists. Next, the oracle assigns each $u \in V$ to its neighbor from DS . Contrary to Section 3.2, labels are not provided.

Each node v receives two numbers from the oracle : the color of the DS node, say w , to which v has been assigned by the oracle and the value of γ_2 . We assume here a DS node is always assigned to itself with the special label 0 indicating that it is a member of DS . It could be easily seen that the length of advice is $O(\log \gamma_2)$ in each node.

Let $\mathcal{F}_i$ be a $(n, \Delta/2^{i-1}, \Delta/2^i)$ -avoiding selector, $\mathcal{F}_i = (S_1^{(i)}, \dots, S_{\ell_i}^{(i)})$. The number of sets in $\mathcal{F}_i$ is $O(\Delta/2^{i-1} \log n)$, see Theorem 9.

The algorithm is split into $\log \Delta$ phases. The i th phase consists of $|\mathcal{F}_i| = \ell_i$ stages. Finally, each stage has $\gamma_2^2 + 1$ blocks, each block has 2 rounds. A node $v \notin DS$ is *active* in stage j of a phase i if it was not switched off earlier and its ID belongs to $S_j^{(i)}$. An active node v transmits its message, combined with its ID in the Round 1 of the block k such that k is the color of the assigned DS node w . If a node $v \in DS$ receives a message in Round 1 of the block k such that the color of v is equal to k then v sends the message back in Round 2 of that block. If $v \notin DS$ sends a message in Round 1 of a block and receives its message back in Round 2 of that block, it switches off for the remaining part of the algorithm. See Algorithm 2 for a pseudo-code.

► **Theorem 13.** *Algorithm LB_quasi (Algorithm 2) is a deterministic distributed quasi-adaptive protocol performing Local Broadcast in $O(\Delta \gamma_2^2 \log n)$ communication rounds. It uses $O(\log \gamma_2)$ bits of advice per node, and does not require collision detection or acknowledgments built-in the model.*

Proof. Consider any two neighboring nodes v, u . Assume that v switches off after some block determined by the current values of the variables *phase*, *stage* and *block*. We claim that u receives the message from v in Round 1 of the given block which we will call the current block. Let w be the DS node to which v is assigned. We denote such node by $DS(v)$, i.e., $DS(v) = w$ according to the previous sentence. Thus $col(v) = col(w) = block$ and the colors of all nodes from DS in distance at most 4 from w are different from *block*. As a result, the color of each node x in distance at most 3 from w such that $DS(x) \neq w$ (i.e., x is not

⁵ The algorithm works with any DS , but the round complexity depends on $\gamma_2(G, DS)$.

■ **Algorithm 2** LB_quasi – Local Broadcast algorithm with $O(\log \gamma_2)$ Advice for a node v .

```

1: Let  $\mathcal{F}_i = \{S_1^{(i)}, \dots, S_{\ell_i}^{(i)}\}$  be a fixed (shared between all nodes)  $(n, \Delta/2^{i-1}, \Delta/2^i)$ -avoiding
   selector, where  $\ell_i = |\mathcal{F}_i|$ , for  $i = 0, 1, \dots, \log \Delta$ .
2: Receive from the oracle the color  $\text{col}(v) = c$  of  $DS$  node  $w$  such that  $v$  is assigned to  $w$ 
3: for  $phase = 1, \dots, \log \Delta$  do
4:   for  $stage = 1, \dots, \ell_{phase}$  do
5:     for  $block = 1, \dots, \gamma_2^2 + 1$  do
6:       Round 1:
7:       if  $v \in S_{stage}^{(phase)}$  and  $\text{col}(v) = block$  and  $v \notin DS$  then
8:         Transmit the message and the ID of  $v$ 
9:       else
10:        Remain silent
11:       end if
12:       Round 2:
13:       if  $\text{col}(v) = block$  and  $v \in DS$  then
14:         if  $v$  received a message in the previous round then
15:           send the received message back
16:         else
17:           Send a dummy message
18:         end if
19:       end if
20:       if  $v$  sent a message in Round 1 and received it back in Round 2 then
21:          $v$  switches off
22:       end if
23:     end for
24:   end for
25: end for

```

assigned to w) is different from $block$. Since v is a neighbor of w , the color of each node in distance at most 2 from v such that $DS(x) \neq DS(v) = w$ is different from $block$ as well. This fact implies that the only reason for which the message from v is not received by its neighbor u in Round 1 of $block$ is that another neighbor of $DS(v) = w$ transmits in that round. Then, however, there is also a collision at w in Round 1 and w will not transmit any message in Round 2 which prevents v from switching off after the current block. But this fact in turn would contradict the assumption that v switches off after the $block$.

