Bounds on Contention Management in Radio Networks

by

Mohsen Ghaffari

B.S. Electrical Engineering, Sharif University of Technology, 2010
B.S. Computer Science, Sharif University of Technology, 2010

Submitted to the Department of Electrical Eng. and Computer Sci.
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2013

Author . . . . . .
Department of Electrical Eng. and Computer Sci.
February 1, 2013

Certified by . . . . . .
Nancy Lynch
Professor
Thesis Supervisor

Accepted by . . . . . .
Leslie Kolodziejski
Chairman, Department Committee on Graduate Theses

Bounds on Contention Management in Radio Networks

by

Mohsen Ghaffari

Submitted to the Department of Electrical Eng. and Computer Sci.
on February 1, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

In this thesis, we study the local broadcast problem in two well-studied wireless network models. The local broadcast problem is a theoretical approach for capturing the contention management issue in wireless networks; it assumes that processes are provided messages, one by one, that must be delivered to their neighbors. We study this problem in two theoretical models of wireless networks, the *classical radio network model* and its more recent generalization, *the dual graph model* which includes the possibility of unreliable time-changing links. Both these models are synchronous; the execution proceeds in lock-step rounds and in each round, each node either transmits a message or listens. In each round of the dual graph model, each unreliable link might be active or inactive, whereas in the classical model, all the links are always active. In each round, each node receives a message if and only if it is listening and exactly one of its neighbors, with respect to the the active links of that round, transmits.

The time complexity of the local broadcast algorithms is measured by two bounds, the acknowledgment bound and the progress bound. Roughly speaking, the former bounds the time it takes each broadcasting node to deliver its message to all its neighbors and the latter bounds the time it takes a node to receive at least one message, assuming it has a broadcasting neighbor. Typically these bounds depend on the maximum contention and the network size.

The standard local broadcast strategy is the *Decay* protocol introduced by Bar-Yehuda et al. [19] in 1987. During the 25-years period in which this strategy has been used, it has remained an open question whether it is optimal. In this paper, we resolve this long-standing question. We present lower bounds on progress and acknowledgment bounds in both the classical and the dual graph model and we show that, with a slight optimization, the Decay protocol matches these lower bounds in both models. However, the tight progress bound of the dual graph model is exponentially larger than the progress bound in the classical model, in its dependence on the maximum contention. This establishes a separation between the two models, proving that progress in the dual graph model is strictly and exponentially harder than its classical predecessor. Combined, our results provide an essentially complete characterization of the local broadcast problem in these two important models.

Thesis Supervisor: Nancy Lynch
Title: Professor

Acknowledgments

I would like to take this opportunity to thank Nancy Lynch for all the support and help she gave me and for her valuable advice. If it were not for Nancy's careful and thorough reviews, this thesis would be far less formal and perhaps full of errors.

I would also like to thank my collaborators Bernhard Haeupler and Calvin Newport who worked with me on the results in this thesis and elsewhere. Bernhard, I have been (and hopefully will continue) enjoying every minute of our discussions and the hours we have spent staring at the boards, trying to come up with solutions for various problems.

Contents

Chapter 1

Introduction

Wireless networks have become an important part of communications networks. This trend is getting more and more pronounced with mobile computation devices like laptops, notebooks, and smart-phones, which use wireless communications and make wireless networks essentially ubiquitous. Wireless networks are distinguished from the wired networks by two main characteristics: their *broadcast-type communication* and their *interference-prone nature*. More precisely, on one hand, when a node transmits a message, this message can potentially reach all of its neighbors; on the other, when two or more neighbors of a node transmit messages simultaneously, these transmissions interfere and this node does not receive either of the messages. In this case, we say *the transmitted messages are lost due to collision*. These two characteristics give rise to a form of contention between nearby nodes on accessing the shared medium.

This contention makes the task of designing higher-level applications and algorithms challenging. It is convenient and preferable to separate the challenge of wireless network contention management from the challenges of solving the higher-level problems that rely on it. The practical community of wireless networks addresses this issue by numerous Medium Access Control (MAC) layer designs [1, 2, 3, 5, 6, 4]. The theory community abstracts this issue as the *local broadcast problem* [31, 35, 36, 38, 40].

In this thesis, we study the local broadcast problem and characterize its complexity with respect to certain measures. We explain the local broadcast problem and the related measures in Section 1.2. Before that, we first present an informal description of the models.

1.1 The Models

We consider two synchronous multi-hop radio network models: the *classical radio network* model and the *dual graph* model.

The classical radio network model was introduced by Bar-Yehuda et al. [19] and is arguably

the most widely-used model in the study of wireless network algorithms in distributed computing community. It describes the communication topology of a multi-hop radio network by a graph and allows each node to broadcast a message to all its neighbors in each round with the restriction that concurrent broadcasts by two or more neighbors of a node u lead to message loss at u, due to collisions.

The *dual graph* model was introduced more recently by Kuhn et al. [31, 33] and generalizes the classical model by allowing some edges in the communication graph to be unreliable, and therefore to drop messages in an adversarial manner. The addition of these unreliable edges is intended to match the reality of radio communication, where links can behave unpredictably due to various reasons such as dynamic fading and ambient interference.

1.2 The Local Broadcast Problem

The informal description of the local broadcast problem is as follows: we have a set of *processes*, which abstract the local broadcast modules of the wireless nodes. On the other hand, we have an *environment*, which abstracts the higher layers of these wireless nodes, i.e., the modules that are trying to solve higher layer problems. The environment sends some messages to the processes, one at a time for each process, and the processes must deliver these messages to their neighbors.

Similar to [31, 38], we characterize the efficiency of a local broadcast algorithm by two metrics: (1) an *acknowledgment bound*, which measures the time for a process that has a message for broadcast to deliver its message to all of its neighbors, and (2) a *progress bound*, which measures the time for a process to receive at least one message, assuming that it has at least one neighbor with a message for transmission.

The acknowledgment bound is obviously interesting. The progress bound has also been shown to be very important for tightly analyzing algorithms for several problems. For instance, this bound plays a crucial role in analyzing the global message broadcast algorithms [31] where the reception of *any* message is usually sufficient to advance the algorithm. The progress bound was first introduced and explicitly specified in [31, 36] but it had already been implicitly used in (the analysis of) many previous works [19, 23, 24, 25, 26, 29]. Both acknowledgment and progress bounds typically depend on two parameters, the maximum contention Δ and the network size n.

1.3 The Standard Approach: the Decay Protocol

The standard approach for contention management in mutli-hop radio networks is the *Decay* protocol introduced by Bar-Yehuda, Goldreich and Itai in 1987 [19]. The core idea in the Decay

	Classical Model	Dual Graph Model
Ack. Upper	$O(\Delta \log n)$**	$O(\Delta' \log n)$**
Ack. Lower	$\Omega(\Delta \log n)$*	$\Omega(\Delta' \log n)$*
Prog. Upper	$O(\log \Delta \log n)$	$O(\Delta' \log n)$**
Prog. Lower	$\Omega(\log \Delta \log n)$**	$\Omega(\Delta' \log n)$*

Figure 1-1: A summary of our upper and lower bound results for *acknowledgment* and *progress* for the local broadcast problem. Results that are new, or significant improvements over the previously best known result, are marked with an "*" while a "**" marks results that where obtained from prior work via minor tweaks.

protocol is that nodes cycle through a number of exponentially decreasing transmission probabilities, with the hope that one of these transmission probabilities will be appropriate for the current level of contention. In more detail, the Decay protocol works as follows: Let Δ be the maximum contention. Rounds are divided into phases, each consisting of $\lceil \log \Delta \rceil$ consequent rounds, and in each phase, processes that have a message for transmission transmit their messages based on the following probabilistic rule: for each $i \in [1, \lceil \log \Delta \rceil]$, each process that has a message for transmission transmits its message with probability 2^{-i}, and remains silent otherwise. One can easily see that with this transmission rule, in each phase, each process that has at least one neighbor with a message for transmission receives at least one message, with probability at least a positive constant. Therefore, in $\Theta(\log n)$ phases, each process that has at least one neighbor with a message for transmission receives at least one message with high probability. This means that the Decay protocol has a progress bound of $O(\log \Delta \log n)$ rounds. From this fact, and noting the symmetry of the probabilities for different sender processes, one can conclude that the Decay protocol has an acknowledgment bound of $O(\Delta \log \Delta \log n)$ rounds.

This simple, randomized and distributed protocol was first introduced in [19] as a submodule for solving the global broadcast problem. It was subsequently adapted to resolve contention in numerous wireless algorithms (e.g., [29, 31, 38]). Then, in [36], this commonly-used strategy was formalized as a solution to the local broadcast problem in the classical model.

1.4 Our Results

The simplicity of the *Decay* protocol, and the fact that it is the commonly-accepted standard contention management technique for classical radio networks raises the important question that (1) "Is *Decay*-style contention management optimal for classical radio networks?"

Moreover, one might ask (2) "Can similar strategies solve the local broadcast problem when unreliability is admitted, e.g., in the dual graph model?" Also, it is interesting to ask (3) "Are there major differences between the time bounds of the local broadcast algorithms in unreliable versus

reliable radio networks?" This last question is important because any major difference would identify cases in which one should be careful about trusting solutions analyzed in the classical model to work correctly or efficiently in a real world deployment where unreliable links are unavoidable.

In this thesis, we answer the above questions and essentially provide a complete characterization of the local broadcast problem. We do this by providing matching upper and lower bounds for both the acknowledgment and the progress bounds, and in both the classical radio network model and the dual graph model. Figure 1-1 shows a summary of these bounds.

1.4.1 Lower Bounds

As our main technical contribution, we present lower bounds for both progress and acknowledgment bounds in both the classical and the dual graph model. All these lower bounds hold even for centralized algorithms.

In Corollary 11.1.6 we show a $\Omega(\log \Delta \log n)$ lower bound for progress in the classical model. In Corollary 11.1.7, we show that $\Omega(\Delta \log n)$ is a lower bound on the acknowledgment in the classical model. These two bounds show that the Decay strategy is almost optimal for both progress and acknowledgment in the classical model. This answers the question (1) above in the affirmative.

Second, we turn our attention to lower bounds for dual graph model. We show in Corollary 11.1.8 and Corollary 11.1.8 that $\Omega(\Delta' \log n)$ is a lower bound for both the progress and the acknowledgement in the dual graph model, where Δ' is the maximum contention in the dual graph network.

1.4.2 Upper Bounds

To cement our lower bounds and complete the picture, we show in Chapter 10 that a variant of the Decay protocol achieves upper bounds that match these lower bound, in both the classical and the dual graph model.

As previously mentioned, in the classical model, the original Decay protocol has progress and acknowledgment bounds of $O(\log \Delta \log n)$ and $O(\Delta \log \Delta \log n)$, respectively. Corollary 11.1.6 shows that this progress bound is already optimal. We present a slightly optimized version of the Decay protocol that, while keeping the progress bound unchanged, achieves the acknowledgment bound $O(\Delta \log n)$, in the classical model (Theorem 10.2.1). This acknowledgment bound matches Corollary 11.1.7.

We also show that our optimized variant of the Decay protocol achieves the progress bound $O(\Delta' \log n)$ and the acknowledgment bound $O(\Delta' \log n)$, in the dual graph model (Theorem 10.2.1). These upper bounds match the lower bounds of Corollary 11.1.8 and Corollary 11.1.8, respectively.

The upper bound results for the dual graph model answer question (2) above in affirmative. Moreover, the $O(\log \Delta \log n)$ progress upper bound of classical model along with the $\Omega(\Delta \log n)$ lower bound of the dual graph model demonstrate an exponential gap between the progress bounds in two models. This provides a positive response for the question (3) above and implies that progress is provably harder (slower) in the face of unreliability.

We remark here that the main results in this thesis are based on a joint work with Bernhard Haeupler, Calvin Newport and Nancy Lynch [41, 42].

1.5 Organization

The local broadcast problem, in its full generality, assumes that different processes can keep receiving broadcast requests (i.e., messages to be broadcast to their neighbors) as time continues. This describes the practical reality of contention management, which is an ongoing process. Our algorithm works in this general setting. To present the core of our lower bounds in a cleaner format, we use a simplified setting which we call the *single-shot setting*: the network is a bipartite network composed of two sides, called *senders* and *receivers*. Each sender has a message (from the start) and it has to deliver it to all of its receiver neighbors in the reliable part of the network. Therefore, in particular, this single-shot setting does not include an environment (which generates broadcast requests continuously). This setting is significantly simpler than the general ongoing case of the local broadcast problem, which we call the *multi-shot* setting. Thanks to this single-shot setting, we are able to present the core of our lower bounds away from the complications needed for the generality of the local broadcast problem. We later show that these lower bounds carry over to the multi-shot setting. Having this in mind, the organization of the main body of the thesis is divided into two parts: Part I, where we present the simplified single-shot setting and the related lower bounds, and Part II, where we present the general multi-shot setting and the related upper and lower bounds.

The more specific organization of the thesis is as follows. We start with presenting some mathematical notations and basic probabilistic inequalities in Chapter 2. These are used in both parts of the thesis. Then, the rest of the thesis is divided into two parts:

- **Part I**: In Chapter 3, we present the models that we use for the single-shot setting. Chapter 4 presents the statement of the local broadcast problem in this setting and the measures that we use for analyzing the performance of the related algorithms. Chapters 5 and 6 presents our lower bounds for the single-shot setting of the classical and the dual graph models, respectively.

- **Part II**: In Chapter 7, we present the models that we use for the multi-shot setting. Chapter

8 presents the statement of the local broadcast problem in the multi-shot setting and the measures that we use for analyzing the performance of the related algorithms. In Chapter 9, we present the related work. In Chapter 10, we present our local broadcast algorithm for both the classical and the dual graph models of this general setting. In Chapter 11, we explain how the lower bounds of the single-shot setting, presented in Chapters 5 and 6, extend to the general multi-shot setting.

We conclude this thesis in Chapter 12.

Chapter 2

Mathematical Preliminaries

In this chapter, we define the notations used throughout this thesis and we also review some probability inequalities.

2.1 Notations

- We use the notations $\mathbb{R}$ and $\mathbb{R}^+$ to denote the set of real numbers and the set of positive real numbers, respectively. We also use the notation $\mathbb{N}$ to denote the set of all positive integers.
- We use the notation $[r, r']$, for integers r and $r' \geq r$, to indicate the sequence $\{r, ..., r'\}$. We also use the notation $[r]$ for integer r to indicate $[1, r]$.
- We use the notation 2^S to denote the power set of set S, i.e., the set of all subsets of S.
- We use the notation Σ_S to denote the set of all finite length sequences over set S, i.e., sequences $\{a_k\}_{k\in\mathbb{N}}$ such that for each $i \in [1, k]$, $a_i \in S$.
- For a graph $H = (V, E)$, for each node $u \in V$, the notation $\mathcal{N}_H(u)$ describes the set of neighbors of u in H. Moreover, we define $\mathcal{N}_H^+(u) = \mathcal{N}_H(u) \cup \{u\}$.
- We use symbols $\perp$ and $\top$ to indicate two special values. These special values respectively indicate *silence* and *collision*. For example, transmitting $\perp$ means remaining silent. We explain the meaning and the usage of these symbols in Section 7.4. We use $\mathcal{M}$ to denote the set of all messages and we assume that $\mathcal{M} \cap \{\perp, \top\} = \emptyset$. We use notations $\mathcal{M}_\perp$ and $\mathcal{M}_{\perp\top}$ to denote sets $\mathcal{M} \cup \{\perp\}$ and $\mathcal{M} \cup \{\perp, \top\}$, respectively.
- We use the notation *w.h.p.* (*with high probability*) to indicate a probability at least $1 - \frac{1}{n}$, where n is the number of the nodes in the network. We present the details of the graph model of the network in Section 7.1

2.2 Some Probability Inequalities

Theorem 2.2.1 (Union Bound). *For any probability space and arbitrary events $\mathcal{E}_1$, $\mathcal{E}_2$, ..., $\mathcal{E}_k$ in this space, we have*

$$Pr[\bigcup_{i=1}^{k} \mathcal{E}_i] \leq \sum_{i=1}^{k} Pr[\mathcal{E}_i]$$

Theorem 2.2.2 (Chernoff Bound). *Let X_1, X_2, ..., X_k be independent Poisson trials such that for each $i \in [1, k]$, $\Pr[X_i = 1] = p_i$, where $p_i \in (0, 1)$. Let $X = \sum_{i=1}^{k} X_i$, and $\mu = \mathbb{E}[X]$. For any $\delta > 0$, we have*

$$Pr[X > (1+\delta)\mu] \leq e^{-\frac{\mu\delta^2}{3}}$$

and

$$Pr[X < (1-\delta)\mu] \leq e^{-\frac{\mu\delta^2}{2}}$$

Next, we present a theorem by Fortuin, Kasteleyn, Ginibre, commonly referred to as the FKG inequality [7, Chapter 6], and a simple corollary of it which we use to prove Theorem 5.2.1. We start by presenting some definitions.

Definition 2.2.3. *A finite* lattice $(L, \leq_L)$ *is a finite set L partially ordered by $\leq_L$, in which every two elements $x, y \in L$ have a least upper bound, denoted by $x \vee$, and a greatest lower bound, denoted $x \wedge y$. A lattice $(L, \leq_L)$ is* distributive *if for all $x, y, z \in L$, we have $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.*

Definition 2.2.4. *Suppose $(L, \leq_L)$ is a finite lattice. A function $f : L \to \mathbb{R}$ is called* non-decreasing *(resp., non-increasing) with respect to $\leq_L$ if $x \leq_L y$ implies $f(x) \leq f(y)$ (resp., if $x \leq_L y$ implies $f(x) \geq f(y)$).*

Definition 2.2.5. *Suppose $(L, \leq_L)$ is a finite lattice. A function $\mu : L \to \mathbb{R}^+$ is called* log-supermodular *if for all $x, y \in L$, we have $\mu(x)\mu(y) \leq \mu(x \wedge y)\mu(x \vee y)$.*

Theorem 2.2.6 (FKG Inequality). *Let $(L, \leq_L)$ be a finite distributive lattice and let $\mu : L \to \mathbb{R}^+$ be a log-supermodular function. If $f, g : L \to \mathbb{R}$ are both non-decreasing functions with respect to $\leq_L$, then we have*

$$\big(\sum_{x \in L} f(x)g(x)\mu(x)\big) \cdot \big(\sum_{x \in L} \mu(x)\big) \geq \big(\sum_{x \in L} f(x)\mu(x)\big) \cdot \big(\sum_{x \in L} g(x)\mu(x)\big)$$

Corollary 2.2.7. *Consider an arbitrary integer $K > 0$ and suppose that A_1 to A_ℓ are ℓ fixed arbitrary subsets of set $[K]$. Choose a subset $B \subseteq [K]$ as follows: for each $k \in [K]$, include k in set B independently with probability $p \in [0, 1]$. We have*

$$\Pr\Big[\forall i \in [1, \ell], A_i \cap B \neq \emptyset\Big] \geq \prod_{i=1}^{\ell} \Pr[A_i \cap B \neq \emptyset]$$

Proof. We first show that for each $j \in [2, \ell]$, we have

$$\Pr\left[A_j \cap B \neq \emptyset \,\middle|\, \forall i \in [1, j-1], A_i \cap B \neq \emptyset\right] \geq \Pr[A_j \cap B \neq \emptyset]$$

Let L be the set of all subsets of $[K]$ ordered by inclusion, i.e., for each two subsets $S, S' \subseteq [K]$, we have $S \leq_L S'$ iff $S \subseteq S'$. With this order, for each two subsets $S, S' \subseteq [K]$, we have $S \wedge S' = S \cup S'$ and $S \vee S' = S \cap S'$. Thus, for each three subsets $S, S', S'' \subseteq [K]$, $S \wedge (S' \vee S'') = S \cup (S' \cap S'') = (S \cup S') \cap (S \cup S'') = (S \wedge S') \vee (S \wedge S'')$, which shows that L with the given order is a distributive lattice. Consider the function $\mu : L \to [0, 1]$ where for any $S \subseteq [K]$, we have $\mu(S) = p^{|S|}(1-p)^{K-|S|}$. It is easy to check that μ is log-supermodular. That is because, for each two subsets $S, S' \subseteq [K]$,

$$\begin{aligned}
\mu(S)\mu(S') &= \big(p^{|S|}(1-p)^{K-|S|}\big)\big(p^{|S'|}(1-p)^{K-|S'|}\big) = p^{|S|+|S'|}(1-p)^{2K-|S|-|S'|} \\
&= p^{|S|+|S'|-|S\cap S'|}p^{|S\cap S'|} \cdot (1-p)^{K-(|S|+|S'|-|S\cap S'|)}(1-p)^{K-(|S\cap S'|)} \\
&= \big(p^{|S\cup S'|}(1-p)^{K-(|S\cup S'|)}\big)\big(p^{|S\cap S'|}(1-p)^{K-(|S\cap S'|)}\big) \\
&= \mu(S \cup S')\mu(S \cap S') = \mu(S \wedge S')\mu(S \vee S')
\end{aligned}$$

Note that the function μ is chosen such that $\mu(S) = \Pr[B = S]$.

