

**The Loudest One Wins:  
Efficient Communication in Theoretical Wireless  
Networks**

by

Evangelia Anna Markatou

Submitted to the Department of Electrical Engineering and Computer  
Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Author .....  
Department of Electrical Engineering and Computer Science  
May 23, 2018

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

Accepted by .....  
Katrina LaCurts  
Chair, Master of Engineering Thesis Committee

**The Loudest One Wins:  
Efficient Communication in Theoretical Wireless Networks**

by

Evangelia Anna Markatou

Submitted to the Department of Electrical Engineering and Computer Science  
on May 23, 2018, in partial fulfillment of the  
requirements for the degree of  
Master of Engineering in Computer Science and Electrical Engineering

**Abstract**

In order to develop the most efficient algorithms for wireless networks, first one must understand their theoretical limitations. To this end, we study the *leader election* and *broadcast* problems in wireless networks, modeling them using the Signal-to-Interference-plus-Noise-Ratio (SINR) model.

Our main result is an algorithm that solves the leader election problem in two communication rounds using power control, with high probability. Previously, it was known that  $\Omega(\log n)$  rounds were sufficient and necessary when using uniform power, where  $n$  is the number of nodes in the network. We explore tradeoffs between communication complexity and power used, and show that to elect a leader in  $t$  rounds, a power range  $\exp(n^{1/\Theta(t)})$  is sufficient and necessary.

In addition, we present an efficient algorithm for the broadcast problem. Using power control, it is possible to achieve a broadcast algorithm that terminates successfully in  $2n$  rounds, w.h.p..

Thesis Supervisor: Nancy Lynch

Title: Professor

## Acknowledgments

First, I would like to thank my advisor Nancy Lynch for her help and support during my Masters. Without her diligence and attentive reading, my Masters Thesis could never have been what it is.

I would like to thank my supervisors Stephan Holzer and Hsin-Hao Su for cultivating my ideas and supporting my desire to research and publish.

I would like to thank Magnús Halldórsson for being a great collaborator and mentor.

Finally, I would like to thank Mira Radeva, Evangelos Markatos, Catherine Chronaki, Gwendolyn Edgar, Barbara Duckworth and Haben Zaid for helpful comments and discussions.

# Contents

<b>1</b>	<b>Introduction</b>	<b>8</b>
1.1	Why Now? . . . . .	8
1.2	Contributions . . . . .	11
1.3	Publications . . . . .	12
1.4	Organization of the Thesis . . . . .	12
<b>2</b>	<b>Models and Related Work</b>	<b>14</b>
2.1	Preliminaries . . . . .	14
2.2	Models . . . . .	14
2.2.1	Radio Network Model . . . . .	15
2.2.2	SINR Model . . . . .	15
2.3	Leader Election Problem . . . . .	17
2.3.1	Prior Work on the Leader Election Problem . . . . .	17
2.3.2	Contributions on the Leader Election Problem . . . . .	18
2.4	Broadcast Problem . . . . .	19
2.4.1	Prior Work on the Broadcast Problem . . . . .	19
2.4.2	Contributions on the Broadcast Problem . . . . .	21

<b>3</b>	<b>Achieving Leader Election in Two Rounds</b>	<b>22</b>
3.1	The Essence of the Algorithm . . . . .	22
3.2	Leader Election Algorithm, $2-LE$ . . . . .	23
3.2.1	Pseudocode for Algorithm $2-LE_v$ . . . . .	24
3.3	Analysis of Algorithm $2-LE$ . . . . .	25
<b>4</b>	<b>Upper Bound on the Power Needed for a 2-Round Leader Election</b>	
	<b>Algorithm</b>	<b>31</b>
4.1	Upper Bound on the Power Needed with no knowledge of $n$ . . . . .	32
4.1.1	Algorithm $2-LE(P_{max})$ . . . . .	32
4.1.2	Analysis of Algorithm $2-LE(P_{max})$ . . . . .	34
4.2	Assuming $n$ , a Smaller Power Bound is Achievable . . . . .	35
4.2.1	Algorithm $2-LEn$ . . . . .	35
4.2.2	Analysis of Algorithm $2-LEn$ . . . . .	37
<b>5</b>	<b>Leader Election: Trading Time for Power</b>	<b>39</b>
5.1	Algorithm $2t-LE(P_{max})$ . . . . .	39
5.2	Analysis of Algorithm $2t-LE(P_{max})$ . . . . .	41
<b>6</b>	<b>Lower Bound on the Power Needed for a <math>t</math>-Round Leader Election</b>	
	<b>Algorithm</b>	<b>45</b>
6.1	Power Needed to Break the Symmetry of a Network . . . . .	46
6.2	Lower Bound on Power Needed for a $t$ -Round Leader Election Algorithm	50
<b>7</b>	<b>All-to-All Broadcast with Power Control</b>	<b>53</b>
7.1	Augmented Leader Election . . . . .	53
7.1.1	Pseudocode for Algorithm $2-LEp_v$ . . . . .	54