So already know that if v eventually switches off during an execution of the algorithm, it performs its local broadcast task successfully. Thus, it remains to prove that actually each node switches off during an execution of Algorithm 2. The above considerations imply that, in order to guarantee that v switches off is some block, it is sufficient that it is the only element of the intersection of $S_{stage}^{(phase)}$ and the set of such nodes w that $DS(v)$. As the set of nodes assigned to a particular element w of DS is included in the set of neighbors of w , its initial size is at most Δ . Let X_{phase} be the set of nodes assigned to w which were not switched before the given $phase$. Note that each node x selected by $S_{stage}^{(phase)}$ as the unique element among X_{phase} (i.e., such that $X_{phase} \cap S_{stage}^{(phase)} = \{x\}$) is switched off in the given phase. With this observation one can easily prove inductively that the size of X_{phase} is at most $\Delta/2^{phase-1}$ and $\mathcal{F}_{phase}$ selects $\Delta/2^{phase}$ of them. Thus eventually, after $\log \Delta$ phases all nodes are switched off.

The number of rounds of the given *phase* $\in [1, \log \Delta]$ is $O(\ell_{\text{phase}} \cdot \gamma_2^2) = O(\Delta/2^{i-1} \cdot \gamma_2^2 \log n)$. And the time of the whole algorithm's execution is $O(\sum_{i=1}^{\log \Delta} \Delta/2^{i-1} \cdot \gamma_2^2 \log n) = O(\Delta \gamma_2^2 \log n)$. ◀

3.4 Adaptivity Reduces the Need of Advice to One Bit

At the beginning, the oracle determines a dominating set DS which minimizes⁶ the value $\gamma_2 = \gamma_2(G, DS)$. Then each node only gets one bit of advice: whether it is in the DS or not. The procedures described next allow the computing of both the assignment and the local numbering at DS nodes by deterministic distributed algorithms. Knowing the assigned DS node and a unique local number in that node (in $\{1, \dots, \Delta + 1\}$), the main algorithm uses a (n, γ_2) -strong selector $\mathcal{F}$ of length $|\mathcal{F}| \in O(\gamma_2^2 \log n)$ as follows: a node transmits in round $t = (i - 1) \cdot (\Delta + 1) + j$, where $i \in [|\mathcal{F}|]$ and $j \in \{1, \dots, \Delta + 1\}$, if its assigned DS node is in the set F_i of selector $\mathcal{F}$ and its local number is j . The pseudocode is in Algorithm 3.

■ **Algorithm 3** Local broadcast after assignment of DS nodes and numbering – Algorithm for node v .

```

1: Let  $\mathcal{F} = \{S_1, \dots, S_\ell\}$  be the fixed shared  $(n, \gamma_2)$ -strong selector, where  $\ell = |\mathcal{F}|$ 
2: for  $i = 1, \dots, \ell$  do
3:   for  $j = 1, \dots, \Delta + 1$  do
4:      $DS_v \leftarrow$  the  $DS$  node assigned to  $v$ 
5:     if  $DS_v \in F_i$  and the local number of  $v$  is equal to  $j$  then
6:        $v$  transmits
7:     end if
8:   end for
9: end for

```

We continue with describing the procedures of assigning DS node and local numbering.

DS -node Assignment Procedure

DS nodes use an (n, γ_2) -selector $\mathcal{F} = \{S_1, S_2, \dots, S_\ell\}$, where $\ell = |\mathcal{F}|$. In round i , DS nodes that are in the set S_i transmit – intuitively, they try to deliver their IDs to neighboring nodes. Each non- DS node sets the sender of the first message received as its assigned DS node. For completeness, each DS node sets itself as its own assigned DS node. This procedure is clearly oblivious. For detailed pseudocode, see Algorithm 4.

⁶ The algorithm works with any DS , but the round complexity depends on $\gamma_2(G, DS)$.

Algorithm 4 Assignment of DS nodes – Algorithm for node v .

```

1: Let  $\mathcal{F} = \{S_1, \dots, S_\ell\}$  be the fixed shared  $(n, \gamma_2)$ -selector, where  $\ell = |\mathcal{F}|$   $(n, \Delta)$ -strong
   selector, where  $k = |\mathcal{F}'|$ 
2:  $A \leftarrow \emptyset$  ▷ denotes the set of nodes assigned to  $v$ 
3: if  $v \in DS$  then
4:    $assignment \leftarrow v$ 
5: end if
6: for  $round = 1, \dots, \ell$  do ▷ first part of the algorithm
7:   if  $v \in S_{round}$  and  $v \in DS$  then
8:     Transmit ID of  $v$ 
9:   end if
10:  if received a transmission with ID of a node  $u$  and  $assignment = null$  then
11:     $assignment \leftarrow u$ 
12:  end if
13: end for

```

► **Lemma 14.** *The DS -node Assignment procedure terminates in $O(\gamma_2 \log(n/\gamma_2) + \Delta \log n)$ communication rounds. At the end, each node v knows the ID of the DS node it is assigned to.*

Proof. From Theorem 6, the length of (n, γ_2) -selector, and thus also the number of communication rounds in the whole procedure, is $O(\gamma_2 \log(n/\gamma_2))$.