Now, fix any $j \in [2, \ell]$ and consider indicator functions $f, g : L \to \{0, 1\}$ as follows: for each set $S \subseteq [K]$, $f(S) = 1$ iff $S \cap A_j \neq \emptyset$ and $g(S) = 1$ iff $\forall i \in [1, j-1]$, we have $A_i \cap B \neq \emptyset$. Clearly, f and g are both non-decreasing with respect to the inclusion order. With these definitions, it follows from the FKG inequality (Theorem 2.2.6) that

$$\begin{aligned}
\Pr\left[\forall i \in [1, j], A_i \cap B \neq \emptyset\right] \cdot 1 &= \big(\sum_{x \in L} f(x)g(x)\mu(x)\big) \cdot \big(\sum_{x \in L} \mu(x)\big) \\
\geq \big(\sum_{x \in L} f(x)\mu(x)\big) \cdot \big(\sum_{x \in L} g(x)\mu(x)\big) &= \Pr[A_j \cap B \neq \emptyset] \cdot \Pr\left[\forall i \in [1, j-1], A_i \cap B \neq \emptyset\right]
\end{aligned}$$

Now dividing the two sides by $\Pr\left[\forall i \in [1, j-1], A_i \cap B \neq \emptyset\right]$, we have

$$\Pr\left[A_j \cap B \neq \emptyset \,\middle|\, \forall i \in [1, j-1], A_i \cap B \neq \emptyset\right] \geq \Pr[A_j \cap B \neq \emptyset]$$

Given this inequality for any $j \in [2, \ell]$, we can complete the proof of the corollary easily as follows:

$$\begin{aligned}
& \Pr\left[\forall i \in [1,\ell], A_i \cap B \neq \emptyset\right] \\
= \quad & \Pr[A_1 \cap B \neq \emptyset] \cdot \prod_{j=2}^{\ell} \Pr\left[A_j \cap B \neq \emptyset \,\middle|\, \forall i \in [1, j-1], A_i \cap B \neq \emptyset\right] \\
\geq \quad & \Pr[A_1 \cap B \neq \emptyset] \cdot \prod_{j=2}^{\ell} \Pr[A_j \cap B \neq \emptyset] = \prod_{j=1}^{\ell} \Pr[A_j \cap B \neq \emptyset]
\end{aligned}$$

□

Part I

The Simplified Single-Shot Setting and Lower Bounds

Chapter 3

The Models in The Single-Shot Setting

In this chapter, we present the definitions of the models that we use for the simplified single-shot setting. We use almost the same models in the second part of the thesis when studying the general multi-shot case of the problem, with the exception of a small number of changes which we briefly mention in this chapter and we explain in detail in Chapter 7.

As mentioned in the introduction, we use two models, namely the *classical radio network model* (also known as the radio network model) and the *dual graph model*. The former model has been extensively studied since the late 1980s [19]-[29], [31, 36, 40] and assumes that all connections in the network are reliable. The latter model was introduced recently in 2009 [31, 33], and is more general in that it includes the possibility of unreliable edges. Since the former model is simply a special case of the latter, we use the dual graph model for defining both models, and also when describing the problem statement in the next chapter. However, in some places, we indicate how a certain result or property changes when we focus on the special case of the classical radio network model.

In the dual graph model of the single-shot setting, a distributed system is composed of a set of processes that are connected to each other via a network, and an adversary that controls the communications on this network to a certain extent (to be explained in Section 3.3). The main difference between this model and the model for the multi-shot setting that we explain in Chapter 7 is that, in the multi-shot version, the distributed system also includes an environment. This environment is an abstraction of the higher levels of the wireless nodes that interact with the local broadcast module (abstracted as processes here). In Chapter 7, we present the details of the definition of the environment and explain the interactions between the environment and the processes.

The rest of this chapter is organized as follows: In Section 3.1, we present the network connection assumptions for the dual graph model. We explain what a process in this model is in Section 3.2. In Section 3.3, we explain how a distributed system in the single-shot setting works as a whole, i.e., what are the executions of an algorithm in this model and what is the role of the adversary. In

all Sections 3.1 to 3.3, we try to present the model as general as possible, in order to keep it similar to its counterpart in the multi-shot setting (Chapter 7). However, in our lower bounds (all results in Chapters 5 and 6), we use a stronger model, *centralized setting*, which we explain in Section 3.4.

3.1 Network Connections

In the dual graph model, radio networks have some reliable links and potentially some unreliable links. In the dual graph model, we define a network (G, G') to consist of two undirected finite graphs, $G = (V, E)$ and $G' = (V, E')$, where we have $E \subseteq E'$. Intuitively, set E is the set of reliable edges while E' is the set of all edges (both reliable and unreliable). We assume that the communications on the unreliable edges, i.e., the edges in set $E' \setminus E$, are controlled by an adversary. We explain this issue in more detail in Section 3.3. When restricting attention to the special case of the classical radio network model, there are no unreliable edges and thus, we simply have $G = G'$, i.e., $E = E'$. We define the size of the network to be $n = |V|$. We remark that graphs G and G' can be disconnected.

We assign processes to graph nodes of the network (G, G'). This assignment is defined by an injective function $id()$ from V to the set of process ids $[N]$ (refer to Section 3.2 for process definitions.). That is, for each $v \in V$, $id(v)$ indicates the id of the process assigned to graph node v. For each $v \in V$, we use notation $proc(v)$ to indicate the process with id $id(v)$, i.e., the process that is assigned to graph node v. We sometimes abuse notation by using the notation *process* u, or sometimes just u, for some graph node $u \in V$, to refer to $proc(u)$. We refer to a network (G, G') and a mapping $id()$ from graph nodes to processes as a *setting*.

Processes can potentially know the network (G, G') or have some partial knowledge about it[1]. This means that processes could have a full description of the network or some information about it built into their states, and the related distributed algorithm is required to work only if this full description matches the description of the network or if this partial information is consistent with the description of the network. To strengthen our results, we remark that lower bounds (all results in Chapters 5 and 6) allow full knowledge of the network graphs (G, G').

3.2 Distributed Algorithms and Processes

We define a distributed algorithm $\mathcal{A}$ to be a collection of N randomized processes, where N is an arbitrary positive integer. Processes are described by probabilistic automata and intuitively, in each process, we have two types of transitions: (1) probabilistic transitions that take the state at the

[1]This is because, in many real world settings, it is reasonable to assume that devices can make some assumptions or inference about the structure of the network.

beginning of a round to an intermediate state and a message to be transmitted on the channel (or silence), (2) deterministic transitions that take an intermediate state and a message received from the channel (or a special value $\perp$ or $\top$) to the state at the end of a round.

The formal definition of processes is as follows: Each process in $\mathcal{A}$ is a 6-tuple $(i, \mathcal{Q}^i, \mathcal{P}^i, S_0^i, \mathcal{F}^i, \mathcal{G}^i)$ and we have:

- $i \in [N]$ is the unique identifier of this process.
- $\mathcal{Q}^i$ and $\mathcal{P}^i$ are two sets of states, and we have $\mathcal{Q}^i \cap \mathcal{P}^i = \emptyset$. We refer to states in these two sets respectively as $\mathcal{Q}$-states and $\mathcal{P}$-states.
- $S_0^i \in \mathcal{Q}^i$ is a single starting state.
- $\mathcal{F}^i$ is a function that captures the probabilistic transitions of the process. For each state $S \in \mathcal{Q}^i$, function $\mathcal{F}^i(S) : \mathcal{P}^i \times \mathcal{M}_\perp \to [0, 1]$ is a probability distribution function over $\mathcal{P}^i \times \mathcal{M}_\perp$. That is, for each state $S' \in \mathcal{Q}^i$, and each $m \in \mathcal{M}_\perp$, $\mathcal{F}^i(S)(S', m)$ is the probability that, given that the process with id i is in state S, it makes a transition to state S' and transmits m if $m \neq \perp$ and remains silent if $m = \perp$. Note that, as mentioned in Section 2.1, in this notation, symbol $\perp$ means remaining silent.
- $\mathcal{G}^i : \mathcal{P}^i \times \mathcal{M}_{\perp\top} \to \mathcal{Q}^i$ is a function that captures the deterministic transitions of the process. For each state $S' \in \mathcal{P}^i$ and each $m' \in \mathcal{M}_{\top\perp}$, $\mathcal{G}^i(S', m')$ is a state in $\mathcal{Q}^i$, and we have the following: if the process with id i is in $\mathcal{P}$-state S' and it receives m' from the channel, then it makes a transition to state $\mathcal{G}^i(S', m')$.

For simplicity, we sometimes use *process* i to refer to the process with id i.

3.3 Executions in the Dual Graph Model

As mentioned at the start of this chapter, in a distributed system in the dual graph model, a set of processes are connected to each other via a network as described in Section 3.1, where the communications over this network are controlled by an adversary. In this section, we explain how this system works as a whole by describing the executions of a distributed algorithm in this model and explaining the role of the adversary. An execution of an algorithm $\mathcal{A}$ in network (G, G') proceeds as follows:

The execution proceeds in synchronous lock-step rounds $1, 2, ...$, where all the participating processes start in the first round. At the start of each round r, every process $proc(u), u \in V$ is in a state $S \in \mathcal{Q}^i$, where $i = id(u)$. In particular, at the start of the first round, $proc(u)$ is in state S_0^i. Then, using function $\mathcal{F}^i(S)$, $proc(u)$ performs its probabilistic state transition to a $\mathcal{P}$-state

$S' \in \mathcal{P}^i$, and it also determines a value $m \in M_\perp$. If $m' \neq \perp$, then $proc(u)$ transmits message m' in round r. Otherwise (i.e., if $m' \neq \perp$), $proc(u)$ remains silent in round r. That is, as explained in Section 3.2, choosing $m' = \perp$ during this transition means that process i remains silent in round r.

Next, the adversary chooses a *reach set* that consists of E and a subset, potentially empty, of edges in $E' - E$. This reach set potentially affects what message (or value $\perp$ or $\top$) each process receives. Intuitively, this reach set describes the links that are active in this round. We emphasize that when focusing on the special case of the classical model, set $E' - E$ is empty and therefore, the reach set is just E. In the dual graph model, we assume that the adversary is *'adaptive offline'* [8] meaning that it has full knowledge of the history of the execution and in particular the state of the network when it is determining the reach sets. This means that when choosing the reach set of round r, the adversary knows everything that happened up to round r of the execution including the outcome of the random coins used for the transition of round r. In particular, the adversary knows which processes are transmitting in round r. Moreover, we assume that the adversary can also make randomized decisions.

After the adversary determines the reach set of round r, depending on this reach set and which processes are transmitting, each process i receives exactly one value in set $\mathcal{M}_{\perp\top}$ from the channel. For a graph node v, let $B_{v,r}$ be the set of all graph nodes u such that $proc(u)$ transmits in r and edge $e = \{u, v\}$ is in the reach set for this round. What $proc(v)$ receives in round r is determined by the following rules:

(A) If $proc(v)$ broadcasts in round r, then it receives only its own message.

(B) If $proc(v)$ does not broadcast, and $|B_{v,r}| = 0$ or $|B_{v,r}| > 1$, then $proc(v)$ receives $\perp$ (indicating *silence*).

(C) If $proc(v)$ does not broadcast, and $|B_{v,r}| = 1$, then $proc(v)$ receives the message sent by $proc(u)$, where u is the single node in $B_{v,r}$.

The rule (A) intuitively means that each process cannot send and receive simultaneously. We remark that the rule (B) means that we do not assume any *collision detection* mechanism in this model. To strengthen our results, we present our lower bounds (all results in Chapters 5 and 6) in the stronger model with collision detection, i.e., where the rule (B) is replaced with the following rule

(B') If $proc(v)$ does not broadcast, and $|B_{v,r}| = 0$, then $proc(v)$ receives $\perp$. If $proc(v)$ does not broadcast, and $|B_{v,r}| > 1$, then $proc(v)$ receives $\top$ (indicating *collision*).

After receiving the messages of round r, every process $proc(u), u \in V$ makes its deterministic transition using function $\mathcal{G}^i$, where $i = id(u)$. That is, suppose process $proc(u), u \in V$ is in a

$\mathcal{P}$-state $S' \in \mathcal{P}^i$, where $i = id(u)$ and it receives $m' \in M_{\perp\top}$ from the channel. Then $proc(u)$ makes a transition to $\mathcal{Q}$-state $\mathcal{G}^i(S', m')$. We remark that at the end of each round, the state of the whole system consists of just $\mathcal{Q}$-states for all processes.

3.4 Centralized Setting For Lower Bounds

In our lower bounds (all results in Chapters 5 and 6), we consider the stronger model of *centralized* algorithms. We define a centralized algorithm to be the same as the distributed algorithms explained in this chapter, with two modifications: (1) the processes know the graph (G, G') and the mapping $id()$ from the beginning of the execution; and (2) when the processes are making their transitions, they know the full history of the execution and thus, their transitions are a function of the full history of the execution. This history in particular includes the current state of all the processes in the network.

Finally notice that in the centralized setting, since each process knows the full history of the execution, in each round r, each process knows exactly which set of its neighbors transmitted. Thus, each process knows whether zero or two or more of its neighbors transmitted in that round. Thus, in the centralized setting, the models with and without collision detection are equivalent.

Chapter 4

The Local Broadcast Problem in The Single-Shot Setting

The intuitive description of the *local broadcast problem* in the single-shot setting is as follows: The nodes of the network are divided into two groups, the senders and the receivers. Moreover, no two senders are neighbors and no two receivers are neighbors. The objective of the problem is that, each sender should deliver a special message that it has to all of its receiver neighbors. In this chapter, we present the formal definition of this problem, and present the time bounds that we use to measure the performance of the algorithms that solve this problem.

4.1 The Local Broadcast Problem

In this single-shot setting, the processes, the network and the executions are as presented in Chapter 3. In the version of the local broadcast problem tailored to this setting, we moreover have:

1. The network (G, G') consists of two undirected finite bipartite graphs, $G = (V, E)$ and $G' = (V, E')$, where we have: $V = S \cup R$, $S \cap R = \emptyset$, $|S| \geq 1$, $|R| \geq 1$, $|V| = n$, and $E \subseteq E'$. Moreover, each edge $e \in E'$ is an unordered pair $\{v, u\}$ such that $v \in S$ and $u \in R$. The nodes in sets S and R are respectively called the *senders* and the *receivers*. Moreover, the processes assigned to the sender nodes and the receiver nodes are respectively called the *sender processes* and the *receiver processes*.

2. Each process i has one special message $m_i \in M$ (encoded in all of its states, and particularly its starting state S_0^i) and for each $i, j \in N$ such that $i \neq j$, we have $m_i \neq m_j$.

3. Each process i has a boolean variable $ack_i \in \{False, True\}$ (in each of its states). In starting state S_0^i, we have $ack_i = False$. Each process i can change the value of variable

ack_i at most once and thus, only from a $False$ value to a $True$ value. Moreover, this change can only happen during a transition from a $\mathcal{P}$-state to a $\mathcal{Q}$-state (the second transition of a round). Formally these two conditions mean that (1) there is no transition from a $\mathcal{Q}$-state Q to a $\mathcal{P}$-state P such that the value of ack_i in states Q and P is different, and (2) there is no state transition from the $\mathcal{P}$-states in which $ack_i = True$ to the $\mathcal{Q}$-states in which $ack_i = False$.

The informal description of the *local broadcast problem* in this setting is that for each sender node $v \in S$, process $proc(v)$ should eventually deliver special message m_i, where $i = id(v)$, to all of its receiver G-neighbors, with high probability. This is formalized as follows: We say that an algorithm $\mathcal{A}$ solves the local broadcast problem provided that, when $\mathcal{A}$ operates in any dual graph (G, G'), we have:

(A) In every execution, for each sender node $v \in S$, process $proc(v)$ eventually sets $ack_i = True$, where $i = id(v)$.

(B) For each particular sender node $v \in S$ and each round r, if process $proc(v)$ has $ack_i = True$, where $i = id(v)$, at the end of round r, then with high probability, for each (receiver) node $u \in \mathcal{N}_G(v)$, process $proc(u)$ has received message m_i by the end of round r. Here, the probability space is based on all the probabilistic choices of the algorithm and the adversary. Moreover, the probability distribution in this requirement is conditional on the event that $proc(v)$ has $ack_i = True$ at the end of round r.

An algorithm $\mathcal{A}$ that solves the local broadcast problem is called a *local broadcast algorithm*.

4.2 The Time Bounds

We measure the performance of a local broadcast algorithm by two bounds: the *acknowledgment bound* and the *progress bound*. For any given local broadcast algorithm $\mathcal{A}$ and any fixed single-shot setting (G, G'), these bounds are defined as follows:

1. A number $t \in \mathbb{N}$ is an *acknowledgment bound* for $\mathcal{A}$ in (G, G') if for each sender process $proc(v)$, at the end of round t, with high probability, $proc(v)$ has $ack_i = True$, where $i = id(v)$.

2. A number $t \in \mathbb{N}$ is a *progress bound* for $\mathcal{A}$ in (G, G') if for each receiver process $proc(u)$ such that node $|\mathcal{N}_G(u)| \geq 1$, with high probability, $proc(u)$ has received at least one special message m_i for an i such that process $i = id(v)$ and $v \in \mathcal{N}_G(v)$, by the end of round t.

In the following lemma, we show that progress bound is less than or equal to acknowledgment bound. The formal statement is as follows:

Lemma 4.2.1. *For any local broadcast algorithm $\mathcal{A}$, any single-shot setting (G, G'), and any $\tau \in \mathbb{N}$, if τ is also an acknowledgment bound for $\mathcal{A}$ in (G, G'), then τ is a progress bound for $\mathcal{A}$ in (G, G').*

Proof. Consider an arbitrary local broadcast algorithm $\mathcal{A}$, a single-shot setting (G, G'), and a $\tau \in \mathbb{N}$ such that τ is an acknowledgment bound for $\mathcal{A}$ in (G, G'). Following the definition of acknowledgment bound, we get that in each execution of $\mathcal{A}$, by the end of round τ, for each sender process with id i we have $ack_i = True$. Moreover, following the definition of the local broadcast problem, we get that in executions of $\mathcal{A}$, by the end of round τ, with high probability we have that each receiver process $proc(u)$ has received the special message m_i, for every i such that $i = id(v)$ and $v \in \mathcal{N}_G(u)$. Thus, by the end of round τ, with high probability, we have that each receiver process $proc(u)$ such that node $|\mathcal{N}_G(u)| \geq 1$ has received at least one special message m_i for an i such that process $i = id(v)$ and $v \in \mathcal{N}_G(v)$. Comparing this with the definition of progress bound shows that τ is a progress bound for $\mathcal{A}$ in (G, G'). □

The acknowledgment and the progress time bounds defined above often depend on the maximum contention of the network, which is defined as follows:

Definition 4.2.2. *In the single-shot setting, the* maximum contention Δ' *(resp., Δ in the classical model) is equal to the maximum G'-degree (resp., G-degree in the classical model) of the receiver nodes.*

We sometimes use the phrase the *maximum receiver degree* instead of the maximum contention. The definition of the maximum contention in the multi-shot setting has similarities to this definition, but is more complicated and requires careful definitions for the amount of contention in each round and for each node. We present that definition in Chapter 8.5.

Chapter 5

Lower Bounds in the Classical Radio Broadcast Model

In this chapter, we focus on the problem of local broadcast in the single-shot setting of the classical model (for formal definitions of this setting, refer to Chapters 3 and 4). We present lower bounds for both the progress and the acknowledgment time bounds. We emphasize that all these lower bounds are presented for centralized algorithms and also, in the model where processes are provided with a collision detection mechanism. Note that these points only strengthen these results.

In Chapter 11 we explain that since the single-shot setting can be viewed as a special case of the multi-shot setting, these lower bounds extend to the multi-shot setting as well. Thus, they prove that the optimized decay protocol for the general multi-shot setting, which we present in Chapter 10, is optimal with respect to progress and acknowledgment times in the classical model. These lower bounds also show that the existing constructions of Ad Hoc Selective Families [27, 28] are optimal. Moreover, in Chapter Chapter 6, we use the lower bound on the acknowledgment time in the classical model that we present here as a basis for deriving lower bounds for progress and acknowledgment times in the dual graph model.

5.1 Progress Time Lower Bound

In this section, we remark that, following the proof of the $\Omega(\log^2 n)$ lower bound of Alon et al. [21] on the time needed for global broadcast of one message in radio networks, and with slight modifications, one can get a lower bound of $\Omega(\log \Delta \log n)$ on the progress bound in the classical model.

Theorem 5.1.1. *For any sufficiently large n and any $\Delta \leq n$, there exists a single-shot setting in the classical model with a bipartite network $\mathcal{H}(n, \Delta)$ of size n and maximum contention of at most*

Δ, *such that for any local broadcast algorithm, the progress bound in* $\mathcal{H}(n, \Delta)$ *is greater than* $\Omega(\log \Delta \log n)$ *rounds.*

Proof Outline. The proof is an easy extension of [21] to networks with maximum contention Δ. This proof uses the probabilistic method [7] to show that such a network $\mathcal{H}(n, \Delta)$ exists. The only change from [21] is that instead of choosing the receiver degrees to vary between $n^{0.4}$ and $n^{0.6}$, we choose the degrees between $\Theta(\Delta^{1/4})$ and $\Theta(\Delta^{1/2})$. This leads to $\Theta(\log \Delta)$ (instead of $\Theta(\log n)$) different classes of degrees, and in turn, to the stated bound. The rest of the proof remains the same as in [21]. □

5.2 Acknowledgment Time Lower Bound

In this section, we present our lower bound on the acknowledgment time in the classical radio broadcast model.

Theorem 5.2.1. *For any sufficiently large* n *and any* $\Delta \in [20 \log n, n^{0.1}]$*, there exists a single-shot setting in the classical model with a bipartite network* $\mathsf{H}(n, \Delta)$ *of size* n *and maximum contention of at most* Δ*, such that the acknowledgment bound of any algorithm in* $\mathsf{H}(n, \Delta)$ *is greater than* $\frac{\Delta \log n}{100}$ *rounds.*

In the proof of this theorem, instead of showing directly that randomized algorithms have low success probability, we show a stronger variant, by proving an impossibility result: In Lemma 5.2.2, we prove that there exists a single-shot setting with a bipartite network $\mathsf{H}(n, \Delta)$ of size n and maximum contention at most Δ in which, even with a centralized algorithm, it is *not possible* to schedule transmissions of nodes in at most $\frac{\Delta \log n}{100}$ rounds such that each receiver receives the message of each of its neighboring senders. In particular, this result shows that in $\mathsf{H}(n, \Delta)$, for any randomized local broadcast algorithm, the probability that in at most $\frac{\Delta \log n}{100}$ rounds, each receiver receives the message of each of its sender neighbors is zero. In proof of Theorem 5.2.1 (presented at the end of this section), we argue that this means that no local broadcast algorithm has an acknowledgment bound of at most $\frac{\Delta \log n}{100}$ in $\mathsf{H}(n, \Delta)$.

To present Lemma 5.2.2, we first present some definitions. A transmission schedule σ of length $L(\sigma)$ for a bipartite network is a sequence $\sigma_1, \ldots, \sigma_{L(\sigma)} \subseteq S$ of sets of senders. Having a sender $u \in \sigma_r$ indicates that at round r the sender u is transmitting its message. For a network G, we say that transmission schedule σ *covers* G if for every $u \in R$ and every $v \in \mathcal{N}_G(u)$, there exists a round r such that $\sigma_r \cap \mathcal{N}_G(u) = \{v\}$, that is, using transmission schedule σ each receiver node receives the message of each of its sender neighbors. Now we are ready to see the main lemma which proves our bound.

Lemma 5.2.2. *For any sufficiently large n and any $\Delta \in [20 \log n, n^{0.1}]$, there exists a single-shot setting with a bipartite network $\mathsf{H}(n, \Delta)$ with size n and maximum receiver degree at most Δ, for which there does not exist a transmission schedule σ such that $L(\sigma) \leq \frac{\Delta \log n}{100}$ and σ covers $\mathsf{H}(n, \Delta)$.*

We next present the proof of Lemma 5.2.2. As in the previous section, our proof uses techniques similar to those of [20, 21, 22] and utilizes the probabilistic method [7] to show the existence of the network $\mathsf{H}(n, \Delta)$ mentioned in Lemma 5.2.2.

Proof Outline. First, we fix a sufficiently large n and $\Delta \in [20 \log n, n^{0.1}]$ and let $\eta = n^{0.12}$ and $\eta' = \eta^8 = n^{0.96}$. Note that if n is sufficiently large, we have $20 \log n \leq n^{0.1}$. Next, we present a probability distribution over a particular family $\mathcal{G}$ of bipartite networks. The common structure of this graph family $\mathcal{G}$ is as follows. All networks of $\mathcal{G}$ have a fixed set of nodes V. Moreover, V is partitioned into two nonempty disjoint sets S and R, which are respectively the set of senders and the set of receivers. We have $|S| = \eta$ and $|R| = n - \eta$. Note that for any sufficiently large n, we have $|R| = n - n^{0.12} \geq n^{0.96} = \eta'$, where the inequality holds because n is sufficiently large.

In order to define the probability distribution of these graphs, we describe the process that chooses a random network from $\mathcal{G}$. We create a random network from $\mathcal{G}$ by independently putting an edge between any $s \in S$ and $r \in R$ with probability $\frac{\Delta}{2\eta}$.

We use the following definitions. Let $\mathcal{BAD}_1$ be the event in the probability space of graphs that the maximum receiver degree of the graph $G \in \mathcal{G}$ is greater than Δ. For each transmission schedule σ, call σ *short* if $L(\sigma) \leq \frac{\Delta \log n}{100}$. Let $\mathcal{BAD}_2$ be the set of graphs $G \in \mathcal{G}$ such that that there exists a short transmission schedule that covers G.

We first show in Lemma 5.2.3 that $\Pr[\mathcal{BAD}_1] \leq \frac{1}{n^2}$. Then we show in Lemma 5.2.4 that $\Pr[\mathcal{BAD}_2] \leq \frac{1}{n^2}$. A union bound then shows that with probability at least $1 - \frac{2}{n^2} > 0$, neither of the two events $\mathcal{BAD}_1$ and $\mathcal{BAD}_2$ happen. This means there exists a graph in $\mathcal{G}$ that has the maximum receiver degree at most Δ and no short transmission schedule covers it, and thus completes the proof. □

Lemma 5.2.3. $\Pr[\mathcal{BAD}_1] \leq \frac{1}{n^2}$

Proof. We show that the probability that the maximum receiver degree of a random graph $G \in \mathcal{G}$ is greater than Δ is at most $\frac{1}{n^2}$. For each receiver $r \in R$, let $X_G(r)$ denote the degree of receiver r in graph $G \in \mathcal{G}$. Then, $\mathbb{E}[X_G(r)] = \eta \cdot \frac{\Delta}{2\eta} = \frac{\Delta}{2}$. Moreover, since edges are added independently, we can use a Chernoff bound (Theorem 2.2.2) and obtain that $\Pr[X_G(r) > \Delta] \leq e^{-\frac{\Delta}{6}}$. Using a union bound over all choices of receiver node r, and noting that $|R| < n$ and $\Delta \geq 20 \log n$, we

complete the proof as follows:

$$\begin{aligned}\Pr[\exists r \in R \; s.t. \; X_G(r) > \Delta] \leq \quad & n \cdot e^{-\frac{\Delta}{6}} < e^{\log n - \frac{\Delta}{6}} \\ < \quad & e^{\log n - 3\log n} = e^{-2\log n} < \frac{1}{n^2}\end{aligned}$$

□

Lemma 5.2.4. $\Pr[\mathcal{BAD}_2] \leq \frac{1}{n^2}$

Proof Outline. For each transmission schedule σ, let $\mathcal{E}_\sigma$ be the event in the probability space of graphs that σ covers graph $G \in \mathcal{G}$. Also, let us denote the set of all short transmission schedules by $\mathcal{SHORT}$. Having these definitions, the probability that there exists a *short* transmission schedule σ that covers $G \in \mathcal{G}$ is $\Pr[\cup_{\sigma \in \mathcal{SHORT}} \mathcal{E}_\sigma]$. That is, $\Pr[\mathcal{BAD}_2] = \Pr[\cup_{\sigma \in \mathcal{SHORT}} \mathcal{E}_\sigma]$. Thus, using a union bound, we can infer that $\Pr[\mathcal{BAD}_2] \leq \sum_{\sigma \in \mathcal{SHORT}} \Pr[\mathcal{E}_\sigma]$. Having this inequality in mind, we first show in Lemma 5.2.5 that for each $\sigma \in \mathcal{SHORT}$, $\Pr[\mathcal{E}_\sigma] \leq e^{-n^{0.72}}$. Then, we show in Lemma 5.2.7 that $|\mathcal{SHORT}| \leq 2^{n^{0.36}}$. At the end, we use the inequality $\Pr[\mathcal{BAD}_2] \leq \sum_{\sigma \in \mathcal{SHORT}} \Pr[\mathcal{E}_\sigma]$ along with Lemmas 5.2.5 and 5.2.7 to complete the proof. □

Lemma 5.2.5. *For each $\sigma \in \mathcal{SHORT}$, $\Pr[\mathcal{E}_\sigma] \leq e^{-n^{0.72}}$.*

Proof. Fix an arbitrary short transmission schedule σ. For each round t of σ, let $N(t)$ denote the number of senders that transmit in round t. Also, call round t an *isolator* if $N(t) = 1$. For each sender $s \in S$, if there exists an isolator round in σ where only s transmits in that round, then call sender s *lost*. Since $L(\sigma) \leq \frac{\Delta \log n}{100} \leq \frac{n^{0.1} \log n}{100} < \frac{n^{0.12}}{2} = \frac{\eta}{2}$, there are at least $\frac{\eta}{2}$ senders that *are not lost*. We remark that inequality $\frac{n^{0.1} \log n}{100} < \frac{n^{0.12}}{2}$ holds because n is sufficiently large.

For each not-lost sender s, we define a potential function $\Phi(s) = \sum_{t \in T_s} \frac{1}{N(t)}$ where T_s is the set of rounds in which sender s transmits. Note, that for each round t, the total potential given to not-lost senders in that round is at most $N(t) \cdot \frac{1}{N(t)} = 1$. Hence, the total potential when summed over all rounds is at most $\frac{\Delta \log n}{100} = \frac{\Delta \log \eta}{12}$. Therefore, since there are at least $\frac{\eta}{2}$ not-lost senders, there exists a not-lost sender s^* for which $\Phi(s^*) \leq \frac{\Delta \log \eta}{6\eta}$. For the rest of the proof, fix a non-lost sender s^* such that $\Phi(s^*) \leq \frac{\Delta \log \eta}{6\eta}$. Now we focus on sender s^* and rounds T_{s^*}. We show the following claim:

Claim 5.2.6. *For each receiver $r \in R$, the probability that receiver r is a neighbor of s^* and it does not receive the message of s^* is at least $\frac{1}{\eta}$.*

Proof. Consider an arbitrary receiver r. For each round $t \in T_{s^*}$, we say receiver r is *blinded* in round t if r is connected to at least one sender node other than s^* that transmits in round t. Having this definition, the proof of Claim 5.2.6 is based on two facts as follows:

(F1) $Pr[r \textit{ is a neighbor of } s^*] = \frac{\Delta}{2\eta} \geq \frac{1}{\eta}$.

(F2) $\Pr[\forall t \in T_{s^*},\ r \textit{ is blinded in round } t] \geq \frac{1}{\eta}$.

In order to prove fact (F2), we use the FKG inequality [7, Chapter 6] and particularly its simplified form presented in Corollary 2.2.7. In particular, in this application of Corollary 2.2.7, set $[K]$ is the set of senders other than s^*, the set B is the set of sender neighbors of receiver r other than sender s^*, $p = \frac{\Delta}{2\eta}$, $\ell = |T_{s^*}|$, and the A set are: for each round $t \in T_{s^*}$, we have one set A_i which is equal to the set of all senders other than s^* that transmit in round t. Thus, using Corollary 2.2.7, we get that

$$\Pr[\forall t \in T_{s^*},\ r \textit{ is blinded in round } t] \geq \prod_{t \in T_{s^*}} \Pr[r \textit{ is blinded in round } t].$$

Now for each $t \in T_{s^*}$, we have

$$\begin{aligned} \Pr[r \textit{ is blinded in round } t] = 1 - (1 - \frac{\Delta}{2\eta})^{N(t)-1} \quad &\geq 1 - e^{-\frac{\Delta}{2\eta}(N(t)-1)} \\ &\overset{(\dagger)}{\geq} 1 - e^{-\frac{\Delta}{4\eta} \cdot N(t)} \overset{(\ddagger)}{\geq} e^{-\frac{4\eta}{\Delta} \cdot \frac{1}{N(t)}}, \end{aligned}$$

where inequality $(\dagger)$ holds because $N(t) \geq 2$ as s^* is not-lost, and inequality $(\ddagger)$ holds as for any $x \geq 0$, we have $e^{-x} + e^{-1/x} \leq 1$. Hence, we have

$$\begin{aligned} \Pr[\forall t \in T_{s^*},\ r \textit{ is blinded in round } t] \geq \prod_{t \in T_{s^*}} \quad & \Pr[r \textit{ is blinded in round } t] \\ \geq \prod_{t \in T_{s^*}} \quad & e^{-\frac{4\eta}{\Delta} \cdot \frac{1}{N(t)}} = e^{-\sum_{t \in T_{s^*}} \frac{4\eta}{\Delta} \cdot \frac{1}{N(t)}} = e^{-\frac{4\eta}{\Delta} \cdot \Phi(s^*)}. \end{aligned}$$

By choice of s^*, we have $\Phi(s^*) \leq \frac{\Delta \log \eta}{6\eta}$. Thus,

$$\Pr[\forall t \in T_{s^*},\ r \textit{ is blinded in round } t] \geq e^{-\frac{4\eta}{\Delta} \cdot \Phi(s^*)} \geq e^{-\frac{4 \log \eta}{6}} = \left(e^{-2/3}\right)^{\log n} > \left(\frac{1}{2}\right)^{\log \eta} = \frac{1}{\eta}.$$

Now note that whether r is connected to each sender other than s^* is independent of whether it is connected to s^*. Thus, the two events considered in parts (F1) and (F2) are independent. Hence, for each receiver r, the probability that r is a neighbor of s^* and in each round $t \in T_{s^*}$, r is blinded is at least $\frac{1}{\eta} \cdot \frac{1}{\eta} = \frac{1}{\eta^2}$. This means that the probability that r is a neighbor of s^* but it never receives the message of s^* is at least $\frac{1}{\eta^2}$. This completes the proof of Claim 5.2.6. □

Now we use Claim 5.2.6 to complete the proof of Lemma 5.2.5. Since edges of different receivers are chosen independently, we get that the probability that there does not exist a receiver r

that is a neighbor of s^* but never receives the message of s^* is less than $(1-\frac{1}{\eta^2})^{|R|} \leq (1-\frac{1}{\eta^2})^{\eta^8} \leq e^{-\eta^6}$. Here, the first inequality holds because $|R| \geq \eta^8$. Hence, the probability that with schedule σ, each receiver receives the message of each of its sender neighbors is less than $e^{-\eta^6}$. Thus, $\Pr[\mathcal{E}_\sigma] \leq e^{-\eta^6} = e^{-n^{0.72}}$ which completes the proof of Lemma 5.2.5. □

Lemma 5.2.7. *$|\mathcal{SHORT}| \leq 2^{n^{0.36}}$, i.e., the total number of distinct short transmission schedules is at most $2^{\eta^3} = 2^{n^{0.36}}$.*

Proof. In each round, there are 2^η options for selecting which subset of senders transmit. On the other hand, each short transmission schedule has at most $\frac{\Delta \log n}{100} < \eta^2$ rounds. Therefore, the total number of ways in which one can choose a short transmission schedule is at most $2^{\eta^3} = 2^{n^{0.36}}$. □

Proof of Lemma 5.2.4. From Lemma 5.2.5, we know that for each $\sigma \in \mathcal{SHORT}$, $\Pr[\mathcal{E}_\sigma] \leq e^{-n^{0.72}}$. On the other hand, from Claim 5.2.7, we know that $|\mathcal{SHORT}| \leq 2^{n^{0.36}}$. Hence, noting that n is sufficiently large, we conclude the proof of Claim 5.2.4 as follows

$$\Pr[\mathcal{BAD}_2] = \Pr[\cup_{\sigma \in \mathcal{SHORT}} \mathcal{E}_\sigma] \leq \sum_{\sigma \in \mathcal{SHORT}} \Pr[\mathcal{E}_\sigma] \leq 2^{n^{0.36}} \cdot e^{-n^{0.72}} \leq e^{n^{0.36} - n^{0.72}} \leq e^{-2\log n} < \frac{1}{n^2}.$$

□

Having proven Lemmas 5.2.3 and 5.2.4, we now finish the proof of Lemma 5.2.2.

Proof of Lemma 5.2.2. The proof follows from Lemmas 5.2.3 and 5.2.4 and a union bound. Using a union bound, we get that $\Pr[\mathcal{BAD}_1 \cup \mathcal{BAD}_2] \leq \Pr[\mathcal{BAD}_1] + \Pr[\mathcal{BAD}_2] \leq \frac{2}{n^2} \leq \frac{1}{n}$. Thus, with probability at least $1 - \frac{1}{n}$, neither of the two events $\mathcal{BAD}_1$ and $\mathcal{BAD}_2$ happen. This means that for a random graph $G \in \mathcal{G}$, with probability at least $1 - \frac{1}{n}$, we have that G has the maximum receiver degree at most Δ and no short transmission schedule covers G. Thus, following the probabilistic method [7], we can infer that there exists a network $\mathsf{H}(n, \Delta)$ such that $\mathsf{H}(n, \Delta)$ has the maximum receiver degree of at most Δ and no short transmission schedule covers $\mathsf{H}(n, \Delta)$. This completes the proof of Lemma 5.2.2. □

We now conclude this section by presenting the proof of the main result of this section, i.e., Theorem 5.2.1, which is mainly based on Lemma 5.2.2.

Proof of Theorem 5.2.1. Fix a sufficiently large n and $\Delta \in [20 \log n, n^{0.1}]$, and let $\mathsf{H}(n, \Delta)$ be the network proven to exist by Lemma 5.2.2. For the sake of contradiction, suppose that there exists an algorithm $\mathcal{A}$ such that $\mathcal{A}$ has acknowledgment bound of at most $\frac{\Delta \log n}{100}$ in $\mathsf{H}(n, \Delta)$. Fix one such algorithm $\mathcal{A}$ and consider the executions of $\mathcal{A}$ on $\mathsf{H}(n, \Delta)$.

For each sender process $proc(v)$, let $\mathcal{E}'_v$, let $\mathcal{E}_v$ be the event that at the end of round $\frac{\Delta \log n}{100}$, we have $ack_i = False$, where $i = id(v)$. By the assumption that $\mathcal{A}$ has acknowledgment bound of at most $\frac{\Delta \log n}{100}$ in $\mathsf{H}(n, \Delta)$ (refer to Section 4.2), we get that $\Pr[\mathcal{E}_v] \leq \frac{1}{n}$.

Moreover, for each sender process $proc(v)$, let $\mathcal{E}'_v$ be the event that at the end of round $\frac{\Delta \log n}{100}$, we have $ack_i = True$, where $i = id(v)$, but there exists a receiver neighbor u of v such that u has not received the message of v by the end of round $\frac{\Delta \log n}{100}$. Following the property (B) in the definition of a local broadcast algorithm (refer to Section 4.1), we get that for each sender process $proc(v)$, $\Pr[\mathcal{E}'_v] \leq \frac{1}{n}$.

Therefore, using a union bound, we get

$$\Pr[\big(\cup_{sender\ v}\mathcal{E}_v\big)\cup\big(\cup_{sender\ v}\mathcal{E}'_v\big)] \leq \sum_{sender\ v} \Pr[\mathcal{E}_v] + \sum_{sender\ v} \Pr[\mathcal{E}'_v] \leq \sum_{sender\ v} \frac{1}{n} + \sum_{sender\ v} \frac{1}{n} = \frac{2n^{0.12}}{n},$$

where the equality follows because the number of senders of $\mathsf{H}(n, \Delta)$ is $n^{0.12}$. This shows that with probability at least $1 - \frac{2n^{0.12}}{n} > 0$, none of the events $\mathcal{E}_v$ or $\mathcal{E}'_v$ (for each sender v) happen. Thus, with probability at least $1 - \frac{2n^{0.12}}{n} > 0$, we have that by the end of round $\frac{\Delta \log n}{100}$, each receiver has received the message of each of its neighboring senders. This shows that there exists an execution of $\mathcal{A}$ in $\mathsf{H}(n, \Delta)$ during which by the end of round $\frac{\Delta \log n}{100}$, each receiver has received the message of each of its neighboring senders. Let α be such an execution and let σ be the transmission schedule of the first $\frac{\Delta \log n}{100}$ rounds of α.