From Definition 5, for any non- DS node v there will be a round in the procedure such that v hears exactly one transmission. The transmissions only come from the DS nodes. Thus, any non- DS node v will be assigned to some DS node. Recall that any DS node is assigned to itself, by default. ◀

Numbering Procedure

This procedure proceeds in $1 + \log \Delta$ stages. In each stage, every DS node assigns numbers to half of its remaining unnumbered assigned nodes, about which it learns in this stage. In the i -th stage, all the nodes use a $(n, \Delta/2^{i-1}, \Delta/2^i)$ -avoiding selector $\mathcal{F}_i$, for $i = 1, \dots, 1 + \log \Delta$. Intuitively, the second parameter of the selector (equal to $\Delta/2^{i-1}$) is the upper bound on the number of neighbors assigned to a DS node by the preceding DS -node Assignment Procedure, which are possible unknown to the DS node, and thus locally unnumbered, at the start of the stage. The third parameter of the selector (equal to $\Delta/2^i$) is the upper bound on the number of assigned neighbors that may remain unnumbered after the stage.

For each set S in $\mathcal{F}_i$, there are two phases: in the first phase, the unnumbered nodes selected by S (i.e., contained in S) transmit their IDs with ID of their DS -assigned node (known prior the procedure, because of the preceding DS -node Assignment Procedure, see also Lemma 14). If a DS node v receives an ID of a node u with the information that u is assigned to v , then, in the second phase, v transmits a response (also called an acknowledgment), composed of a unique number assignment and IDs of v and u . This (locally) unique number starts with 2 (number 1 is reserved for the DS node itself), and is incremented by 1 after each such activity of the DS node v .

The first phase only uses a single communication round, though not all messages may be successfully received. The second phase is accomplished by using the (n, γ_2) -strong selector $\mathcal{G} = \{T_1, T_2, \dots, T_\ell\}$, where $\ell = |\mathcal{G}|$. Hence, the second phase takes ℓ communication rounds – each DS node v that is supposed to send an acknowledgment (with content described earlier) does so in rounds i such that $v \in T_i$. For detail pseudocode, see Algorithm 5.

■ **Algorithm 5** Numbering nodes – Algorithm for node v .

```

1: Let  $\mathcal{F}_i$ , for  $i = 1, \dots, 1 + \log \Delta$ , be the fixed shared  $(n, \Delta/2^{i-1}, \Delta/2^i)$ -avoiding selector
2: Let  $\mathcal{G} = \{T_1, \dots, T_\ell\}$  be the fixed shared  $(n, \alpha_2)$ -strong selector, where  $\ell = |\mathcal{G}|$ 
3: Let assignment denote the DS node that  $v$  is assigned to
4: Let  $A$  denote the set of nodes assigned to  $v$  ▷ only used if  $v \in DS$ 
5: number  $\leftarrow null$  ▷ denotes the number assigned to  $v$ 
6: if  $v \in DS$  then
7:   number  $\leftarrow 1$ 
8: end if
9: current  $\leftarrow 2$  ▷ denotes the next number that will be given to a neighbor
10: for stage =  $1, \dots, 1 + \log \Delta$  do
11:   for  $S \in \mathcal{F}_{stage}$  do
12:     received  $\leftarrow null$  ▷ ID of a node that  $v$  will respond to in the second phase
13:     round  $\leftarrow 1$  ▷ denotes the round number of processing  $S$ 
14:     if number = null and  $v \notin DS$  and  $v \in S$  then
15:       Transmit ID of  $v$ 
16:     end if
17:     if  $v \in DS$  and received ID of a node  $u$  and  $u \in A$  then
18:       received  $\leftarrow u$ 
19:     end if
20:     for round =  $2, \dots, \ell + 1$  do ▷ second phase
21:       if  $v \in DS$  and received  $\neq null$  and  $v \in T_{round-1}$  then
22:         Transmit (current, ID of  $v$ , ID of received)
23:         current  $\leftarrow current + 1$ 
24:       end if
25:       if received ( $a, b, c$ ) and  $b = assignment$  and  $c = v$  then
26:         number  $\leftarrow a$ 
27:       end if
28:     end for
29:   end for
30: end for

```

► **Lemma 15.** *The Numbering procedure finishes in $O(\Delta \gamma_2^2 \log n)$ communication rounds. Consider any node v and a node u such that v is assigned to u . The Numbering procedure guarantees that v knows a unique number $1, \dots, \Delta + 1$ among all the nodes assigned to u .*

Proof. For each set S in $\mathcal{F}_i$ for any i , phase one takes 1 round and phase two takes $|\mathcal{G}|$ rounds. According to Theorem 11, there are $1 + |\mathcal{G}| = O(\gamma_2^2 \log n)$ communication rounds for each S and i . There are $|\mathcal{F}_i|$ such sets S in a stage i . According to Theorem 9, there exists a constant c such that $|\mathcal{F}_i| \leq c\Delta/2^i \log n$ for each i . Therefore, $\sum_{i=1}^{\log \Delta} |\mathcal{F}_i| \leq 2c\Delta \log n = O(\Delta \log n)$. Hence, there are $O(\Delta \gamma_2^2 \log^2 n)$ communication rounds in total.