By the choice of α and thus σ, we get that when we run short transmission schedule σ in $\mathsf{H}(n, \Delta)$, each receiver receives the message of each of its neighboring senders. That is, short transmission schedule σ covers $\mathsf{H}(n, \Delta)$. This is in contradiction with Lemma 5.2.2. From this contradiction, we conclude that no algorithm $\mathcal{A}$ such that $\mathcal{A}$ has acknowledgment bound of at most $\frac{\Delta \log n}{100}$ in $\mathsf{H}(n, \Delta)$ exists. This completes the proof. □

Chapter 6

Lower Bounds in the Dual Graph Model

In this section, we present our lower bounds for the single-shot setting of the dual graph model. We show a lower bound of $\Omega(\Delta' \log n)$ on the progress time of the local broadcast algorithms in this setting. Moreover, this progress lower bound directly yields a lower bound with the same value on the acknowledgment time in the single-shot setting of the dual graph model. We emphasize that all these lower bounds are presented for centralized algorithms and also, in the model where processes are provided with a collision detection mechanism. Note that these points only strengthen these results.

We explain in Chapter 11 that since the single-shot setting can be viewed as a special case of the multi-shot setting, these lower bounds also extend to the multi-shot setting. These extensions show that the optimized decay protocol for the general multi-shot setting, which we present in Chapter 10, has optimal acknowledgment and progress bounds in the dual graph model.

6.1 Progress Time Lower Bound

Recall that in Theorem 5.2.1 of Section 5.2, we proved a lower bound of $\Omega(\Delta \log n)$ for the acknowledgment time in the classical radio broadcast model. As a core part of the proof of Theorem 5.2.1, we showed in Lemma 5.2.2 that for any sufficiently large n and any $\Delta \in [20 \log n, n^{0.1}]$, there exists a single-shot setting with a bipartite network $\mathsf{H}(n, \Delta)$ with size n and maximum receiver degree at most Δ, for which there does not exist a transmission schedule σ such that $L(\sigma) \leq \frac{\Delta \log n}{100}$ and σ covers $\mathsf{H}(n, \Delta)$. In this section, we use Lemma 5.2.2 to show a lower bound of $\Omega(\Delta' \log n)$ on the progress time in the dual graph model, where Δ' is the maximum receiver degree in graph G'. For that purpose, we first present some definitions.

For each algorithm A and each network (G, G') in the single-shot setting of the dual graph model (for formal definitions of this setting, refer to Chapters 3 and 4), we say that an execution α of A is *progressive* in (G, G') if during this execution, every receiver receives at least one message.

Now we are ready to see the main result of this section.

Theorem 6.1.1. *For any sufficiently large n and each $\Delta' \in [20 \log n, n^{\frac{1}{11}}]$, there exists a multi-shot setting in the dual graph model with a bipartite network $H^*(n, \Delta')$ of size n and maximum contention of at most Δ', such that the progress bound of any algorithm in $H^*(n, \Delta')$ is greater than $\frac{\Delta' \log n}{120}$ rounds.*

Proof Outline. In order to prove this lower bound, in Lemma 6.1.2, we show a reduction from the acknowledgment bound in the bipartite networks of the classical model to the progress bound in the bipartite networks of the dual graph model. In particular, this means that if there exists an algorithm that has progress bound at most $\frac{\Delta'_1 \log n_1}{120}$ in any single-shot setting of the dual graph model with size n_1 and maximum receiver G'-degree Δ'_1, then for any bipartite network H in the classical model with size n_2 and maximum receiver degree at most Δ_2, there exists a transmission schedule $\sigma(H)$ with length smaller than $\frac{\Delta_2 \log n_2}{100}$ that covers H. Then, we use Lemma 5.2.2 to complete the lower bound. □

Lemma 6.1.2. *Consider arbitrary n_2 and Δ_2 and let $n_1 = n_2(\Delta_2 + 1)$ and $\Delta'_1 = \Delta_2$. Suppose that in the dual graph model, for each bipartite network with n_1 nodes and maximum receiver G'-degree at most Δ'_1, there exists a local broadcast algorithm A with progress bound of at most $f(n_1, \Delta'_1)$. Then, for each bipartite network H with n_2 nodes and maximum receiver degree at most Δ_2 in the classical radio broadcast model, there exists a transmission schedule σ_H with length at most $f(n_2(\Delta_2 + 1), \Delta_2)$ that covers H.*

Proof. Consider an arbitrary n_2 and Δ_2 and let $n_1 = n_2(\Delta_2+1)$ and $\Delta'_1 = \Delta_2$. Suppose that in the dual graph model and for each bipartite network with n_1 nodes and maximum receiver G'-degree at most Δ'_1, there exists a local broadcast algorithm A for this network with progress bound of at most $f(n_1, \Delta'_1)$. Let H be an arbitrary network in the classical radio broadcast model with n_2 nodes and maximum receiver degree at most Δ_2. We show a transmission schedule σ_H of length at most $f(n_2(\Delta_2 + 1), \Delta_2)$ that covers H.

For this, using network H, we first construct a special bipartite network *Dual(H)* = (G, G') in the dual graph model that has n_1 nodes and maximum receiver G'-degree at most Δ'_1. Then, by the above assumption, we know that there exists a local broadcast algorithm A for this network with progress bound at most $f(n_1, \Delta'_1) = f(n_2(\Delta_2 + 1), \Delta_2)$ rounds. We define transmission schedule σ_H by emulating what algorithm A does in the network Dual(H) and in the presence of a special adversary. Then, we argue that σ_H covers H.

The network Dual(H) in the dual graph model is constructed as follows: The set of sender nodes in the Dual(H) is exactly the same as those in H. Let η be the number of senders of H and let m be the number of edges of H. We first add $n_1 - \eta - m$ isolated receivers in Dual(H).

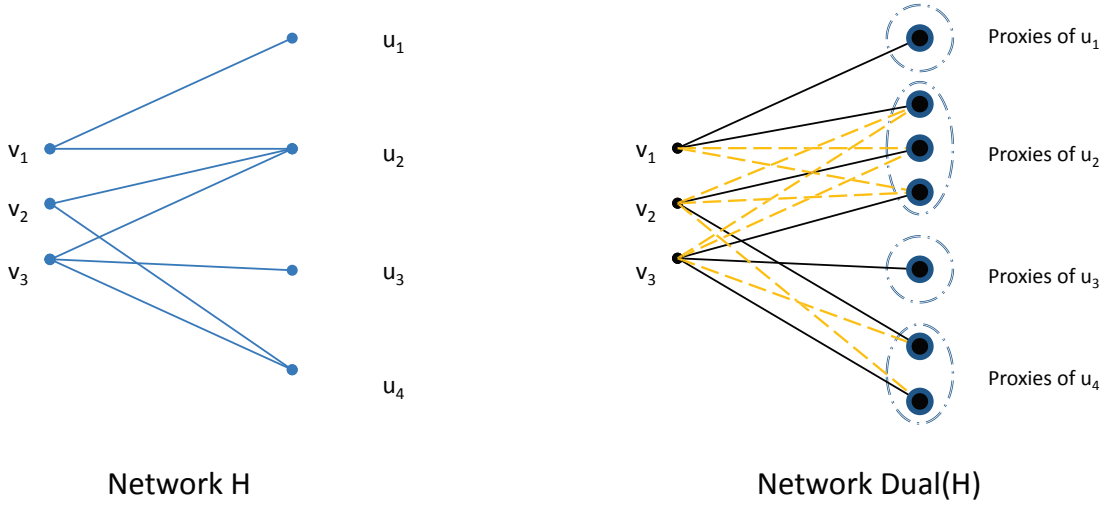

Figure 6-1: An example of the construction of the network Dual(H)

These receivers are isolated in Dual(H), i.e., there is no G'-edge incident on any of these isolated receivers. Now for each receiver u of H, let $d_H(u)$ be the degree of node u in graph H. Also, let us call the senders that are adjacent to u the *associates of* u. In the network Dual(H), we replace receiver u with $d_H(u)$ receivers and we call these new receivers the *proxies of* u. In graph G of Dual(H), we match proxies of u with associates of u, i.e., we connect each proxy to exactly one associate and vice versa. These connections are the only ones in G. In graph G' of Dual(H), we connect all proxies of u to all associates of u. These connections are the only ones in G'.

Figure 6-1 presents an example of the construction of network Dual(H), without showing the isolated receivers. The left side shows the network H and the right side shows the network Dual(H). On the right side, the black lines show the reliable edges E and the orange dashed lines show the unreliable edges $E' - E$.

Note that because of the construction, we have that the maximum degree of the receivers in G' is at most Δ_2. Also, since each receiver u is replaced by $d_H(u)$ proxies (new receiver nodes), the total number of proxies is $\sum_{\textit{receiver } u} d_H(u) = m$. Moreover, the number of senders is η and the number of isolated receivers is $n_1 - \eta - m$. Thus, the total number of nodes in Dual(H) is exactly $\eta + m + (n_1 - \eta - m) = n_1$.

We remark here that clearly the local broadcast problem does not impose any requirement for delivering any message to any of the isolated receiver nodes. These isolated receivers are used only for making the number of nodes of Dual(H) exactly n_1.

Now, we present our special adversary. In particular, we present a set of deterministic rules for determining the reach set (the set of active edges) of each round in the Dual(H) network: For each round r and each receiver node w of Dual(H),

1. if exactly one G'-neighbor of w transmits, then the adversary activates only the edges between w and its G-neighbors,

2. otherwise, the adversary activates all the edges between w and its G'-neighbors.

Intuitively, these rules try to make the number of successful message deliveries as small as possible. In particular, it is easy to see that each receiver can receive messages only from its G-neighbors. We use this fact in the rest of the proof.

Now we focus on the executions of algorithm A in the Dual(H) network and in the presence of the aforementioned special adversary and consider the space of these executions. For each receiver u in Dual(H), let $\mathcal{E}_u$ be the event that receiver u does not receive a message by the end of round $f(n_2(\Delta_2+1),\Delta_2)$. By the assumption that A has progress bound $f(n_2(\Delta_2+1),\Delta_2)$ for network Dual(H), we get that $\Pr[\mathcal{E}] \leq \frac{1}{n_1}$. Using a union bound we get that $\Pr[\cup_{receiver\ u}\mathcal{E}_u] \leq \sum_{\cup_{receiver\ u}} \Pr[\mathcal{E}_u] \leq \sum_{\cup_{receiver\ u}} \frac{1}{n_1} < 1$, where the last strict inequality follows because Dual(H) has and thus, that strictly less at least one sender n_1 receivers. Since $\Pr[\cup_{receiver\ u}\mathcal{E}_u] < 1$, there is a positive probability that none of events $\mathcal{E}_u$ happens. Thus, there exists at least one progressive execution α of A with length at most $f(n_2(\Delta_2+1),\Delta_2)$ rounds, in the presence of the special adversary adversary. Let transmission schedule σ_H be the transmission schedule of execution α. In order to complete the proof, we show that in the classical model σ_H covers H. Consider an arbitrary receiver node u in H and let v be an arbitrary sender neighbor of u in H. We show that when we run σ_H on H, u receives the message of v.

We know that in the network Dual(H), there exists a receiver w that is a proxy of u such that in graph G of Dual(H), w is matched to v. Since α is progressive, in α, w receives at least one message. On the other hand, because of the rules used by our special adversary, in execution α, w can receive messages only from v (its only G-neighbor). Thus, there exists a round r such that w receives the message of v in round r of α. Now, note that because of the second rule used by the special adversary, if in round r of α, receiver w receives the message of a node v, that means that no other G'-neighbor of w transmits in round r of α. Thus, no sender neighbor of u in graph H transmits in round r of transmission schedule σ_H. Hence, using transmission schedule σ_H in the classical radio broadcast model in graph H, node u receives the message of sender v in round r of σ_H. Therefore, using transmission schedule σ_H in the classical broadcast model and in network H, u receives the message of v. □

Proof of Theorem 6.1.1. The proof follows from Lemma 5.2.2 and Lemma 6.1.2. Fix an arbitrary sufficiently large n_1 and $\Delta_1' \in [20\log n_1, n_1^{\frac{1}{11}}]$. Note that for a sufficiently large n_1, we have $20\log n_1 \leq {n_1}^{\frac{1}{11}}$. Let $n_2 = \frac{n_1}{\Delta_1'+1}$ and $\Delta_2 = \Delta_1'$. By Lemma 5.2.2, we know that in the classical radio broadcast model, there exists a bipartite network $H(n_2,\Delta_2)$ with n_2 nodes and maximum

receiver degree at most Δ_2 such that no transmission schedule with length at most $\frac{\Delta_2 \log n_2}{100}$ rounds covers $H(n_2, \Delta_2)$. In Lemma 6.1.2, set

$$f(n_1, \Delta_1') = \frac{\Delta_1' \log\left(n_1/(\Delta_1'+1)\right)}{100} \geq \frac{\Delta_1' \log\left(n_1/(n_1^{\frac{1}{11}}+1)\right)}{100} > \frac{\Delta_1' \log\left(n_1^{10/12}\right)}{100} = \frac{\Delta_1' \log n_1}{120}$$

We can conclude that there exists a bipartite network with n_1 nodes and maximum receiver G'-degree at most Δ_1' such that there does not exist a local broadcast algorithm with progress bound at most $f(n_1, \Delta_1')$ in this network. Calling this network $H^*(n_1, \Delta_1')$ finishes the proof of Theorem 6.1.1. □

6.2 Acknowledgment Time Lower Bound

In this section, we mention that as a simple corollary of the progress time lower bound Theorem 6.1.1, we get an acknowledgment lower bound for the single-shot setting in the dual graph model.

Theorem 6.2.1. *For any sufficiently large n and each $\Delta' \in [20 \log n, n^{\frac{1}{11}}]$, there exists a single-shot setting in the dual graph model with a bipartite network $H^*(n, \Delta')$ of size n and maximum contention of at most Δ', such that the acknowledgment bound of any algorithm in $H^*(n, \Delta')$ is greater than $\frac{\Delta' \log n}{120}$ rounds.*

Proof. The proof follows directly from Lemma 4.2.1 and Theorem 6.1.1. Fix an arbitrary sufficiently large n and $\Delta' \in [20 \log n, n^{\frac{1}{11}}]$. Note that for a sufficiently large n, we have $20 \log n \leq n^{\frac{1}{11}}$. Let $H^*(n, \Delta')$ be the single-shot setting proven to exist in Theorem 6.1.1. Consider an arbitrary local broadcast algorithm $\mathcal{A}$. If $\mathcal{A}$ has an acknowledgment bound at most $\frac{\Delta' \log n}{120}$ in $H^*(n, \Delta')$, then using Lemma 4.2.1, we get that $\mathcal{A}$ also has a progress bound at most $\frac{\Delta' \log n}{120}$ in $H^*(n, \Delta')$. However, from Theorem 6.1.1, we know that no such local broadcast algorithm exists. Hence, the acknowledgment bound of $\mathcal{A}$ is greater than $\frac{\Delta' \log n}{120}$ rounds. □

Part II

The General Multi-Shot Setting

Chapter 7

The Models in The Multi-Shot Setting

In this chapter, we present the definitions of the models for the multi-shot setting. Similar to the single-shot setting (refer to Chapter 3), we use two models, namely the *classical radio network model* (also known as the radio network model) and the *dual graph model*. The main difference between the models in the multi-shot setting and the models in the single-shot setting is that, in the multi-shot setting, the system also includes an environment automaton. Intuitively, the environment abstracts the higher layers of the wireless nodes which solve higher level problems and for that, they interact with the local broadcast module.

Similar to Chapter 3, and since the classical model is a special case of the dual graph model, we use the dual graph model for defining the models in the multi-shot setting, and also when describing the problem statement in the next chapter. However, in some places, we indicate how a certain result or property changes when we focus on the special case of the classical radio network model.

In the multi-shot dual graph model, a distributed system is composed of a set of processes that are connected to each other via a network and that also interact with an environment. In the particular case of the local broadcast problem, intuitively, the processes abstract the local broadcast modules of the wireless nodes and the environment abstracts the higher layers of these wireless nodes. In Section 7.1, we present the network connections assumptions for the dual graph model. We explain the environment in Section 7.2. In Section 7.3, we explain what a process in this model is. In Section 7.4, we explain how a distributed system works as a whole, i.e., what are the executions of an algorithm in this model. For out lower bounds, we use a stronger model, *centralized setting*, which we explain in Section 7.5. We remark here that Sections 7.1 and 7.5 are exactly the same as their counterparts in single-shot setting (Sections 3.1 and 3.4, respectively), but Section 7.2 is new (i.e., does not have a single-shot counterpart) and Sections 7.3 and 7.4 are different than their counterparts in the single-shot setting (Sections 3.2 and 3.3, respectively).

7.1 Network Connections

In the dual graph model, radio networks have some reliable links and potentially some unreliable links. We define a network (G, G') to consist of two undirected finite graphs, $G = (V, E)$ and $G' = (V, E')$, where we have $E \subseteq E'$. Intuitively, set E is the set of reliable edges while E' is the set of all edges (both reliable and unreliable). We assume that the communications on the unreliable edges, i.e., the edges in set $E' \setminus E$, are controlled by an adversary. We explain this issue in more detail in Section 7.4. When restricting attention to the special case of the classical radio network model, there are no unreliable edges and thus, we simply have $G = G'$, i.e., $E = E'$. We define the size of the network to be $n = |V|$. Finally, graphs G and G' can be disconnected.

We assign processes to graph nodes of the network (G, G'). This assignment is defined by an injective function $id()$ from V to the set of process ids $[N]$. (We define processes in Section 3.2.) That is, for each $v \in V$, $id(v)$ indicates the id of the process assigned to graph node v. For each $v \in V$, we use notation $proc(v)$ to indicate the process with id $id(v)$, i.e., the process that is assigned to graph node v. We sometimes abuse notation by using the notation *process* u, or sometimes just u, for some graph node $u \in V$, to refer to $proc(u)$. We refer to a network (G, G') and a mapping $id()$ from graph nodes to processes as a *setting*.

Processes can potentially know the network (G, G') or have some partial knowledge about it[1]. This means that processes could have a full description of the network or some information about it built into their states, and the related distributed algorithm is required to work only if this full description matches the description of the network or if this partial information is consistent with the description of the network. To strengthen our results, we remark that our upper bounds make no assumptions about (G, G') other than knowing a polynomial bound on the size n of the network and a constant-factor bound on the maximum contention (explained in Chapter 8), while our lower bounds allow full knowledge of the network graphs (G, G').

7.2 The Environment

In this section, we present the definition of an environment. Informally, the environment is a probabilistic automaton that interacts with the processes assigned to graph nodes through a set of input and output actions. Recall that we explained the process mapping to graph nodes, described by function $id()$, in Section 7.1.

For consistency, throughout the thesis, we classify actions as inputs or outputs from the viewpoint of processes. That is, an *input action* is an action that the environment performs to pass some

[1]This is because, in many real world settings, it is reasonable to assume that devices can make some assumptions or inference about the structure of the network.

information to the processes and an *output action* is an action that a process performs to pass some information to the environment.

We use notations $\mathcal{I}$ and $\mathcal{O}$ to denote the set of input actions and the set of output actions, respectively. Moreover, we assume that set $\mathcal{I}$ is partitioned into sets $\mathcal{I}_i$, one for each process i at graph node v where $i = id(v)$. Similarly, we assume that set $\mathcal{O}$ is partitioned into sets $\mathcal{O}_i$, one for each process i at graph node v where $i = id(v)$. The sets $\mathcal{I}_i$ and $\mathcal{O}_i$ respectively represent the sets of inputs and outputs at the process with id i at graph node v where $i = id(v)$.

Informally, in the environment automaton we have only one type of transition: probabilistic transitions that take a state and a sequence of outputs for the process at each graph node to another state and a sequence of inputs for the process at each graph node. These transitions occur at the start of the rounds. In the first transition at the start of round 1, the environment starts in an initial state and the related output sequences are $\emptyset$.

More formally, for each network (G, G') and a mapping $id()$ from graph nodes to processes, an environment Env is a 3-tuple $(\mathcal{T}, T_0, \mathcal{H})$ and we have:

- $\mathcal{T}$ is a set of states.
- $T_0 \in \mathcal{T}$ is a single starting state
- $\mathcal{H}$ is a function that captures the probabilistic transitions of the environment. Consider an arbitrary state $S \in \mathcal{T}$. Also, consider a mapping $O()$ from processes at graph nodes to output sequences. That is, for each process i such that $i = id(v)$ for a graph node v, $O(i)$ is an output sequence[2] in $\Sigma_{\mathcal{O}_i}$. Then, function $\mathcal{H}(S, O)$ is a probability distribution function over the space of all states in $\mathcal{T}$ and all mappings from processes at graph nodes to input sequences. More precisely, we have: Consider an arbitrary state $S' \in \mathcal{T}$. Also, consider a mapping $I()$ from processes at graph nodes to output sequences. That is, for each process i such that $i = id(v)$ for a graph node v, $I(i)$ is an input sequence in $\Sigma_{\mathcal{I}_i}$. Then, $\mathcal{H}(S, O)(S', I)$ is the probability that, given that the environment Env is in state S and for each graph node v, Env receives the output sequence $O(i)$ from the process i at node v (i.e., $i = id(v)$), the environment Env makes a transition to state S' and for each graph node v, Env performs the inputs in sequence $I(i)$ for process i at graph v.

Informally, the third bullet above explains that $\mathcal{H}(S, O)(S', I)$ is the probability that, when the environment Env is in state S and receives output sequences determined by $O()$ from processes ($O()$ determines one sequence of outputs for the process at each graph node), it makes transition to state S' and generates input sequences determined by $I()$ to be passed to processes ($I()$ determines one sequence of inputs for the process at each graph node).

[2]Recall from Section 2.1 that we use notation Σ_S to denote the set of finite length sequences over set S.

7.3 Distributed Algorithms and Processes

The main difference in the definitions of this section with those of Section 3.2 is the introduction of input and output actions, which are the actions in the interface between the processes and the environment.

We define a distributed algorithm $\mathcal{A}$ to be a collection of N randomized processes, where N is an arbitrary positive integer. Processes are described by probabilistic automata and informally, in each process, we have two types of transitions: (1) probabilistic transitions that take the state at the beginning of a round and a sequence of inputs to an intermediate state and a message to be transmitted on the channel (or silence), and (2) deterministic transitions that take an intermediate state and a message received from the channel (or a special value $\perp$ or $\top$) to the state at the end of a round and a sequence of outputs, which are returned to the environment. We remark that at the start of each round, the processes make their probabilistic transitions after the environment has made its transition. We explain this issue in more details in Section 7.4. The specific definition and meaning of these actions in the case of the local broadcast problem are presented in Chapter 8.

The formal definition of processes is as follows: Each process in $\mathcal{A}$ is a 6-tuple $(i, \mathcal{Q}^i, \mathcal{P}^i, Q_0^i, \mathcal{F}^i, \mathcal{G}^i)$ and we have:

- $i \in [N]$ is the unique identifier of this process.
- $\mathcal{Q}^i$ and $\mathcal{P}^i$ are two sets of states, and we have $\mathcal{Q}^i \cap \mathcal{P}^i = \emptyset$. We refer to states in these two sets respectively as $\mathcal{Q}$-states and $\mathcal{P}$-states.
- $Q_0^i \in \mathcal{Q}^i$ is a single starting state.
- $\mathcal{F}^i$ is a function that captures the probabilistic transitions of the process. For each state $S \in \mathcal{Q}^i$ and each input sequence $I \in \Sigma_{\mathcal{I}_i}$, function $\mathcal{F}^i(S, I) : \mathcal{P}^i \times \mathcal{M}_\perp \to [0, 1]$ is a probability distribution function over $\mathcal{P}^i \times \mathcal{M}_\perp$. That is, for each state $S' \in \mathcal{Q}^i$, and each $m \in \mathcal{M}_\perp$, $\mathcal{F}^i(S, I)(S', m)$ is the probability that, given that the process with id i is in state S and it receives the input sequence I, it makes a transition to state S' and transmits m if $m \neq \perp$ and remains silent if $m = \perp$. Note that as mentioned in Section 2.1, in this notation, symbol $\perp$ means remaining silent.
- $\mathcal{G}^i : \mathcal{P}^i \times \mathcal{M}_{\perp\top} \to \mathcal{Q}^i \times \Sigma_{\mathcal{O}_i}$ is a function that captures the deterministic transitions of the process. For each state $S' \in \mathcal{P}^i$ and each $m' \in \mathcal{M}_{\top\perp}$, $\mathcal{G}^i(S', m')$ determines a state in $Q \in \mathcal{Q}^i$ and an output sequence $O \in \Sigma_{\mathcal{O}_i}$, and we have the following: if the process with id i is in $\mathcal{P}$-state S' and it receives m' from the channel, then it makes a transition to state Q and performs output sequence O.

For simplicity, we sometimes use *process* i to refer to the process with id i.

We emphasize that the main difference in the definition of the processes in this section with that of Section 3.2 is in including input and output actions. These changes particularly appear in the last two bullet-points of the definition of processes presented above.

7.4 Executions in the Dual Graph Model

As mentioned at the start of this chapter, in a distributed system in the multi-shot setting of the dual graph model, a set of processes are connected to each other via a network, where the communications on some of connections are controlled by an adversary, and the processes interact with an environment automaton. In this section, we explain how this system works as a whole by describing the executions of a distributed algorithm in this model. An execution of an algorithm $\mathcal{A}$ in network (G, G') proceeds as follows:

The execution proceeds in synchronous lock-step rounds $1, 2, ...$, where all the participating processes start in the first round.

At the beginning of each round r, the environment is in a state in $\mathcal{T}$. Then, the environment makes its probabilistic state transition using function $\mathcal{H}$ and performs some input actions. If $r = 1$, then the transition of the environment depends only on its initial state T_0. If $r \geq 2$, then the transition of the environment depends on its state at the end of round $r - 1$ (equivalently right before the start of round r) and the outputs that the environment receives at the end of round $r - 1$. The outputs that the environment receives at the end of round $r - 1$ are described by a mapping $O^{r-1}()$ that determines an output sequence $O^{r-1}(i) \in \Sigma_{\mathcal{O}_i}$ for each process i assigned to graph node v where $i = id(v)$. The inputs that the environment generates at the start of round r are described by a mapping $I^r()$ that determines an input sequence $I^r(i) \in \Sigma_{\mathcal{I}_i}$ for each process i assigned to graph node v where $i = id(v)$.

On the other hand, at the beginning of each round r, every process $proc(v), v \in V$, with $i = id(v)$ is in a $\mathcal{Q}$-state. As a result of the input actions that the environment performs during its state transition, each process $proc(v)$, , with $i = id(v)$, receives an input sequence $I^r(i) \in \Sigma_{\mathcal{I}_i}$ from the environment. Then, using function $\mathcal{F}^i$ where $i = id(v)$, $proc(v)$ performs its probabilistic state transition to a P-state, and it also determines what message it transmits in round r or that it remains silent in this round, by determining $m \in M_\perp$. Recall from Section 7.3 that in our notation, during this transition, choosing $m = \perp$ means that process i remains silent. We moreover emphasize that at the start of each round, the processes make their probabilistic transitions after the environment has made its probabilistic transition.

Next, the adversary chooses a *reach set* that consists of E and a subset, potentially empty, of edges in $E' - E$. This reach set describes the links that are active in this round. We emphasize that

when focusing on the special case of the classical model, set $E' - E$ is empty and therefore, the reach set is just E.

In the dual graph model, we assume that the adversary is *'adaptive offline'* [8] meaning that it has full knowledge about the processes. This means that when choosing the reach set of round r, the adversary knows everything about the processes up to round r of the execution including the input and output actions that processes have received or performed and the outcome of the random coins used for the transitions of all rounds up to and including round r. Most importantly, the adversary knows which processes are transmitting in round r. Moreover, we assume that the adversary can also make randomized decisions.

After the adversary determines the reach set of round r, depending on this reach set and which processes are transmitting, each process i receives exactly one value in set $\mathcal{M}_{\perp\top}$ from the channel. For a graph node v, let $B_{v,r}$ be the set of all graph nodes u such that $proc(u)$ transmits in r and edge $e = \{u, v\}$ is in the reach set for this round. What $proc(v)$ receives in round r is determined by the following rules:

(A) If $proc(v)$ broadcasts in round r, then it receives only its own message.

(B) If $proc(v)$ does not broadcast, and $|B_{v,r}| = 0$ or $|B_{v,r}| > 1$, then $proc(v)$ receives $\perp$ (indicating *silence*).

(C) If $proc(v)$ does not broadcast, and $|B_{v,r}| = 1$, then $proc(v)$ receives the message sent by $proc(u)$, where u is the single node in $B_{v,r}$.

The rule (A) means that each process cannot send and receive simultaneously. We remark that the rule (B) means that we do not assume any *collision detection* mechanism in this model. To strengthen our results, we present our lower bounds (Chapter 11) in the stronger model with collision detection, i.e., when the rule (B) is replaced with the following rule

(B') If $proc(v)$ does not broadcast, and $|B_{v,r}| = 0$, then $proc(v)$ receives $\perp$. If $proc(v)$ does not broadcast, and $|B_{v,r}| > 1$, then $proc(v)$ receives $\top$ (indicating *collision*).

After receiving the messages of round r, every process $proc(v), v \in V$ makes its deterministic transition using function $\mathcal{G}^i$, where $i = id(v)$. That is, suppose process $proc(v), v \in V$, is in a $\mathcal{P}$-state $S' \in \mathcal{P}^i$, where $i = id(v)$, and it receives $m' \in M_{\perp\top}$ from the channel. $\mathcal{G}^i(S', m')$ determines a state in $S'' \in \mathcal{Q}^i$ and an output sequence $O^r(i) \in \Sigma_{\mathcal{O}_i}$. Then $proc(v)$ makes a transition to $\mathcal{Q}$-state S'' and performs output sequence $O^r(i)$.

The main difference in this section with respect to Section 3.3 is that here, the system also includes an environment and this environment interacts with the processes through input and output actions. Thus, the execution includes the state transitions of the environment, which happen at the

start of each round, the probabilistic transitions of the processes, which are done after the transition of the environment in that round and depend on the inputs that they receive from the environment, and the deterministic transitions of the processes, which determine the outputs that they generate.

7.5 Centralized Algorithms

In our lower bounds (Chapter 11), we consider the stronger model of *centralized* algorithms. We define a centralized algorithm to be the same as the distributed algorithms explained in this chapter, with two modifications: (1) the processes know the graph (G, G') and the mapping $id()$ from the beginning of the execution; and (2) when the processes are making their transitions, they know the full history of the execution and thus, their transitions are a function of the full history of the execution. This history in particular includes the current state of all the processes in the network.

Chapter 8

The Local Broadcast Problem in The Multi-Shot Setting

In this chapter, we present the definition of the local broadcast problem in the multi-shot setting. As stated in the introduction, the informal definition of the local broadcast problem in the multi-shot setting is that processes are given messages, one by one, and they should deliver these messages to all of their neighbors. In this chapter, we formalize this definition. We remark that this problem was first introduced by Kuhn, Lynch, and Newport [31, 32] and later generalized to a probabilistic version by Khabbazian, Kowalski, Kuhn, and Lynch [36, 37]. These papers referred to the local broadcast problem as *abstract MAC*. Our definition is similar to that of [36] except that we consider synchronous rounds.

We start by explaining the interface between each process and the environment, in Section 8.1. In Section 8.4, we define the local broadcast problem. Section 8.5 presents the time bounds that we use to measure the performance of a local broadcast algorithm.

8.1 The Interface Between The Processes and The Environment

The first step in formalizing the problem statement is to fix the input/output interface between each process assigned to a graph node — each process abstracts the *local broadcast module* of a wireless node — and the environment — which abstracts the higher levels all wireless node. As explained in Section 7.2, the sets of input actions I and output actions O are partitioned into sets respectively $\mathcal{I}_i$ and $\mathcal{O}_i$, one for each process $proc(v)$ at graph node v, where $i = id(v)$. The purpose of this section is to explain what the actions in sets $\mathcal{I}_i$ and $\mathcal{O}_i$ are. The interface for process $proc(v)$, with $i = id(v)$, consists of three kinds of actions, seen from the viewpoint of $proc(v)$, as follows:

1. $bcast(m)_i$, an input action that provides the process $proc(v)$ with a message $m \in \mathcal{M}$ that has to be broadcast to all nodes in $\mathcal{N}_G(v)$, i.e., node v's reliable neighborhood.

2. $ack(m)_i$, an output action that $proc(v)$ performs to inform the higher layers that the message $m \in \mathcal{M}$ has been delivered to all reliable neighbors of node v successfully. This delivery guarantee is only probabilistic and is made precise in Section 8.4.

3. $rcv(m)_i$, an output action that $proc(v)$ performs to transfer the message $m \in \mathcal{M}$, received through the radio channel, to the higher layers.

We remark that in all these three kinds of actions, the message m is in the set $\mathcal{M}$, which as presented in Section 2.1, is the set of messages used for the communications on the channel. That is, the messages in the interactions have the same type as the messages in the lower level channel communications. Note that this is a restriction on the algorithm as for instance, a message transmitted on the channel can not be just a part of a message received in a $bcast()$ input or can not combine two or more (or parts of two or more) messages receive in $bcast()$ inputs.

8.2 Constraints for the Processes

Process $proc(v)$ is allowed to perform action $rcv(m)_i$, where $i = id(v)$, at the end of a round r only if process $proc(v)$ receives message m on the channel from a process $proc(u)$ for $u \in \mathcal{N}_{G'}(v)$, in a round $r' \leq r$. Also process $proc(v)$ outputs $rcv(m)_i$ at most once.

8.3 Constraints for the Environment

We restrict the behavior of the environment to generate $bcast()$ in a *well-formed* manner. Informally, there should be a strict alternation between $bcast()_i$ actions and corresponding $ack()_i$ actions, for each process $proc(v)$ that $i = id(v)$. The formal explanation of these constraints is as follows. For every execution and every process v, with $i = id(v)$, the environment is allowed to generate a $bcast(m)_i$ at the start of a round r only if one of the following conditions is satisfied:

1. $bcast(m)_i$ is the first input to u in the execution;

2. the last input action at v was $bcast(m')_i$ and the process $proc(v)$ has already performed $ack(m')_i$ by the end of round $r - 1$.

We emphasize that there can be any number of $rcv(m'')_i$ actions between $bcast(m')_i$ and $ack(m')_i$.

We moreover require that for each process $proc(v)$ at a graph node v and each message m, the environment does not perform action $bcast(m)_i$, where $i = id(v)$, more than once.

8.4 The Local Broadcast Problem

In this section, we present the definition of the local broadcast problem. Formally, we say that an algorithm $\mathcal{A}$ *solves the local broadcast problem* if and only if, for every well-formed environment automaton Env, executions of $\mathcal{A}$ with Env satisfy the following three properties:

(A) In every execution, for every process v, with $i = id(v)$, for each $bcast(m)_i$ input, process v eventually responds with a single $ack(m)_i$ output, and these are the only ack outputs generated by v.

(B) In every execution, for each process v, with $i = id(v)$, if process v generates a $rcv(m)_i$ output at the end of round r, then there is a neighbor $u \in \mathcal{N}'_G(v)$ with $j = id(u)$, that receives a $bcast(m)_j$ input by the start of round r and does not output $ack(m)_j$ by the end of round $r - 1$.

(C) Consider an arbitrary process v and rounds r and r'. Let α be a closed execution of $\mathcal{A}$ that ends right after the probabilistic transition of the environment Env in round r and in which Env generates a $bcast(m)_i$ input, where $i = id(v)$, to process v at the start of round r. Consider the space of all executions that extend α and let $\mathcal{E}$ be the event in this space that the process v outputs $ack(m)_i$ at the end of round r'. Let $\mathcal{E}'$ be the event in the same space that each process $u \in \mathcal{N}_G(v)$, with $j = id(u)$ generates output $rcv(m)_j$ by the end of a round $r'' \leq r'$. If $\Pr(\mathcal{E}) > 0$, then $\Pr(\mathcal{E}'|\mathcal{E}) \geq 1 - \frac{1}{n}$.

Informally, property (C) states that if process $proc(v)$ acknowledges that the message m is delivered to all neighbors by outputting $ack(m)_i$, where $i = id(v)$, at the end of a round r, then with high probability, by the end of round r, all G-neighbors u of v have output $rcv(m)_j$, where $j = id(u)$.

An algorithm $\mathcal{A}$ that solves the local broadcast problem is called a *local broadcast algorithm*.

8.5 The Time Bounds

We measure the performance of a local broadcast algorithm with respect to the two bounds first formalized in [31]: the *acknowledgment bound*, and the *progress bound*. These bounds are normally considered as a functions of the local contention. Before defining these bounds, we first present the definitions that we use to describe the local contention during a given round interval. The following are defined with respect to a fixed execution.

(i) We say a process v, with id $i = id(v)$ is *active with message m in round r* iff it receives a $bcast(m)_v$ input at the start of a round $\leq r$ and does not generate a corresponding $ack(m)_v$

output by the end of a round $\leq r-1$. We furthermore call a message m *active* in round r if there is a process that is active with it in round r.

(ii) For process v and round r, the local contention $c(v,r)$ is equal to the number of active processes in $\mathcal{N}^+_{G'}(v)$ in round r.

We use notation Δ' (or Δ for the classical model) to denote an upper bound on the maximum local contention $c(v,r)$ over all processes v and all rounds r of the execution that is under consideration. In our upper bound results, we assume that processes know Δ' (or Δ for the classical model). This means that the processes have Δ' (or Δ for the classical model) encoded in their states and the algorithm is required to work if for each round r and each node u, the local contention $c(v,r) \leq \Delta'$.

Now we go back to defining the time bounds. We first present informal descriptions of these bounds: Informally, (1) the *acknowledgment bound* is a bound on the time between a $bcast(m)_i$ and the corresponding $ack(m)_i$, (2) the *progress bound* is a bound on the time for a process to receive at least one message when it has one or more G neighbors with messages to send.

The *acknowledgment* bound represents a standard way of measuring the performance of local communication. The *progress* bound is less commonly studied but is crucial for obtaining tight performance bounds in certain classes of applications, such as global broadcast. See [31, 36, 40] for examples of places where the progress bound is used explicitly. Also, [19, 23, 24, 25, 26, 29] use the progress bound implicitly throughout their analysis.

We now present the definitions of the *acknowledgment bound* and the *progress bound*. For any given local broadcast algorithm $\mathcal{A}$ and any fixed multi-shot setting (G, G') with the guranteed maximum contention Δ', these bounds are defined as follows:

1. A number $t \in \mathbb{N}$ is an *acknowledgment bound* for $\mathcal{A}$ in (G, G') iff the following holds:

   Consider an arbitrary process v, an arbitrary well-formed environment Env, and a round r. Let α be a closed execution of $\mathcal{A}$ that ends right after the probabilistic transition of the environment Env in round r and in which Env generates a $bcast(m)_i$ input, where $i = id(v)$, to process v at the start of round r. Consider the space of all executions that extend α and let $\mathcal{E}$ be the event in this space that the process v outputs $ack(m)_i$ by the end of round $r+t$. Then we have $Pr(\mathcal{E}) \geq 1 - \frac{1}{n}$.

2. A number $t \in \mathbb{N}$ is a *progress bound* for $\mathcal{A}$ in (G, G') iff the following holds: Consider an arbitrary process v, an arbitrary well-formed environment Env, and a round r. Let α be a closed execution of $\mathcal{A}$ that ends right at the end of round $r-1$. Consider the space of all executions that extend α and let $\mathcal{E}$ be the event in this space that for each round $r' \in [r, r+t]$, there exists at least one neighbor $u \in \mathcal{N}_G(v)$ that is active in round r'. Also, let $\mathcal{E}'$ be the event in the same probability space that v generates a $rcv(m)_i$ output, where $i = id(v)$, by

the end of round $r + t$ for a message m that was active in $\mathcal{N}_{G'}(v)$ in some round in $[r, r + t]$. If $\Pr(\mathcal{E}) > 0$, then $\Pr(\mathcal{E}'|\mathcal{E}) > 1 - \frac{1}{n}$.

We remark that in the above conditions, the related probability distributions are with respect to the probabilistic choices of the algorithm, the environment, and the adversary.

Chapter 9

Related Work

9.1 Single-Hop Networks

The k-selection problem is the restricted case of the local broadcast problem for single-hop networks, in the classical model. This problem is defined as follows. The network is a clique of size n, and k arbitrary processes are active with messages. The problem is for all of these active processes to deliver their messages to all the nodes in the network. This problem received a vast range of attention throughout 70's and 80's, and under different names, see *e.g.* [9]- [16]. For this problem, Tsybakov and Mikhailov [9], Capetanakis [10, 11], and Hayes [12], (independently) presented deterministic tree algorithms with time complexity of $O(k + k\log(\frac{n}{k}))$ rounds. Komlos and Greenberg [17] showed if processes know the value of k, then there exists algorithms that work with the same time complexity in networks that do not provide any collision detection mechanism. Greenberg and Winograd [16] showed a lower bound of $\Omega(\frac{k\log n}{\log k})$ for time complexity of deterministic solutions of this problem in the case of networks with collision detection.