When we look at all the phases of a single stage i , then, from Fact 8, there are at most $\Delta/2^{i-1}$ non-*DS* assigned neighbors that failed to transmit their IDs to their assigned *DS* node.

In phase two, every acknowledgment is received: From Definition 10, there is a round j such that the acknowledgment from node v to node u is the only message that node u hears in round j . Thus, every acknowledgment is guaranteed to be received by its recipient at least once.

Therefore, at the end of stage i , there are at most $\Delta/2^{i-1}$ unnumbered assigned neighbors for each *DS* node. In particular, at the end of stage $1 + \log \Delta$, there are less than 1, i.e., exactly 0 unnumbered assigned neighbors for each *DS* node. Finally, the local numbering

at each DS node is done using subsequent integers from 1 up to possibly $\Delta + 1$, and each node is numbered once (when it transmits successfully to the assigned DS node for the first time). $\blacktriangleleft$

Below we conclude the analysis of the whole algorithm: the main algorithm, preceded by the two procedures.

► **Theorem 16.** *If nodes know n, Δ, γ_2 , or their linear upper bounds, then a single bit of advice allows a deterministic distributed Local Broadcast protocol working in $O(\Delta \gamma_2^2 \log^2 n)$ communication rounds. The protocol may need a constant adaptivity (i.e., rounds in which it uses some information to redefine its next schedule), but does not require collision detection or acknowledgments built-in the model.*

Proof. By Lemma 14, after the DS -node Assignment Procedure, each node is assigned to one DS node (in the neighborhood, including itself). By Lemma 15, after the next Numbering Procedure, each node in DS learns about all neighbors assigned to it and give them unique numbers from 2 to $\Delta + 1$, reserving number 1 for itself.

Now we analyze the Local Broadcast property of the main algorithm. Suppose, to the contrary, that some node v has not successfully broadcasted to some of its neighbors, u . Let v^*, u^* be the DS nodes assigned to v, u , respectively, by the DS -node Assignment Procedure. Note that both v^*, u^* are in 2-hop neighborhood of u . Therefore, there is a set F_i in the strong selector $\mathcal{F}$ used by the main algorithm such that $v^* \in F_i$ and node of other DS nodes in 2-hop neighborhood of u belongs to F_i . It follows from the property of strong selector $\mathcal{F}$ and the upper bound γ_2 on the number of DS nodes in any 2-hop neighborhood of any node (in particular, node u). It follows that no neighbors of u transmits in rounds $(i-1) \cdot (\Delta+1) + x$, for $x \in \{1, \dots, \Delta+1\}$, except of those assigned to the DS node v^* (such as node v). Let x^* be the locally unique number of node v . Then, in round $(i-1) \cdot (\Delta+1) + x^*$, also no other node assigned to v^* transmits, by the property of unique numbering of nodes assigned to v^* . Hence, v is the only transmitting node in the neighborhood of u , including u , in that round, and so u receives its message. This contradiction proves that the main algorithm, preceded by the two procedures, guarantees Local Broadcast.

By Lemmas 14 and 15, and by the number of rounds $|\mathcal{F}| \cdot (\Delta+1) \in O(\Delta \gamma_2^2 \log n)$ of the main algorithm, we derive the upper bound $O(\Delta \gamma_2^2 \log^2 n)$ on the total number of rounds. $\blacktriangleleft$

4 Lower Bound via Hitting Game

In this section, we prove a lower bound for Local Broadcast in the classic RN model with collision detection as defined in Section 2. First, we present a technical tool, of potentially independent interest, to be used later to prove our main lower bound.

4.1 Directed-matching Hitting Game

We prove in this section a lower bound on the number of rounds to finish a directed-matching hitting game between a deterministic algorithm and a randomized adversary on an undirected graph $G = (V, E)$, where V is the set of nodes and E the set of edges. The parameters of the game are the number of nodes n , the maximum degree Δ , and the 2-local domination number $\gamma_2 = \gamma_2(G)$ (defined in Section 2). The following additional definitions will be used.

A **directed matching** on a graph $G = (V, E)$ is a set $M = \{(u_1, w_1), (u_2, w_2), (u_3, w_3), \dots\}$ of ordered pairs of nodes such that the edges $M' = \{\{u_1, w_1\}, \{u_2, w_2\}, \{u_3, w_3\}, \dots\}$ form a matching in G . We call a subset of nodes $Q \subseteq V$ a **query** on V . Given a query $Q \subseteq V$

and a pair of nodes $u, w \in V$, we say that Q **hits the ordered pair** (u, w) if and only if $\{u, w\} \in E$, $u \in Q$, and for all $v \in Q$ such that $v \neq u$, the following holds: $\{v, w\} \notin E$. Given a set of queries $\mathcal{Q} = \{Q_1, Q_2, \dots\}$ and a directed matching M , we say that $\mathcal{Q}$ **hits the directed matching** M if and only if, for every ordered pair $(u, w) \in M$, there is a query $Q \in \mathcal{Q}$ that hits the ordered pair (u, w) .