On the other hand, Tsybakov and Mikhailov [9], and Massey [13], and Greenberg and Lander [14] present randomized algorithms that solve this problem in expected time of $O(k)$ rounds. One can see that with simple modifications, these algorithms yield high-probability randomized algorithms that have time complexity of $O(k) + \mathit{poly}\log(n)$ rounds.

9.2 Multi-Hop Networks

Chlamatac and Kutten [18] were the first to introduce the classical radio network model. Bar-Yehuda et al. [19] studied the theoretical problem of local broadcast in synchronized multi-hop radio networks as a submodule for the broader goal of global broadcast. For this, they introduced the *Decay* protocol, a randomized distributed algorithm that solves the local broadcast problem.

Since then, the Decay protocol has been the standard method for resolving contention in wireless networks (see *e.g.* [29, 31, 36, 38]). In this paper, we prove that a slightly modified version of Decay protocol achieves optimal progress and acknowledgment bounds in both the classical radio network model and the dual graph model. A summary of these time bounds is presented in Figure 1-1.

Deterministic solutions to the local broadcast problem are typically based on combinatorial objects called *Selective Families*, see *e.g.* [24]-[28]. Clementi et al. [26] construct (n, k)-selective families of size $O(k \log n)$ ([26, Theorem 1.3]) and show that this bound is tight for these selective families ([26, Theorem 1.4]). Using these selective families, one can get local broadcast algorithms that have progress bound of $O(\Delta \log n)$, in the classical model. These families do not provide any local broadcast algorithm in the dual graph model. Also, in the same paper, the authors construct (n, k)-strongly-selective families of size $O(k^2 \log n)$ ([26, Theorem 1.5]). They also show (in [26, Theorem 1.6]) that this bound is also, in principle, tight for selective families when $k \leq \sqrt{2n} - 1$. Using these strongly selective families, one can get local broadcast algorithms with acknowledgment bound of $O(\Delta^2 \log n)$ in the classical model and also, with acknowledgment bound of $O((\Delta')^2 \log n)$ in the dual graph model. As can be seen from our results (summarized in Figure 1-1) and particularly the upper bounds that we present in Theorem 10.2.1, all three of the above time bounds are far from the optimal bounds of the local broadcast problem. This shows that when randomized solutions are admissible, solutions based on these notions of selective families are not optimal.

In [27], Clementi et al. introduce a new type of selective families called Ad-Hoc Selective Families which provide new solutions for the local broadcast problem, if we assume that processes know the network and the mapping $proc()$ from graph nodes to processes. Clementi et al. show in [27, Theorem 1] that for any given collection $\mathcal{F}$ of subsets of set $[n]$, each with size in range $[\Delta_{min}, \Delta_{max}]$, there exists an ad-hoc selective family of size $O((1 + \log(\Delta_{max}/\Delta_{min})) \cdot \log |F|)$. This, under the assumption of processes knowing the network and the mapping $proc()$, translates to a deterministic local broadcast algorithm with progress bound of $O(\log \Delta \log n)$, in the classical model. This family do not yield any broadcast algorithms for the dual graph model. Also, in [28], Clementi et al. show that for any given collection $\mathcal{F}$ of subsets of set $[n]$, each of size at most Δ, there exists a Strongly-Selective version of Ad-Hoc Selective Families that has size $O(\Delta \log |F|)$ (without using the name ad hoc). This result shows that, again under the assumption of knowledge of the network and the mapping $proc()$, there exists a deterministic local broadcast algorithms with acknowledgment bounds of $O(\Delta \log n)$ and $O(\Delta' \log n)$, respectively in the classical and dual graph models. Our lower bounds for the classical model, which we present in Corollaries Corollary 11.1.6 and Corollary 11.1.7, show that both of the above upper bounds on the size of these objects are tight.

Chapter 10

The Upper Bounds in The Multi-Shot Setting

In this chapter, we present our local broadcast algorithm for the multi-shot setting of both the classical model and the dual graph model. We remark that this algorithm is achieved by a small and simple change to the Decay protocol of Bar-Yehuda, Goldreich and Itai [19]. The Decay protocol was designed as a sub-module for the global broadcast problem in the classical model.

In Chapter 11, we show our acknowledgment and progress lower bounds for the multi-shot setting, which show that this optimized variant of the Decay protocol has asymptotically optimal progress and acknowledgment bounds in both the classical and the dual graph models.

This chapter is organized as follows. In Section 10.1, we present our optimized variant of the Decay protocol. We present the analysis of this algorithm in Section 10.2.

10.1 The Optimized Decay Protocol

Since the classical model is simply a special case of the dual graph model, we use the dual graph model for explaining the algorithm.

In the optimized Decay algorithm, the rounds are divided into phases, each consisting of $2\log \Delta'$ consecutive rounds[1]. Also, the phases of different processes are synchronized with each other. When a process v receives an input $bcast(m)_i$, where $i = id(v)$, at the start of a round r, it waits till the start of the next phase, i.e., the first phase that starts in a round $r' \geq r$. Then, for $100\frac{\Delta' \log n}{\log \Delta'}$ phases, processes v tries transmitting message m, according to Algorithm 10.1 in each phase (explained in the next paragraph). At the end of the last round of the last of these phases, the process v outputs $ack(m)_i$.

[1]For simplicity, throughout this chapter, we assume that Δ' is a power of 2. Otherwise, we can use $2\lceil \log \Delta' \rceil$ as the length of each phase. It is easy to see that all the calculations remain correct up to a small constant factor.

If a process v has received input $bcast(m)_i$, where $i = id(v)$, by the start of a phase p and v has not output $ack(m)_i$ by the same time (the start of phase p), we say that in phase p, process v is *ready with message* m or simply *ready*. During each such phase p, process v runs Algorithm 10.1. In particular, each phase is as follows: in odd rounds, the process v transmits message m with exponentially decreasing probabilities $\frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{\Delta'}$. Recall that we assumed that Δ' is a power of 2. In even rounds, the process v transmits message m with probability $\frac{1}{\Delta'}$. On the other hand, if a process u listens in a round r and receives a message m in that round, and u has not output $rcv(m_j)$, where $j = id(u)$, by the end of round $r-1$, then u outputs $rcv(m_j)$ at the end of round r.

We use the odd rounds for achieving a good progress bound in the classical model, whereas all the other bounds, i.e., the acknowledgment in the classical model and the progress and acknowledgment in the dual graph model, only use the even rounds. This point is made clear in the analysis of this algorithm in Section 10.2.

Algorithm 1 A phase of the optimized Decay in process v with $i = id(v)$

```
for r := 1 to 2log Δ' do
    if r mod 2 = 1 then
        with probability 2^{-⌈r/2⌉} transmit m, otherwise listen
    else
        with probability 1/Δ' transmit m, otherwise listen
    endIf
endfor
output ack(m)_i
```

10.2 The Analysis of the Optimized Decay Protocol

In this section, we analyze the optimized Decay protocol and show that this algorithm has progress and acknowledgment bounds of, respectively, $O(\log \Delta \log n)$ and $O(\Delta \log n)$ in the classical model, and progress and acknowledgment bounds of, respectively, $O(\Delta' \log n)$ and $O(\Delta' \log n)$ in the dual graph model. Specifically, note that the dependence of the progress bound on the maximum contention, Δ or Δ', changes exponentially between the classical model and the dual graph model.

Theorem 10.2.1. *The optimized Decay algorithm solves the local broadcast problem in both the classical and the dual graph model. In the classical model, this algorithm has progress and acknowledgment bounds, respectively,* $100 \log \Delta \log n$ *and* $300 \Delta \log n$*. In the dual graph model, this algorithm has progress and acknowledgment bounds, respectively,* $600 \Delta' \log n$ *and* $300 \Delta' \log n$*.*

We break the theorem into smaller parts and prove these parts separately.

Lemma 10.2.2. *The optimized Decay algorithm solves the local broadcast problem in the dual graph model and has acknowledgment bound of* $300\Delta' \log n$.

Proof. It is clear that the optimized Decay algorithm satisfies properties (A) and (B) of local broadcast algorithms (refer to Section 8.4). We first show that the optimized Decay algorithm has acknowledgment bound of $300\Delta' \log n$. Then in order to complete the proof, we show that the optimized Decay algorithm satisfies property (C) of local broadcast algorithms (refer to Section 8.4) as well.

In order to prove that the optimized Decay algorithm has acknowledgment bound of $300\Delta' \log n$, consider a process v with id $i = id(v)$ and a round r such that v receives an input of $bcast(m)_i$ in round r. Let P the set of first $100\frac{\Delta' \log n}{\log \Delta'}$ phases that start in a round $\geq r$, i.e., the set of $100\frac{\Delta'}{\log \Delta'}$ phases during which v is ready with m. Note that process v acknowledges message m at the end of (the last round of) the last phase of P. Since each phase has $2\log \Delta'$ round, and the first phase of P starts by round $r + 2\log \Delta'$, we get that process v outputs $ack(m)_i$ by the end of round $r' = r + 200\Delta' \log n + 2\log \Delta' < r + 300\Delta' \log n$. Note that this guarantee is deterministic. This shows that the optimized Decay algorithm has acknowledgment bound $300\Delta' \log n$ (refer to Section 8.5). Note that since the optimized Decay algorithm provides a deterministic guarantee on the time till acknowledging a message, checking the definition of the acknowledgment bound in this case does not require any conditioning and is considerably simpler than the statement of the acknowledgment bound in Section 8.5.

Now, in order to conclude the proof, we show that the optimized Decay algorithm satisfies property (C) of local broadcast algorithms (refer to Section 8.4) as well. Let process v and round r be defined as above and let α be a closed execution of $\mathcal{A}$ that ends right after the probabilistic transition of the environment Env in round r and in which Env generates a $bcast(m)_i$ input, where $i = id(v)$, to process v at the start of round r. Let P be defined as above, i.e., P is the set of first $100\frac{\Delta' \log n}{\log \Delta'}$ phases that start in a round $\geq r$. Note that v outputs $ack(m)_i$ only at the end of round r' that is the last round of the last phase of P. Thus, we only need to check property (C) for this particular round r'. That is, if we consider the space of all executions that extend α, then the event $\mathcal{E}$ that v outputs $ack(m)_i$ at the end of round r' happens in every such execution and thus, conditioning on this event does not change the probabilities. Now let $\mathcal{E}'$ be the event in the space of all executions that extend α that each process $u \in \mathcal{N}_G(v)$, with $j = id(u)$ generates output $rcv(m)_j$ by the end of a round $r'' \leq r'$. We show that $\Pr(\mathcal{E}') \geq 1 - \frac{1}{n}$.

Throughout the rest of this proof, our probabilities are based on the space of all executions that extend α. Consider an arbitrary process $u \in \mathcal{N}_G(v)$. To prove that $\Pr(\mathcal{E}') \geq 1 - \frac{1}{n}$, we show that by the end of the last phase of P, with probability at least $1 - \frac{1}{n^{25}}$, u receives message m. A union bound over all nodes in $\mathcal{N}_G(v)$ then shows that $\Pr(\mathcal{E}') \geq 1 - \frac{1}{n^{24}} \geq 1 - \frac{1}{n}$ and thus completes the proof.

To show that u receives m by the end of the last phase of P with probability at least $1 - \frac{1}{n^{25}}$, we focus on the even rounds of phases in P. Note that there are at least $100\frac{\Delta' \log n}{\log \Delta'} \cdot \log \Delta' = 100\Delta' \log n$ even rounds in phases in P. For each even round r in a phase in P, the probability that u receives the message of v is at least

$$\frac{1}{\Delta'} (1 - \frac{1}{\Delta'})^{c(u,r)-1},$$

where the first term is the probability of transmission of process v and the other term is the probability that rest of the active processes in $\mathcal{N}^+_{G'}(u)$ remain silent. We moreover have

$$\frac{1}{\Delta'} (1 - \frac{1}{\Delta'})^{c(u,r)-1} \geq \frac{1}{\Delta'} (1 - \frac{1}{\Delta'})^{\Delta'-1} \geq \frac{1}{\Delta'} (1 - \frac{1}{\Delta'})^{\Delta'} \geq \frac{1}{4\Delta'}.$$

Since there are at least $100\Delta' \log n$ even round in phases of P, we get that the probability that u does not receive m in any of these rounds is at most $(1 - \frac{1}{4\Delta'})^{100\Delta' \log n} < e^{-25 \log n} < \frac{1}{n^{25}}$. Hence, with probability at least $1 - \frac{1}{n^{25}}$, u receives message m by the end of the last phase of P. A union bound over all nodes $u \in \mathcal{N}_G(v)$ shows that with probability at least $1 - \frac{1}{n^{24}}$, each G-neighbor u of v receives message m by the end of the last phase of P. That is, $\Pr(\mathcal{E}') \geq 1 - \frac{1}{n^{24}} \geq 1 - \frac{1}{n}$, which completes the proof. □

Corollary 10.2.3. *The optimized Decay solves solves the local broadcast problem in the classical model and has an acknowledgment bound of* $300\Delta \log n$.

Proof. The corollary follows directly from Lemma 10.2.2 by setting $G = G'$. □

Corollary 10.2.4. *The optimized Decay algorithm has progress bound* $600\Delta' \log n$ *in the dual graph model.*

Proof. Consider an arbitrary process v and an arbitrary round r. Let α be a closed execution of $\mathcal{A}$ that ends right at the end of round $r - 1$. Consider the space of all executions that extend α. For the rest of the proof, we focus on this probability space. Let $\mathcal{E}$ be the event in this space that for each round $r' \in [r, r + 600\Delta' \log n]$, there exists at least one neighbor $u \in \mathcal{N}_G(v)$ that is active in round r'. Also, let $\mathcal{E}'$ be the event in the same probability space that v generates a $rcv(m)_i$ output, where $i = id(v)$, by the end of round $r + 600\Delta' \log n$ for a message m that was active in $\mathcal{N}_{G'}(v)$ in some round in $[r, r + 600\Delta' \log n]$. We show that if $\Pr(\mathcal{E}) > 0$, then $\Pr(\mathcal{E}'|\mathcal{E}) > 1 - \frac{1}{n}$. Refer to Section 8.5 to see that this matches the definition of the progress bound.

Consider the executions that extend α and in which event $\mathcal{E}$ happens. Let process $proc(u)$ be a process for a node $u \in \mathcal{N}_G(v)$ that receives a $bcast(m)_j$, where $j = id(u)$, at the start of a round $r' \in [r, r + 300\Delta' \log n]$. Note that such a process $proc(u)$ exists because event $\mathcal{E}$ happens and each process is active with one message for at most $200\Delta' \log n + 2 \log \Delta' \log n \leq 250\Delta' \log n$

rounds. In other words, if no such process $proc(u)$ exists, then since each process is active with one message for at most $250\Delta' \log n$ rounds, in rounds $[r + 250\Delta' \log n + 1, r + 300\Delta' \log n]$ no G-neighbor of v would be active, which means that $\mathcal{E}$ does not happen. Since we assumed that $\mathcal{E}$ happens, we get that such a process $proc(u)$ exists.

Now we focus on process $proc(u)$ that receives a $bcast(m)_j$, where $j = id(u)$, at the start of a round $r' \in [r, r + 300\Delta' \log n]$. Moreover, we focus on rounds $[r', r' + 300\Delta' \log n]$. We show that with high probability, process v receives message m from u in a round in interval $[r', r' + 300\Delta' \log n] \subset [r, r+600\Delta' \log n]$. This proves that $\Pr(\mathcal{E}'|\mathcal{E}) \geq 1-\frac{1}{n}$, which completes the proof. The calculations of this part are similar to those in the last paragraph of the proof of Lemma 10.2.2 except for changing the names of u and v. For completeness, we repeat these calculations.

Let P be the set of first $100\frac{\Delta' \log n}{\log \Delta'}$ phases that start in a round $\geq r'$. We show that v receives m by the end of the last phase of P with probability at least $1 - \frac{1}{n^{25}}$. For this we focus on the even rounds of phases in P. Note that there are at least $100\frac{\Delta' \log n}{\log \Delta'} \cdot \log \Delta' = 100\Delta' \log n$ even rounds in phases in P. For each even round r in a phase in P, the probability that v receives the message m from v is at least

$$\frac{1}{\Delta'} (1 - \frac{1}{\Delta'})^{c(v,r)-1},$$

where the first term is the probability of transmission of process u and the other term is the probability that rest of the active processes in $\mathcal{N}^+_{G'}(v)$ remain silent. We moreover have

$$\frac{1}{\Delta'} (1 - \frac{1}{\Delta'})^{c(v,r)-1} \geq \frac{1}{\Delta'} (1 - \frac{1}{\Delta'})^{\Delta'-1} \geq \frac{1}{\Delta'} (1 - \frac{1}{\Delta'})^{\Delta'} \geq \frac{1}{4\Delta'}.$$

Since there are at least $100\Delta' \log n$ even round in phases of P, we get that the probability that v does not receive m in any of these rounds is at most $(1 - \frac{1}{4\Delta'})^{100\Delta' \log n} < e^{-25 \log n} < \frac{1}{n^{25}}$. Hence, with probability at least $1 - \frac{1}{n^{25}}$, v receives message m by the end of the last phase of P. Note that we obtained this using the assumption that event $\mathcal{E}$ happens. Thus, if in an execution extending α event $\mathcal{E}$ happens, then with probability at least $1 - \frac{1}{n^{25}}$, event $\mathcal{E}'$ happens as well. Hence, $\Pr(\mathcal{E}'|\mathcal{E}) \geq 1 - \frac{1}{n^{25}} \geq 1 - \frac{1}{n}$, which completes the proof. □

Lemma 10.2.5. *The optimized Decay algorithm has progress bound* $100 \log \Delta \log n$ *in the classical model.*

Proof. Consider an arbitrary process v and an arbitrary round r. Let α be a closed execution of $\mathcal{A}$ that ends right at the end of round $r-1$. Consider the space of all executions that extend α. For the rest of the proof, we focus on this probability space. Let $\mathcal{E}$ be the event in this space that for each round $r' \in [r, r + 100 \log \Delta \log n]$, there exists at least one neighbor $u \in \mathcal{N}_G(v)$ that is active in round r'. Also, let $\mathcal{E}'$ be the event in the same probability space that v generates a $rcv(m)_i$ output, where $i = id(v)$, by the end of round $r + 100\Delta \log n$ for a message m that was active in $\mathcal{N}_{G'}(v)$ in

some round in $[r, r + 100 \log \Delta \log n]$. We show that if $\Pr(\mathcal{E}) > 0$, then $\Pr(\mathcal{E}'|\mathcal{E}) > 1 - \frac{1}{n}$. Refer to Section 8.5 to see that this matches the definition of the progress bound.

Assume that event $\mathcal{E}$ happens. Informally, we show that there are at least $50 \log \Delta \log n$ consecutive rounds r'' in interval $[r, r+100 \log \Delta \log n]$ such that in each round r'', at least one G-neighbor of v is ready. Then we show that in these $50 \log \Delta \log n$ rounds, with high probability, v receives at least one message. This shows that $\Pr(\mathcal{E}'|\mathcal{E}) > 1 - \frac{1}{n}$ and thus completes the proof.

We divide the proof into two cases as follows: First, we assume that α is such that there exists at least one G-neighbor w of v that receives a $bcast(m)_j$, where $j = id(w)$, by the end of round $r - 1$ and w is active till the end of round $r + 50 \log \Delta \log n$. Note that since the round in which a process w outputs $ack(m)_j$, where $j = id(w)$, only depends on the round in which w receives the related $bcast(m)_j$ and this dependence is deterministic, whether α satisfies the above assumption only depends on α. In particular, whether α satisfies the above assumption does not depend on the probabilistic choices after α. We first prove the claim for the first case, where this assumption is satisfied, by focusing on rounds $[r, r + 50 \log \Delta \log n]$. Then, in the second case, we study the situation where α does not satisfy this assumption. We show that in the second case, since $\mathcal{E}$ happens, there exists at least one G-neighbor w' of v that receives a $bcast(m)_{j'}$, where $j' = id(w')$, in a round in interval $[r, r + 50 \log \Delta \log n + 1]$. Having this, we prove the claim for the second case by focusing on rounds $[r + 50 \log \Delta \log n + 1, r + 100 \log \Delta \log n]$.

First Case We assume that α is such that there exists at least one G-neighbor w of v that receives a $bcast(m)_j$, where $j = id(w)$, by the end of round $r - 1$ and w is active till the end of round $r + 50 \log \Delta \log n$. We show that with high probability, $\mathcal{E}'$ happens. Let P be the set of phases that are in round interval $[r, r + 50 \log \Delta \log n]$. Let $R(v, r)$ denote the number of processes in $\mathcal{N}_G(v)$ that are ready in round r. Consider an arbitrary phase $p \in P$. Note that since a process becomes ready only at the start of a phase and it acknowledges only at the end of a phase, for each two round r, r' of phase p, we have $R(v, r) = R(v, r')$. Let $k = R(v, r)$ for a round r of phase p. Since process w as explained above exists, we have $k = R(v, r) \geq 1$. Consider odd round $r = 2\lceil \log k \rceil - 1$ of phase p. Note that we have that $2^{\lceil \frac{r}{2} \rceil} = 2^{\lceil \log k \rceil} \in [k, 2k]$. Then, the probability that v receives a message in round r from some ready node in $\mathcal{N}_G(v)$ is at least

$$\frac{k}{2^{\lceil \frac{r}{2} \rceil}} (1 - \frac{1}{2^{\lceil \frac{r}{2} \rceil}})^{k-1} \geq \frac{k}{2k} (1 - \frac{1}{k})^{k-1} \geq \frac{k}{2k} 4^{-\frac{k-1}{k}} > \frac{1}{8}.$$

Thus, for each phase $p \in P$, the probability that v receives at least one message during phase p is at least $\frac{1}{8}$. Therefore, the probability that v does not receive a message in any of phases in P is at most $(1 - \frac{1}{8})^{40 \log n} \leq e^{-\frac{50 \log n}{8}} < e^{-6 \log n} < \frac{1}{n}$. Hence, in the first case, event $\mathcal{E}'$ happens with probability at least $1 - \frac{1}{n}$. Recall that we had assumed that $\mathcal{E}$ happens. Thus, in the first case, we

have $Pr[\mathcal{E}'|\mathcal{E}] \geq 1 - \frac{1}{n}$. This completes the proof for the first case.

Second Case Now we assume that α is such that there is no G-neighbor w of v that receives a $bcast(m)_j$, where $j = id(w)$, by the end of round $r-1$ and w is active till the end of round $r + 50 \log \Delta \log n$. Hence, by the end of round $50 \log \Delta \log n$, all the processes w at G-neighbors of v have output $ack(m)_j$, where $j = id(w)$ for the message m that they received the related $bcast(m)_j$ by the end of round $r-1$. In this case, since we assumed that $\mathcal{E}$ happens, for each round $r' \in [r + 50 \log \Delta \log n + 1, r + 100 \log \Delta \log n]$, there must be at least one neighbor $u \in \mathcal{N}_G(v)$ that is active in round r'. Because of this, we get that there exists at least one G-neighbor w' of v that receives a $bcast(m)_{j'}$, where $j' = id(w')$, in a round in interval $[r, r + 50 \log \Delta \log n + 1]$. Now the calculations of the proof are similar to those in the first case but with focus on rounds $[r + 50 \log \Delta \log n + 1, r + 100 \log \Delta \log n]$. Let P' be the set of phases that are in round interval $[r + 50 \log \Delta \log n + 1, r + 100 \log \Delta \log n]$. Let $R(v, r)$ denote the number of processes in $\mathcal{N}_G(v)$ that are ready in round r. Consider an arbitrary phase $p \in P'$. Note that since a process becomes ready only at the start of a phase and it acknowledges only at the end of a phase, for each two round r, r' of phase p, we have $R(v, r) = R(v, r')$. Let $k = R(v, r)$ for a round r of phase p. Since process w' as explained above exists, we have $k = R(v, r) \geq 1$. Consider odd round $r = 2\lceil \log k \rceil - 1$ of phase p. Note that we have that $2^{\lceil \frac{r}{2} \rceil} = 2^{\lceil \log k \rceil} \in [k, 2k]$. Then, the probability that v receives a message in round r from some ready node in $\mathcal{N}_G(v)$ is at least

$$\frac{k}{2^{\lceil \frac{r}{2} \rceil}} (1 - \frac{1}{2^{\lceil \frac{r}{2} \rceil}})^{k-1} \geq \frac{k}{2k} (1 - \frac{1}{k})^{k-1} \geq \frac{k}{2k} 4^{-\frac{k-1}{k}} > \frac{1}{8}.$$

Thus, for each phase $p \in P'$, the probability that v receives at least one message during phase p is at least $\frac{1}{8}$. Therefore, the probability that v does not receive a message in any of phases in P' is at most $(1 - \frac{1}{8})^{50 \log n} \leq e^{-\frac{50 \log n}{8}} < e^{-6 \log n} < \frac{1}{n}$. Hence, in the second case, event $\mathcal{E}'$ happens with probability at least $1 - \frac{1}{n}$. Recall that we had assumed that $\mathcal{E}$ happens. Thus, in the second case, we have $Pr[\mathcal{E}'|\mathcal{E}] \geq 1 - \frac{1}{n}$. This completes the proof for the second case.

□

Proof of Theorem 10.2.1. Proof follows directly from Lemma 10.2.2, Corollary 10.2.3, Corollary 10.2.4, and Lemma 10.2.5. □

Chapter 11

The Lower Bounds in The Multi-Shot Setting

In this chapter, we present our progress and acknowledgment lower bounds for the multi-shot setting in both the classical and the dual graph models. All these lower bounds match the respective upper bounds presented in Chapter 10.

Our lower bounds for the multi-shot setting are obtained by extending the lower bounds of the single-shot setting, presented in Theorems 5.1.1, 5.2.1, 6.1.1, 6.2.1, to the multi-shot setting. These lower bounds are presented in Corollaries 11.1.6, 11.1.7, 11.1.8, and 11.1.9, respectively.

We remark that the first two corollaries are about the classical model whereas the last two corollaries are about the dual graph model. Moreover, the first and the third corollaries are about the progress bound whereas the second and the fourth corollaries are about the acknowledgment bound.

Corollary 11.1.6. *For any sufficiently large n and any $\Delta \leq n$, there exists a multi-shot setting in the classical model with a bipartite network $\mathcal{H}(n, \Delta)$ of size n where for each node u and each round r, the contention $c(u, r) \leq \Delta$, and such that the progress bound of any algorithm in $\mathcal{H}(n, \Delta)$ is greater than $\Omega(\log \Delta \log n)$ rounds.*

Corollary 11.1.7. *For any sufficiently large n and any $\Delta \in [20 \log n, n^{0.1}]$, there exists a multi-shot setting in the classical model with a bipartite network $\mathsf{H}(n, \Delta)$ of size n where for each node u and each round r, the contention $c(u, r) \leq \Delta$, and such that the acknowledgment bound of any algorithm in $\mathsf{H}(n, \Delta)$ is greater than $\frac{\Delta \log n}{100}$ rounds.*

Corollary 11.1.8. *For any sufficiently large n and each $\Delta' \in [20 \log n, n^{\frac{1}{11}}]$, there exists a multi-shot setting in the dual graph model with a bipartite network $H^*(n, \Delta')$ of size n where for each node u and each round r, the contention $c(u, r) \leq \Delta'$, and such that the progress bound of any algorithm in $H^*(n, \Delta')$ is greater than $\frac{\Delta' \log n}{120}$ rounds.*

Corollary 11.1.9. *For any sufficiently large n and each $\Delta' \in [20 \log n, n^{\frac{1}{11}}]$, there exists a multi-shot setting in the dual graph model with a bipartite network $H^*(n, \Delta')$ of size n where for each node u and each round r, the contention $c(u, r) \leq \Delta'$, and such that the acknowledgment bound of any algorithm in $H^*(n, \Delta')$ is greater than $\frac{\Delta' \log n}{120}$ rounds.*

The proof outline of all these extensions is similar, and is as follows:

Proof outline for Corollaries 11.1.6, 11.1.7, 11.1.8, and 11.1.9: We use the same networks as in Theorems 5.1.1, 5.2.1, 6.1.1, and 6.2.1. However, we present a particular behavior of the environment in the multi-shot setting of these bipartite networks that 'simulates' the local broadcast problem of the single-shot setting. Then, the proof is by contradiction. We show that if there exists a local broadcast algorithm $\mathcal{A}$ that 'breaks' Corollary 11.1.6, 11.1.7, 11.1.8, or 11.1.9, i.e., has a smaller progress or acknowledgment bound in the related network, then there exists a local broadcast algorithm $\mathcal{B}$ in the single shot setting that, respectively, 'breaks' Theorem 5.1.1, 5.2.1, 6.1.1, or 6.2.1. □

Next, we present the proof of Corollaries 11.1.8 and 11.1.9. The proofs of the Corollaries 11.1.6 and 11.1.7 are respectively similar to the proofs of Corollaries 11.1.8 and Corollary 11.1.9, with the exception of changing the values of the parameters (e.g., the value of the bound) and the related reference theorem of the single-shot setting. Thus we skip presenting a full proof for Corollaries 11.1.6 and 11.1.7.

Proof of Corollary 11.1.8. Fix an arbitrary sufficiently large n and a $\Delta' \in [20 \log n, n^{\frac{1}{11}}]$ and let $H^*(n, \Delta')$ be the network of the single-shot setting proven to exist in Theorem 6.1.1. We use the same network $H^*(n, \Delta')$, this time in the multi-shot setting.

We first explain the procedure for simulating the single-shot setting of network $H^*(n, \Delta')$ in the multi-shot setting. In the multi-shot setting, for each sender process $proc(v)$, the environment performs a $bcast(m_i)_i$ input action, where $i = id(v)$, at the start of the first round. Moreover, these are the only $bcast()$ inputs throughout the whole execution.

It is easy to see in the multi-shot setting with these special $bcast()$ inputs, for each receiver node v, the contention of process $proc(v)$ is non-increasing in time and in particular, for each round $r \geq 1$, we have $c(v, r) \leq c(v, 1)$. Moreover, for each receiver node v, we have $c(v, 1) = d_{H^*(n,\Delta')}(v)$, where $d_{H^*(n,\Delta')}(v)$ is the G'-degree of receiver node v in network $H^*(n, \Delta')$. Since the maximum G'-receiver degree in $H^*(n, \Delta')$ is at most Δ', we get that for each round $r \geq 1$ of the multi-shot setting problem, $c(v, r) \leq \Delta'$. That is, in the multi-shot setting of network $H^*(n, \Delta')$ with the aforementioned special inputs, the maximum contention is at most Δ'.

Given the multi-shot setting with these special $bcast()$ inputs, for the sake of contradiction, suppose that there exists a local broadcast algorithm $\mathcal{A}$ that has progress bound at most $\frac{\Delta' \log n}{120}$

rounds in this multi-shot setting. We show that there exists a local broadcast algorithm $\mathcal{B}$ that has progress bound at most $\frac{\Delta' \log n}{120}$ rounds in the single-shot setting with bipartite network $H^*(n, \Delta')$.

The algorithm $\mathcal{B}$ is the same as algorithm $\mathcal{A}$ except for two changes: (1) processes in algorithm $\mathcal{B}$ do not receive input actions and do not perform output actions, (2) for any sender process $proc(v)$, if in algorithm $\mathcal{A}$, $proc(v)$ performs an action $ack(m_i)_i$ in round r, then in algorithm $\mathcal{B}$, process $proc(v)$ sets variable $ack_i = True$ in round r.

We next show that $\mathcal{B}$ is a local broadcast algorithm for the single-shot setting with bipartite network $H^*(n, \Delta')$ and that $\mathcal{B}$ has progress bound at most $\frac{\Delta' \log n}{120}$ rounds in this setting.

First note that since algorithm $\mathcal{A}$ satisfies the property (A) of the local broadcast algorithms in the multi-shot setting (presented in Section 8.4), when running $\mathcal{A}$ in the multi-shot setting network $H^*(n, \Delta')$, each sender $proc(v)$ eventually performs an $ack(m_i)_i$ output action, where $i = id(v)$. Thus, when running algorithm $\mathcal{B}$ in the single-shot setting network $H^*(n, \Delta')$, each sender $proc(v)$ eventually sets $ack_i = True$. This proves that $\mathcal{B}$ satisfies property (A) of the local broadcast algorithms in the single-shot setting (presented in Section 4.1).

We now show that $\mathcal{B}$ satisfies property (B) of the local broadcast algorithms in the single-shot setting (presented in Section 4.1). For this, note that $\mathcal{A}$ satisfies the property (C) of the local broadcast algorithms in the multi-shot setting (presented in Section 8.4). Moreover, because of the special from of $bcast()$ inputs that happen only at the start of round 1, the set of executions extending α in the definition of property (C) of local broadcast algorithms in the multi-shot setting (presented in Section 8.4) is in fact the set of all executions with these special $bcast()$ inputs. Thus, the space of executions extending α is in fact the space of all executions with these fixed special $bcast()$ inputs. Because of the property (C) of local broadcast algorithms, we get that if a sender process $proc(v)$ performs an $ack(m_i)_i$ output action at the end of round r, then with high probability, each receiver process $proc(u)$ such that $u \in \mathcal{N}_G(u)$, where G is the reliable part of network $H^*(n, \Delta')$, has output $rcv(m_i)_j$, where $j = id(u)$. Thus, because of the constraints for the processes in the local broadcast algorithm presented in Section 8.2, we get the following: when running algorithm $\mathcal{B}$ in the single-shot setting network $H^*(n, \Delta')$, if a sender process $proc(v)$ sets $ack_i = True$ at the end of round r, then with high probability, each receiver process $proc(u)$ such that $u \in \mathcal{N}_G(u)$, where G is the reliable part of network $H^*(n, \Delta')$, has received the message m_i by the end of round r. This proves that $\mathcal{B}$ satisfies property (B) of the local broadcast algorithms in the single-shot setting (presented in Section 4.1).

Finally, we prove that $\mathcal{B}$ has progress bound at most $\frac{\Delta' \log n}{120}$ rounds in the single-shot setting with network $H^*(n, \Delta')$. Since this contradicts Theorem 6.1.1, proving this progress bound finishes the proof.

First note that similar to the above paragraph, the executions extending α in the definition of the progress bound of multi-shot setting (presented in Section 8.5) are in fact all the executions

with the special $bcast()$ inputs described above. Thus, the space of executions extending α is in fact the space of all executions with these fixed special $bcast()$ inputs.

Now consider an arbitrary receiver process $proc(u)$ that has at least one sender G-neighbor. We claim that when we run $\mathcal{B}$ in the single-shot setting with network $H^*(n, \Delta')$, with high probability, $proc(u)$ receives at least one message m_i by the end of round $\frac{\Delta' \log n}{120}$. We prove this claim in two cases as follows:

First Case Consider the executions of $\mathcal{A}$ in the multi-shot setting with the special $bcast()$ inputs described above in which there exists at least one G-neighbor w of u that has outputs $ack(m_j)_j$, with $j = id(w)$, by the end of round $\frac{\Delta' \log n}{120}$. Then following the property (C) of the local broadcast algorithms in the multi-shot setting (presented in Section 8.4), which is satisfied by $\mathcal{A}$, we get that in these executions of $\mathcal{A}$, with high probability, process $proc(u)$ outputs $rcv(m_j)_i$, where $i = id(u)$, by the end of round $\frac{\Delta' \log n}{120}$. Because of the constraints for the processes in the local broadcast algorithm presented in Section 8.2, this means that in these executions of $\mathcal{A}$, with high probability, process $proc(u)$ receives at least one message m_j by the end of round $\frac{\Delta' \log n}{120}$. Hence, in the corresponding executions of $\mathcal{B}$ in the single-shot setting with network $H^*(n, \Delta')$, with high probability, $proc(u)$ receives at least one message m_j by the end of round $\frac{\Delta' \log n}{120}$. This completes the proof of the claim for the first case.

Second Case Now consider the executions of $\mathcal{A}$ in the multi-shot setting with the special $bcast()$ inputs that do not satisfy the assumption of the first case. That is, the executions of $\mathcal{A}$ in this setting in which no G-neighbor w of v has outputs $ack(m_j)_j$, where $j = id(w)$, by the end of round $\frac{\Delta' \log n}{120}$.

In each of these executions, all G-neighbors of v are active in each round of interval $[1, \frac{\Delta' \log n}{120}]$. Since v has at least one sender G-neighbor, we get that in each round of $[1, \frac{\Delta' \log n}{120}]$, at least one G-neighbor of v is active. Having this, since $\mathcal{A}$ has progress bound at most $\frac{\Delta' \log n}{120}$ rounds in the multi-shot setting with network $H^*(n, \Delta')$ with the special $bcast()$ inputs described above, we get that in these executions of $\mathcal{A}$, with high probability, $proc(u)$ outputs at least one $rcv(m_i)_j$ action, where $j = id(u)$, by the end of round $\frac{\Delta' \log n}{120}$. Because of the constraints for the processes in the local broadcast algorithm presented in Section 8.2, this means that in these executions of $\mathcal{A}$, with high probability, process $proc(u)$ receives at least one message m_i by the end of round $\frac{\Delta' \log n}{120}$. Hence, in the corresponding executions of $\mathcal{B}$ in the single-shot setting with network $H^*(n, \Delta')$, with high probability, $proc(u)$ receives at least one message m_i by the end of round $\frac{\Delta' \log n}{120}$. This completes the proof of the claim for the second case.

Now we know that when running $\mathcal{B}$ in the single-shot setting with network $H^*(n, \Delta')$ (in either of the above two cases), for each arbitrary process $proc(u)$, we have that with high probability,

$proc(u)$ receives at least one message m_i by the end of round $\frac{\Delta' \log n}{120}$. This proves that $\mathcal{B}$ has progress bound [1] at most $\frac{\Delta' \log n}{120}$ rounds, in the single-shot setting with network $H^*(n, \Delta')$. This is in contradiction with Theorem 6.1.1 and this contradiction completes the proof of Corollary 11.1.8. □

Proof of Corollary 11.1.9. Fix an arbitrary sufficiently large n and a $\Delta' \in [20 \log n, n^{\frac{1}{11}}]$ and let $H^*(n, \Delta')$ be the network of the single-shot setting proven to exist in Theorem 6.2.1. We use the same network $H^*(n, \Delta')$, this time in the multi-shot setting.

We first explain the procedure for simulating the single-shot setting of network $H^*(n, \Delta')$ in the multi-shot setting. In the multi-shot setting, for each sender process $proc(v)$, the environment performs a $bcast(m_i)_i$ input action, where $id = id(v)$, at the start of the first round. Moreover, these are the only $bcast()$ inputs throughout the whole execution.

It is easy to see in the multi-shot setting with these special $bcast()$ inputs, for each receiver node v, the contention of process $proc(v)$ is non-increasing in time and in particular, for each round $r \geq 1$, we have $c(v, r) \leq c(v, 1)$. Moreover, for each receiver node v, we have $c(v, 1) = d_{H^*(n,\Delta')}(v)$, where $d_{H^*(n,\Delta')}(v)$ is the G'-degree of receiver node v in network $H^*(n, \Delta')$. Since the maximum G'-receiver degree in $H^*(n, \Delta')$ is at most Δ', we get that for each round $r \geq 1$ of the multi-shot setting problem, $c(v, r) \leq \Delta'$. That is, in the multi-shot setting of network $H^*(n, \Delta')$ with the aforementioned special inputs, the maximum contention is at most Δ'.

Given the multi-shot setting with these special $bcast()$ inputs, for the sake of contradiction, suppose that there exists a local broadcast algorithm $\mathcal{A}$ that has acknowledgment bound at most $\frac{\Delta' \log n}{120}$ rounds in this multi-shot setting. We show that there exists a local broadcast algorithm $\mathcal{B}$ that has acknowledgment bound at most $\frac{\Delta' \log n}{120}$ rounds in the single-shot setting with bipartite network $H^*(n, \Delta')$.

The algorithm $\mathcal{B}$ is the same as algorithm $\mathcal{A}$ except for two changes: (1) processes in algorithm $\mathcal{B}$ do not receive input actions and do not perform output actions, (2) for any sender process $proc(v)$, if in algorithm $\mathcal{A}$, $proc(v)$ performs an $ack(m_i)_i$ action, where $i = id(v)$, in round r, then in algorithm $\mathcal{B}$, process $proc(v)$ sets variable $ack_i = True$ in round r.

We next show that $\mathcal{B}$ is a local broadcast algorithm for the single-shot setting with bipartite network $H^*(n, \Delta')$ and that $\mathcal{B}$ has acknowledgment bound at most $\frac{\Delta' \log n}{120}$ rounds in this setting. We remark that the first part is completely similar to the same part of the proof of Corollary 11.1.8.