In the directed-matching hitting game, the adversary and the algorithm compete, with the algorithm trying to hit all the edges of a directed matching chosen by the adversary and the adversary trying to prevent edges of the directed matching from being hit by adding edges under some restrictions. The definition of the game is given in Figure 1.

Directed-matching Hitting Game with parameters γ_2 , Δ and n :

- **Game initialization.** Before the game starts, the adversary chooses a directed matching M of size $n/2$ (without loss of generality, we assume n to be even). We label the nodes of this directed matching by $M = \{(u_1, w_1), (u_2, w_2), \dots, (u_{n/2}, w_{n/2})\}$ for reference. We denote $U = \{u_i | i \in [n/2]\}$ and $W = \{w_i | i \in [n/2]\}$. The corresponding set of undirected edges $E_0 = \{\{u_i, w_i\} | i \in [n/2]\}$ is the initial set of edges in the graph. On the other hand, the algorithm chooses a sequence of queries $\langle Q_1, Q_2, \dots \rangle$, hidden from the adversary, where $Q_r \subseteq U$ for any $r = 1, 2, \dots$.
- **Rules of one round of the game.** At the beginning of each round r , the algorithm announces the query $Q_r \setminus H_r$, where H_r contains every node $u_i \in U$ such that the ordered pair (u_i, w_i) has been hit before round r . (Informally, the algorithm is allowed to remove the members of H_r from the query Q_r it prepared in advance.) Then, the adversary may add some edges to the graph, restricted to the requirements:
 - the maximum degree of the resulting graph is at most Δ ,
 - the 2-local domination number of the resulting graph is at most γ_2 ,
 - if an edge $\{u_j, w_i\}$, $i, j \in [n/2]$, is added in round r , then (u_i, w_i) has not been hit before r .
- **Ending conditions.** The game ends in a round τ once (every edge of) the directed matching M has been hit by the queries $\langle Q_1, Q_2, \dots, Q_\tau \rangle$ announced by the algorithm.
- **Objective of each player.** The objective of the algorithm is to minimize the number of rounds τ , whereas the objective of the adversary is to maximize it.

■ **Figure 1** Game definition.

In the above context, we prove the following.

► **Theorem 17.** *Consider a directed-matching hitting game with parameters $\gamma_2 > 1$, $\Delta > 1$ and $n \geq 3$. For each algorithm player $\mathcal{P}_A$, there exists an adversary player $\mathcal{P}_B$ such that the number of rounds τ needed to finish the game is $\tau \in \Omega \left(\min \left\{ \left(\frac{\min\{\Delta, \gamma_2\}}{\log n} \right)^2, n \right\} \right)$.*

The proof of Theorem 17 (deferred to the full version of this paper) is based on showing a strategy for the adversary for each query of the sequence announced by the algorithm. The strategy is to initially partition the directed matching in *gadgets* (subsets of ordered pairs) at random. Then, add edges to each gadget to make it bipartite-complete. These added edges will guarantee that hits in large queries are very unlikely. Some additional edges are added to each gadget so that a chosen node dominates the gadget. Starting from this initial graph, for each small query the adversary adds edges to prevent hits from nodes that do not appear very frequently in small queries. Other hits are allowed because they are difficult or impossible to prevent. In our analysis we prove that such a strategy fulfills the requirements

of the game, and that within the claimed time lower bound only a fraction of the directed matching can be hit, all with positive probability. Hence, by the probabilistic method, there exists a deterministic strategy that proves the claim.

4.2 Local Broadcast in the RN Model

In this section, we prove a lower bound for Local Broadcast using the lower bound for the directed-matching hitting game proved in Section 4.1. Our lower bound applies to quasi-adaptive protocols, and hence to oblivious protocols as well.

► **Theorem 18.** *Consider any deterministic quasi-adaptive protocol $\mathcal{P}$ that solves Local Broadcast in the RN model. Let τ be the number of rounds needed by $\mathcal{P}$ in the worst case. Then, for each $\mathcal{P}$, there exists an adversarial input network with set V of n nodes, 2-local domination number at most γ_2 , and maximum degree Δ such that $\tau \in \Omega\left(\min\left\{\left(\frac{\min\{\Delta, \gamma_2\}}{\log n}\right)^2, n\right\}\right)$.*

Proof. For the sake of contradiction, assume there exists a deterministic quasi-adaptive Local Broadcast protocol $\mathcal{P} = [b_{v,r}]_{v \in V, r \in \tau}$ such that $\tau \in o(\min\{n, (\min\{\Delta, \gamma_2\}/\log n)^2\})$ for all input radio networks with a set V of n nodes, 2-local domination number at most γ_2 , and maximum degree Δ . We show in the rest of the proof that then we can use such Local Broadcast protocol (with some modification) as the algorithm player in the game of Section 4.1, and use the network built by the adversary player during the game as the input of the Local Broadcast protocol to reach a contradiction with the lower bound in Theorem 17. The details follow.