First note that since algorithm $\mathcal{A}$ satisfies the property (A) of the local broadcast algorithms in the multi-shot setting (presented in Section 8.4), when running $\mathcal{A}$ in the multi-shot setting network $H^*(n, \Delta')$, each sender $proc(v)$ eventually performs an $ack(m_i)_i$ output action, where $i = id(v)$. Thus, when running algorithm $\mathcal{B}$ in the single-shot setting network $H^*(n, \Delta')$, each sender $proc(v)$

[1] Refer to Section 4.2 for the definition of the progress bound in the single-shot setting.

eventually sets $ack_i = True$. This proves that $\mathcal{B}$ satisfies property (A) of the local broadcast algorithms in the single-shot setting (presented in Section 4.1).

We now show that $\mathcal{B}$ satisfies property (B) of the local broadcast algorithms in the single-shot setting (presented in Section 4.1). For this, note that $\mathcal{A}$ satisfies the property (C) of the local broadcast algorithms in the multi-shot setting (presented in Section 8.4). Moreover, because of the special from of $bcast()$ inputs that happen only at the start of round 1, executions extending α in the definition of property (C) of local broadcast algorithms in the multi-shot setting (presented in Section 8.4) are in fact the set of all executions with these special $bcast()$ inputs. Thus, the space of executions extending α is in fact the space of all executions with these fixed special $bcast()$ inputs.

Because of the property (C) of local broadcast algorithms, we get that for each sender process $proc(v)$ and each round r, if process $proc(v)$ performs an $ack(m_i)_i$ output action, where $i = id(v)$, at the end of round r, then with high probability, each receiver process $proc(u)$ such that $u \in \mathcal{N}_G(u)$, where G is the reliable part of network $H^*(n, \Delta')$, has output $rcv(m_i)_j$, where $j = id(u)$. Thus, because of the constraints for the processes in the local broadcast algorithm presented in Section 8.2, we get the following: when running algorithm $\mathcal{B}$ in the single-shot setting network $H^*(n, \Delta')$, if a sender process $proc(v)$ sets $ack_i = v$ at the end of round r, then with high probability, each receiver process $proc(u)$ such that $u \in \mathcal{N}_G(u)$, where G is the reliable part of network $H^*(n, \Delta')$, has received the message m_i by the end of round r. This proves that $\mathcal{B}$ satisfies property (B) of the local broadcast algorithms in the single-shot setting (presented in Section 4.1).

Finally, we prove that $\mathcal{B}$ has acknowledgment bound at most $\frac{\Delta' \log n}{120}$ rounds in the single-shot setting with network $H^*(n, \Delta')$. Since this contradicts Theorem 6.2.1, proving this acknowledgment bound finishes the proof.

First note that similar to the above paragraph, the set of executions extending α in the definition of the progress bound of multi-shot setting (presented in Section 8.5) is in fact the set of all executions with the special $bcast()$ inputs described above. Thus, the space of executions extending α is in fact the space of all executions with these fixed special $bcast()$ inputs.

Now consider an arbitrary sender process $proc(v)$. Note that algorithm $\mathcal{A}$ has acknowledgment bound at most $\frac{\Delta' \log n}{120}$ rounds in the multi-shot setting with network $H^*(n, \Delta')$ with the special $bcast()$ inputs. Thus, when we run $\mathcal{A}$ in this setting, for each sender process $proc(v)$, with high probability we have that $proc(v)$ outputs $ack(m_i)_i$, where $i = id(v)$, by the end of round $\frac{\Delta' \log n}{120}$. Hence, when we run $\mathcal{B}$ in the single-shot setting with network $H^*(n, \Delta')$, for each sender process $proc(v)$, with high probability we have that $proc(v)$ sets $ack_i = True$, where $i = id(v)$, by the end of round $\frac{\Delta' \log n}{120}$. This proves that algorithm $\mathcal{B}$ has acknowledgment bound [2] at most $\frac{\Delta' \log n}{120}$ rounds, in the single-shot setting with network $H^*(n, \Delta')$. This is in contradiction with Theorem 6.2.1 and this contradiction completes the proof of Corollary 11.1.9. □

[2] Refer to Section 4.2 for the definition of the acknowledgment bound in the single-shot setting.

Chapter 12

Conclusion

In this thesis, we studied the local broadcast problem, which is a theoretical approach to capturing the contention management issue of the radio networks. In the local broadcast problem (in its general multi-shot case), the processes receive messages one by one, and they should deliver these messages to all their neighbors. We studied the local broadcast problem in two models, namely, the classical radio network model and the dual graph model. The former model is a standard and well-used model for the radio networks. The latter is a more recent model that generalizes the former by including a set of unreliable adversarially-controlled edges. These unreliable edges try to capture the reality of the practical radio networks where usually, some of the connections are unavoidably unreliable.

For these two models, we studied the local broadcast problem with respect to two specific measures: the acknowledgment bound and the progress bound. Roughly speaking, the acknowledgment bound measures the time it takes each process to deliver its message to all its neighbors in the reliable graph. This measure is a usual and natural way for capturing the performance of the local broadcast algorithms. The progress bound however is from a different viewpoint; roughly speaking, it measures the time it takes one process to receive at least one message from its neighbors, regardless of which message, assuming that there is at least one reliable neighbor that is transmitting. The progress bound has been a crucial tool in getting tighter analysis of many higher-layer problems such as global broadcast.

The key point in this thesis was showing that a slightly optimized variant of the Decay protocol, which is a standard solution for the local broadcast problem introduced by Bar-Yehuda, Goldreich, and Itai [19], achieves asymptotically optimal progress and acknowledgment bounds in both the classical and the dual graph models. In a closer view, we showed the following results:

It has been known that in the classical model, the acknowledgment bound achieves progress and acknowledgment bounds, respectively, $O(\log \Delta \log n)$ and $O(\Delta \log \Delta \log n)$ rounds. We showed in Chapter 10 (particularly Theorem 10.2.1) that by a simple change in this algorithm, we can re-

duce the acknowledgment bound to $O(\Delta \log n)$ rounds while keeping the progress bound $O(\log \Delta \log n)$ rounds. In Corollaries 11.1.6 and 11.1.7, we showed that $\Omega(\log \Delta \log n)$ and $\Omega(\Delta \log n)$ rounds are lower bounds for, respectively, progress and acknowledgment bounds in the classical model. These lower bounds prove that the optimized variant of the decay algorithm presented in Section 10.1 has asymptotically optimal progress and acknowledgment bounds in the classical model.

We moreover showed that the analysis of the optimized decay algorithm extend to the dual graph model, proving that it achieves progress and acknowledgment bounds, respectively, $O(\Delta' \log n)$ and $O(\Delta' \log n)$. In Corollaries 11.1.8 and 11.1.9, we showed that $\Omega(\Delta' \log n)$ is a lower bound for both progress and acknowledgment bounds in the dual graph model. These results prove that the optimized variant of the decay algorithm presented in Section 10.1 achieves optimal progress and acknowledgment in the dual graph model as well.

The results about the progress bound shed light on an important difference between the classical model and the dual graph model. In particular, while the optimized decay algorithm achieves progress bound $O(\log \Delta \log n)$ in the classical model, the progress bound in the dual graph model is lower bounded by $\Omega(\Delta' \log n)$. This shows an exponential difference between the dependence of the progress bounds of the two models on the maximum contention. This proves that in the presence of unreliability, progress is unavoidably harder (slower). It also has the practical message that we cannot completely trust the performance analysis of the contention management solutions when they are based on a reliable model as the classical model but the algorithm is used in the radio network where unreliability is usually unavoidable.

Future Work There are a number of interesting questions that remain to be studied:

- Perhaps the most important question is regarding the power of the adversary in the dual graph model. In the dual graph model that we studied in this thesis, the communications on the unreliable edges are controlled by an offline adaptive adversary [8]. This adversary in particular knows the outcome of the coin flips that the processes use for deciding whether to transmit or listen in the current round. It can be argued that this amount of knowledge and power might be unrealistic for the usual practical radio networks, because the unreliability of the practical radio network appears to be more oblivious to the random choices of the processes. This raises the following natural question: "How do the upper and lower bounds change if we relax the adversary and consider an *online adaptive adversary* or an *oblivous adversary* [8]?". An online adaptive adversary does not know the outcome of the coin flips of the current round. An oblivious adversary has no knowledge about the outcome of the coin flips used throughout the execution and has to make its decisions at the very start of execution.

- Another issue is whether the guarantees provided by the local broadcast algorithms are deterministic or probabilistic. Note that a deterministic algorithm would clearly provide deterministic guarantees, guarantees that are correct in every execution. The algorithms that we studied in this thesis have a probabilistic guarantee: when a process v acknowledges that it has delivered a message m to all of its neighbors (by outputting $ack(m)_i$, where $i = id(v)$), there is a small but nonzero probability at most $\frac{1}{n}$ that the message m is not delivered to all the neighbors. This might be considered as a deficiency for these randomized algorithms. Also, this small probability of faulty guarantee might be even intolerable for some higher layer algorithms that use the local broadcast algorithm as a sub-module and might rely on the fact that its acknowledgment outputs must guarantee the delivery of the messages deterministically. It is interesting to study the complexity of the local broadcast problem when the algorithms are required to provide this stronger deterministic guarantee. Note that even though deterministic algorithms will have deterministic guarantees automatically, this deterministic guarantee can be also achieved by using a randomized algorithms.

- The next question is related to the special structure of the networks used in our lower bound results. Even though our lower bound results show that there exist networks which require large progress or acknowledgment bounds in either the classical or the dual graph model, it can be argued that such bad networks are rather unlikely to appear in practice. Regarding this point, an important question is to study the complexity of the local broadcast problem when the network graph is guaranteed to be from a well-behaved family. This family can be chosen such that it matches the reality of the practical networks better than the general arbitrary graphs. For instance, it is important to study how the complexity of the problem changes if the network graph has bounded independence [34]. Bounded independence graphs have been argued to be close to the reality of practical radio network.

- It is interesting to investigate the complexity of the local broadcast problem in other radio network models and in particular, in the SINR model [30]. The SINR model has been recently gained more attention in the theory of wireless networks community as many believe it to be closer to reality.

- The next future work direction is to study the approach of solving the higher layer problems on top of the local broadcast algorithms. In particular, it is interesting to compare the performance of the higher layer algorithms that work on top of a local broadcast algorithm with that of the algorithms that do not rely on such a building block and solve the contention management issue of the radio networks directly. Since the higher layer algorithms of the second type solve the contention management issue with the intention of optimizing the solution for a particular problem, we might expect that in many problems, the performance of

the second type of higher layer algorithms can be better than that of the first type. However, most likely, this comes at the cost of having a more involved algorithm that deals with the challenges of the higher layer problem as well as the challenges of contention management. It is also interesting to see how different definitions of the guarantees that a local broadcast algorithm provides might affect the performance of the higher layer algorithms that are built on top of it.

- An important future work direction is about another type of the contention management issue that is different than the local broadcast problem but is closely related to it. In the majority of the practical MAC layers deployed today, the MAC layer tries to deliver the message of the transmitting process to only one particular neighbor of that process. This is different than the local broadcast problem where the goal is to broadcast the message to all the neighbors. Also, for many theoretical higher layer problems, such a local message delivery service might be enough. It is interesting to study the complexity of this weaker variant of the contention management issue. We think that the techniques used in this thesis for deriving lower bounds for the local broadcast problem might be useful for obtaining lower bounds for algorithms that implement this weaker local message delivery service, i.e., where the goal is to deliver the message to only one particular neighbor.

In conclusion, we believe that the complete characterization of the local broadcast problem that we provide in this thesis for the classical and the dual graph radio models can be a good starting point in further understanding the contention management issue of the radio networks. We hope that the results and the techniques presented in this thesis might be helpful while studying these future work directions.

Bibliography

[1] Bachir, A., Dohler, M., Wattayne, T., and Leung, K.: " MAC Essentials for Wireless Sensor Networks". IEEE Communications Surveys and Tutorials 12, 2 (2010), 222-248.

[2] Shan, H., Zhuang, W., and Wand, Z.: "Distributed Cooperative MAC for Multihop Wireless Networks". IEEE Communications Magazine 47, 2 (February 2009), 126-133.

[3] Sato, N., and Fujii, T.: "A MAC Protocol for Multi-Packet Ad-Hoc Wireless Network Utilizing Multi-Antenna". In Proceedings of the IEEE Conference on Consumer Communications and Networking (2009).

[4] Sayed, S., and Yand, Y.: "BTAC: A Busy Tone Based Cooperative MAC Protocol for Wireless Local Area Networks". In Proceedings of the Interneational Conference on Communications and Networking in China (2008).

[5] Sun, Y., Gurewitz, O., and Johnson, D. B.: "RI-MAC: a Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks", Proceedings of the ACM Conference on Embedded Network Sensor Systems, 2008.

[6] Rhee, I., Warrier, A., Aia, M., Min, J., and Sichitiu, M. L.: "Z-MAC: a Hybrid MAC for Wireless Sensor Networks". IEEE/ACM Trans. on Net. 16 (June 2008), 511-524.

[7] Alon, N. and Spencer, J. H.: "The probabilistic method". John Wiley & Sons, New York, 1992.

[8] Allan Borodin and Ran El-Yaniv. : "Online Computation and Competitive Analysis". Cambridge University Press, New York, NY, USA. 1998

[9] Tsybakov, B. S., and Mikhailov, V. A.: "Free synchronous packet access in a broadcast channel with feedback". Prob. Infi Transmission 14, 4 (April 1978), 259-280. (Translated from Russian original in Prob. Peredach. Znf, Ott-Dec., 1977).

[10] Capetanakis, J.: "Tree algorithms for packet broadcast channels". IEEE Trans. ZnJ: Theory IT-25, 5 (Sept. 1979) 505-5 15.

[11] Capetanakis, J.: "Generalized TDMA: The multi-accessing tree protocol". IEEE Trans. Commun. COM-27, 10 (Oct. 1979), 1479-1484.

[12] Hayes, J. F.: "An adaptive technique for local distribution". IEEE Trans. Commun. COM-26, (Aug. 1978), 1178-l 186.

[13] Massey, J. L.: "Collision-resolution algorithms and random-access communications". Technical Report UCLA-ENG-8016, School of Engineering and Applied Science, University of California, Los Angeles, Los Angeles, Calif., April 1980.

[14] Greenberg, A. G. and Lander, R . E.: "Estimating the multiplicities of conflicts in multiple access channels". In Proceedings of the 24th Annual Symposium on Foundations of Computer Science (Tucson, AZ.). IEEE, New York, 1983, 383-392.

[15] Willard, D.: "Log-logarithmic protocols for resolving Ethernet and semaphore conflicts". In Proceedings of the 16th Annual ACM Symposium on Theory of Computing (Washington, DC., Apr. 30-May 2). ACM, New York, 1984,5 12-52 1.

[16] Greenberg, A. G., and Winograd, S.: "A Lower Bound on the Time Needed in the Worst Case to Resolve Conflicts Deterministically in Multiple Access Channels". J. ACM 32, 3 (July 1985), 589-596.

[17] Komlos, J. and Greenberg, A. G.,: "An asymptotically nonadaptive algorithm for conflict resolution in multiple-access channels". IEEE Trans. on Information Theory, 31 (1985)303 - 306.

[18] Chlamtac, I., Kutten, S. : "On Broadcasting in Radio Networks–Problem Analysis and Protocol Design". IEEE Trans. on Communications (1985).

[19] Bar-Yehuda, R., Goldreich, O., and Itai, A.: "On the time-complexity of broadcast in radio networks: an exponential gap between determinism randomization". In PODC 87: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing, pages 98-108, New York, NY, USA, 1987. ACM.

[20] Alon, N., Bar-Noy, A., Linial, N., and Peleg., D.: "A lower bound for radio broadcast". J. Comput. Syst. Sci., 43(2):290-298, 1991.

[21] Alon, N., Bar-Noy, A., Linial, N., and Peleg., D. : "On the complexity of radio communication". In Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC '89), D. S. Johnson (Ed.). ACM, New York, NY, USA, 274-285.

[22] Alon, N., Bar-Noy, A., Linial, N., and Peleg., D. : "Single round simulation on radio networks". J. Algorithms 13, 2 (June 1992), 188-210.

[23] Chrobak, M., Gasieniec, L., and Rytter, W.: "Fast broadcasting and gossiping in radio networks", J. Algorithms 43, 2 (May 2002), 177-189.

[24] Chlebus, B. S., Gasieniec, L., Gibbons, A., Pelc, A., and Rytter, W.: "Deterministic broadcasting in unknown radio networks". In Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms (SODA '00). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 861-870.

[25] Chlebus, B. S., Gasieniec, L., Ostlin, A., and Robson, J. M.: "Deterministic Radio Broadcasting" ICALP 2000.

[26] Clementi, A., Monti, A. , and Silvestri, R.: "Selective families, superimposed codes, and broadcasting on unknown radio networks". In the annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 709-718, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[27] Clementi, A., Crescenzi, P., Monti, A., Penna, P., and Silvestri, R.: "On Computing Ad-hoc Selective Families". In Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 5th International Workshop on Randomization and Approximation Techniques in Computer Science: Approximation, Randomization and Combinatorial Optimization, pages 211–222, 2001.

[28] Clementi, A., Monti, A. , and Silvestri, R.: "Round robin is optimal for fault-tolerant broadcasting on wireless networks". J. Parallel Distrib. Comput., 64(1):89-96, 2004.

[29] Gasieniec, L., Peleg, D., and Xin, Q.: "Faster communication in known topology radio networks". In Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing (PODC '05). ACM, New York, NY, USA, 129-137.

[30] Lotker, Z., and Peleg, D. : "Structure and algorithms in the SINR wireless model". SIGACT News 41, 2 (June 2010), 74-84.

[31] Kuhn, F., Lynch, N., and Newport, C. : "The Abstract MAC Layer". In I. Keidar, editor, Distributed Computing: DISC 2009: 23rd International Symposium on Distributed Computing, Elche/Elx, Spain, September 23-25, 2009, volume 5805 of Lecture Notes in Computer Science, pages 48-62, 2009. Springer.

[32] Kuhn, F., Lynch, N., and Newport, C. : "The Abstract MAC Layer". Technical Report MIT-CSAIL-TR-2010-040, Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, August 2010.

[33] Kuhn, F., Lynch, N., and Newport, C.: "Brief Announcement: Hardness of Broadcasting in Wireless Networks with Unreliable Communication". Proceedings of the ACM Symposium on the Principles of Distributed Computing (PODC), Calgary, Alberta, Canada, August 2009.

[34] Kuhn, F., Nieberg, T., Moscibroda, T., and Wattenhofer, R. : "Local approximation schemes for ad hoc and sensor networks". In Proceedings of the 2005 joint workshop on Foundations of mobile computing (DIALM-POMC '05). ACM, New York, NY, USA, 97-103.

[35] Cornejo, A., Lynch, N., Viqar, S., and Welch, J.: "A Neighbor Discovery Service Using an Abstract MAC Layer". Forty-Seventh Annual Allerton Conference, Champaign-Urbana, IL, October, 2009. Invited paper.

[36] Khabbazian, M., Kuhn, F., Kowalski, D. R., and Lynch. N.: "Decomposing broadcast algorithms using abstract MAC layers". In Proceedings of the 6th International Workshop on Foundations of Mobile Computing (DIALM-POMC '10). ACM, New York, NY, USA, 13-22.

[37] Khabbazian, M., Kuhn, F., Kowalski, D. R., and Lynch. N.: "Decomposing broadcast algorithms using abstract MAC layers". To appear in Ad Hoc Networks. Elsevier.

[38] Khabbazian, M., Kuhn, F., Lynch, N., Medard, M., and ParandehGheibi, A.: "MAC Design for Analog Network Coding". FOMC 2011: The Seventh ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing, San Jose, CA, June 2011.

[39] Censor-Hillel, K., Gilbert, S., Kuhn, F., Lynch, N., and Newport, C.: "Structuring Unreliable Radio Networks". Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, San Jose, California, June 6-8, 2011.

[40] Lynch, N., and Radeva, T., and Sastry, S.: "Asynchronous Leader Election and MIS Using Abstract MAC Layer". Submitted for publication, 2011.

[41] Ghaffari, M., Haeupler, B., Lynch, N., and Newport. C. : "Bounds on Contention Management in Radio Networks" In Proceedings of the 26th international conference on Distributed Computing (DISC'12), Marcos K. Aguilera (Ed.). Springer-Verlag, Berlin, Heidelberg, 223-237.

[42] Ghaffari, M., Haeupler, B., Lynch, N., and Newport, C.: "Bounds on Contention Management in Radio Networks". http://arxiv.org/abs/1206.0154.