Consider a protocol $\mathcal{P}'$ derived from $\mathcal{P}$ by removing half of the schedule. Specifically, for any even partition $V = \{U, W\}$, it is, $\mathcal{P}' = [b_{u,r}]_{u \in U, r \in \tau}$. (Without loss of generality, we assume that n is even.) Let the **restricted Local Broadcast** problem for a subset of nodes $U \subseteq V$ be the same as the general Local Broadcast, except that it is restricted to messages of nodes in U . That is, each node $u \in U$ holds a message m_u , and the problem is solved once m_u is delivered to every node in $N(u)$ (recall that $N(u)$ is the set of neighbors of u in V). Given the assumption above, protocol $\mathcal{P}'$ solves the *restricted* Local Broadcast problem for U in time $\tau \in o(\min\{n, (\min\{\Delta, \gamma_2\}/\log n)^2\})$. The proof of the latter is a relatively straightforward argument, which we nevertheless include for completeness. Assume, for the sake of contradiction, the opposite. That is, for some input network executing protocol $\mathcal{P}$, the message of some node $u \in U$ is delivered to all nodes in $N(u)$, but if protocol $\mathcal{P}'$ is executed in the same input network, there is a pair of nodes $u \in U$ and $w \in W$, such that $w \in N(u)$ but the message m_u was not received by w . Let r be the round when m_u is delivered to w while running protocol $\mathcal{P}$. If the message was delivered to w it means that no other node in $N(w)$ was transmitting in round r , i.e., $b_{x,r} = \text{false}$, for all $x \in N(w)$ such that $x \neq u$. Then, it is also $b_{x,r} = \text{false}$, for all $x \in N(w) \setminus W$ such that $x \neq u$. Hence, m_u is also delivered to w in round r (or before) if nodes execute $\mathcal{P}'$, which is a contradiction.

Then, we use $\mathcal{P}'$ as the algorithm player $\mathcal{P}_A$ of the directed-matching hitting game presented in Section 4.1. That is, we use the schedule of transmissions of $\mathcal{P}'$ as the sequence of queries of $\mathcal{P}_A$. Then, following the strategy specified in the proof of Theorem 17, by round τ , the adversary has built a graph $G_\tau = (V, E_\tau)$. According to Theorem 17, with positive probability, G_τ is one of the adversarial graphs that force the game to take $\Omega(\min\{n, (\min\{\Delta, \gamma_2\}/\log n)^2\})$ rounds before hitting the directed matching M embedded in G_τ (in fact, with high probability, as the analysis shows). Notice that due to the restrictions of the adversary in the game (see Figure 1), G_τ has the maximum degree Δ , local 2-domination number at most γ_2 , and edges are added only to prevent the hit of matching

ordered-pairs that have not been hit earlier. Thus, edges added during the game would not have prevented hits that happened earlier, had those edges been in the graph from the beginning. Therefore, if G_τ had been used from round 1 of the game, the lower bound on the time to hit M claimed in Theorem 17 still holds.

Consider a radio network with topology graph G_τ . Given our initial assumption that $\mathcal{P}$ solves Local Broadcast on any network in time $o(\min\{n, (\min\{\Delta, \gamma_2\}/\log n)^2\})$, as we showed above, $\mathcal{P}'$ must solve *restricted* Local Broadcast on G_τ also in the same time. However, to solve restricted Local Broadcast on G_τ , the directed matching M must be hit. Thus, the running time of $o(\min\{n, (\min\{\Delta, \gamma_2\}/\log n)^2\})$ is a contradiction with the lower bound in Theorem 17. $\blacktriangleleft$

5 Conclusions and Open Directions

Most research on RN addressed global problems, often using probabilistic algorithms especially when the network topology is not known. We foray into local problems, in particular Local Broadcast, and the use of deterministic algorithms for them. We demonstrated that in case of oblivious and quasi-adaptive solutions, not much can be improved beyond quite pessimistic lower bounds. We showed on the other hand that a small amount of knowledge and/or a small amount of adaptivity suffice for time complexity improvement.

This work leaves many questions open. What is the inherent complexity of other local tasks? Which global tasks can be performed without using Local Broadcast as a primitive, and thus, possibly have even a lower complexity? Even for Local Broadcast there remain many questions. Our bounds are for the case that all the nodes wish to perform Local Broadcast. Can one derive better algorithms for the case that only some x nodes wish to locally broadcast? Must x be known to help? We provided three points in the tradeoff between adaptivity and knowledge. What is the complete tradeoff?

References

- 1 Norman Abramson. The aloha system: Another alternative for computer communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, pages 281–285, 1970. doi:10.1145/1478462.1478502.
- 2 Noga Alon, Amotz Bar-Noy, Nathan Linial, and David Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43(2):290–298, 1991. doi:10.1016/0022-0000(91)90015-W.
- 3 Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *Journal of Computer and System Sciences*, 45(1):104–126, 1992. doi:10.1016/0022-0000(92)90042-H.
- 4 Annalisa De Bonis, Leszek Gasieniec, and Ugo Vaccaro. Optimal two-stage algorithms for group testing problems. *SIAM J. Comput.*, 34(5):1253–1270, 2005. doi:10.1137/S0097539703428002.
- 5 Joan Boyar, Lene M Favrholdt, Christian Kudahl, Kim S Larsen, and Jesper W Mikkelsen. Online algorithms with advice: A survey. *ACM Computing Surveys (CSUR)*, 50(2):1–34, 2017. doi:10.1145/3056461.
- 6 Gewu Bu, Maria Potop-Butucaru, and Mikaël Rabie. Wireless broadcast with short labels. In *International Conference on Networked Systems*, pages 146–169. Springer, 2020. doi:10.1007/978-3-030-67087-0_10.
- 7 J. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Trans. Inf. Theor.*, 25(5):505–515, September 1979. doi:10.1109/TIT.1979.1056093.

- 8 Mahdi Cheraghchi and Vasileios Nakos. Combinatorial group testing and sparse recovery schemes with near-optimal decoding time. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS'20 2020, Durham, NC, USA, November 16-19, 2020*, pages 1203–1213. IEEE, 2020. doi:10.1109/FOCS46700.2020.00115.
- 9 Imrich Chlamtac and Shay Kutten. On broadcasting in radio networks-problem analysis and protocol design. *IEEE Transactions on Communications*, 33(12):1240–1246, 1985. doi:10.1109/TCOM.1985.1096245.
- 10 Bogdan S. Chlebus and Dariusz R. Kowalski. Almost optimal explicit selectors. In Maciej Liskiewicz and Rüdiger Reischuk, editors, *Fundamentals of Computation Theory, 15th International Symposium, FCT 2005, Lübeck, Germany, August 17-20, 2005, Proceedings*, volume 3623 of *Lecture Notes in Computer Science*, pages 270–280. Springer, 2005. doi:10.1007/11537311_24.
- 11 Israel Cidon and Osnat Mokryn. Propagation and leader election in a multihop broadcast environment. In *Distributed Computing: 12th International Symposium, DISC'98 Andros, Greece, September 24–26, 1998 Proceedings 12*, pages 104–118. Springer, 1998. doi:10.1007/BFB0056477.
- 12 Andrea EF Clementi, Angelo Monti, and Riccardo Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In *Proceedings of the twelfth annual ACM-SIAM Symposium on Discrete algorithms (SODA'01)*, pages 709–718, 2001.
- 13 Andrea E.F. Clementi, Angelo Monti, and Riccardo Silvestri. Distributed broadcast in radio networks of unknown topology. *Theoretical Computer Science*, 302(1):337–364, 2003. doi:10.1016/S0304-3975(02)00851-4.
- 14 Peter Davies. Optimal message-passing with noisy beeps. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing (PODC'23)*, pages 300–309, 2023. doi:10.1145/3583668.3594594.
- 15 Peter Davies. Uniting general-graph and geometric-based radio networks via independence number parametrization. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing (PODC'23)*, pages 290–299, 2023. doi:10.1145/3583668.3594595.
- 16 Anders Dessmark and Andrzej Pelc. Broadcasting in geometric radio networks. *Journal of Discrete Algorithms*, 5(1):187–201, 2007. doi:10.1016/J.JDA.2006.07.001.
- 17 Krzysztof Diks, Evangelos Kranakis, Danny Krizanc, and Andrzej Pelc. The impact of information on broadcasting time in linear radio networks. *Theoretical Computer Science*, 287(2):449–471, 2002. doi:10.1016/S0304-3975(01)00256-0.
- 18 Faith Ellen and Seth Gilbert. Constant-length labelling schemes for faster deterministic radio broadcast. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'20)*, pages 213–222, 2020. doi:10.1145/3350755.3400238.
- 19 Faith Ellen, Barun Gorain, Avery Miller, and Andrzej Pelc. Constant-length labeling schemes for deterministic radio broadcast. *ACM Transactions on Parallel Computing*, 8(3):1–17, 2021. doi:10.1145/3470633.
- 20 Paul Erdős, Peter Frankl, and Zoltán Füredi. Families of finite sets in which no set is covered by the union of r others. *Israel J. Math.*, 51(1-2):79–89, 1985.
- 21 Ralph J. Faudree, Zdeněk Ryjáček, and Richard H. Schelp. On local and global independence numbers of a graph. *Discrete Applied Mathematics*, 132(1):79–84, 2003. Stability in Graphs and Related Topics. doi:10.1016/S0166-218X(03)00391-3.
- 22 Pierre Fraigniaud, David Ilcinkas, and Andrzej Pelc. Communication algorithms with advice. *J. Comput. Syst. Sci.*, 76(3-4):222–232, 2010. doi:10.1016/J.JCSS.2009.07.002.
- 23 Adam Gańczorz, Tomasz Jurdziński, and Andrzej Pelc. Optimal-length labeling schemes for fast deterministic communication in radio networks. *arXiv preprint arXiv:2410.07382*, 2024.
- 24 Cyril Gavoille, David Peleg, Stéphane Pérennes, and Ran Raz. Distance labeling in graphs. *Journal of algorithms*, 53(1):85–112, 2004. doi:10.1016/J.JALGOR.2004.05.002.

- 25 Mohsen Ghaffari and Bernhard Haeupler. Near optimal leader election in multi-hop radio networks. In *Proceedings of the twenty-fourth annual ACM-SIAM Symposium on Discrete algorithms (SODA'13)*, pages 748–766. SIAM, 2013. doi:10.1137/1.9781611973105.54.
- 26 Seth Gilbert, Nancy A. Lynch, Calvin Newport, and Dominik Pajak. On simple back-off in unreliable radio networks. *Theor. Comput. Sci.*, 806:489–508, 2020. doi:10.1016/J.TCS.2019.08.027.
- 27 Christian Glacet, Avery Miller, and Andrzej Pelc. Time vs. information tradeoffs for leader election in anonymous trees. *ACM Trans. Algorithms*, 13(3):31:1–31:41, 2017. doi:10.1145/3039870.
- 28 Barun Gorain and Andrzej Pelc. Finding the size of a radio network with short labels. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, pages 1–10, 2018. doi:10.1145/3154273.3154298.
- 29 Barun Gorain and Andrzej Pelc. Finding the size and the diameter of a radio network using short labels. *Theoretical Computer Science*, 864:20–33, 2021. doi:10.1016/J.TCS.2021.02.004.
- 30 Albert G. Greenberg and Shmuel Winograd. A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels. *J. ACM*, 32(3):589–596, 1985. doi:10.1145/3828.214125.
- 31 J.F. Hayes. An adaptive technique for local distribution. *IEEE Communications Magazine*, 19(2), 1981.
- 32 Michael A. Henning. *Distance Domination in Graphs*, pages 205–250. Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-51117-3_7.
- 33 David Ilcinkas, Dariusz R Kowalski, and Andrzej Pelc. Fast radio broadcasting with advice. *Theoretical Computer Science*, 411(14-15):1544–1557, 2010. doi:10.1016/J.TCS.2010.01.004.
- 34 Michal Katz, Nir A Katz, Amos Korman, and David Peleg. Labeling schemes for flow and connectivity. *SIAM Journal on Computing*, 34(1):23–40, 2004. doi:10.1137/S0097539703433912.
- 35 William Kautz and Roy Singleton. Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory*, 10(4):363–377, 1964. doi:10.1109/TIT.1964.1053689.
- 36 Janos Komlós and Albert Greenberg. An asymptotically nonadaptive algorithm for conflict resolution in multiple-access channels. *IEEE Transactions on Information Theory*, 31:302–306, April 1985. doi:10.1109/TIT.1985.1057020.
- 37 Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. In *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 9–18, 2005. doi:10.1145/1073814.1073817.
- 38 Dariusz R. Kowalski. On selection problem in radio networks. In *Proceedings of the Twenty-fourth Annual ACM Symposium on Principles of Distributed Computing*, PODC '05, pages 158–166, New York, NY, USA, 2005. ACM. doi:10.1145/1073814.1073843.
- 39 Eyal Kushilevitz and Yishay Mansour. An $\omega(d \log(n/d))$ lower bound for broadcast in radio networks. *SIAM journal on Computing*, 27(3):702–712, 1998.
- 40 Nancy Lynch and Calvin Newport. A (truly) local broadcast layer for unreliable radio networks. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC'15)*, pages 109–118, 2015. doi:10.1145/2767386.2767411.
- 41 Robert Metcalfe and David R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Commun. ACM*, 19(7):395–404, 1976. doi:10.1145/360248.360253.
- 42 David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, Philadelphia, PA, USA, 2000.
- 43 David Peleg. Time-efficient broadcasting in radio networks: A review. In Tomasz Janowski and Hrushikesh Mohanty, editors, *Distributed Computing and Internet Technology, 4th International Conference, ICDIT 2007, Bangalore, India, December 17-20, Proceedings*, volume 4882 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2007. doi:10.1007/978-3-540-77115-9_1.
- 44 B S. Tsybakov and V A. Mikhailov. Free synchronous packet access in broadcast channel with feedback. *Probl Inf Transm*, 14(4), October 1978.
- 45 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM (JACM)*, 52(1):1–24, 2005. doi:10.1145/1044731.1044732.