7.1.2	Analysis of Algorithm 2- <i>LEp</i> . . . . .	56
7.2	All-to-All Broadcast Algorithm . . . . .	58
7.2.1	The LeCirio Algorithm . . . . .	58
7.2.2	Analysis of LeCirio . . . . .	61
<b>8</b>	<b>Contributions</b>	<b>66</b>
8.1	Future Work . . . . .	66

# List of Figures

1-1	Demonstrating the capture effect. Nodes $A$ and $B$ are broadcasting using the same power, and nodes $D, E$ and $C$ are listening. Due to the network topology, the proximity between $A$ and $D$ allows $D$ to receive $A$ 's message. Similarly, $E$ receives $B$ 's message. However, because $C$ is approximately the same distance away from $A$ and $B$ , $C$ receives no message. . . . .	9
1-2	Demonstrating power control. Nodes $A$ and $B$ are broadcasting, and node $C$ is listening. Even though $C$ is approximately the same distance away from $A$ and $B$ , because $A$ is broadcasting with a much higher power, $C$ receives $A$ 's message. . . . .	10
1-3	Generalizing power control. All the red nodes are broadcasting, and node $C$ is listening. Even though $C$ is approximately the same distance away from $A$ and the cluster of nodes, because $A$ is broadcasting with a much higher power, $C$ receives $A$ 's message. . . . .	11

# Chapter 1

## Introduction

Wireless networks are everywhere around us; wireless devices, from satellites in the sky to Internet of Things components, need to communicate efficiently. In this thesis, we use the Signal-to-Interference-plus-Noise-Ratio (SINR) model to realistically model wireless networks and develop efficient algorithms.

### 1.1 Why Now?

Originally, wireless networks were modeled using the *radio network model*, as for example in work by Chlamtac and Kutten [7]. In this model, there are  $n$  nodes in the network, and in every communication round, each one of them has the option to broadcast a message, or listen. A message is successfully received by a listening node if exactly one of its neighboring nodes is broadcasting. We define the model more formally in Chapter 2.

This model fails to capture interference in wireless networks realistically. This is where the *fading radio network model*, and more specifically, the SINR model comes



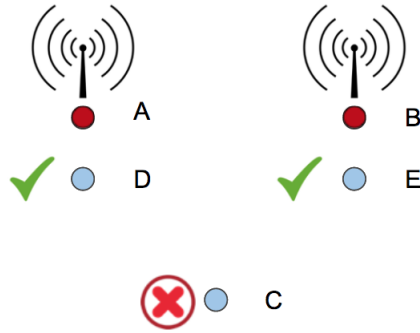


Figure 1-1: Demonstrating the capture effect. Nodes  $A$  and  $B$  are broadcasting using the same power, and nodes  $D$ ,  $E$  and  $C$  are listening. Due to the network topology, the proximity between  $A$  and  $D$  allows  $D$  to receive  $A$ 's message. Similarly,  $E$  receives  $B$ 's message. However, because  $C$  is approximately the same distance away from  $A$  and  $B$ ,  $C$  receives no message.

in. In this model, there are  $n$  nodes in the network, and in every communication round, each one of them has the option to broadcast a message, or listen. A message is successfully received by a listening node if the signal of the broadcasting node is louder than the interference, which consists of ambient noise and other broadcasting nodes' signals. The larger the distance between the sender and the receiver, the smaller the received signal is.

The SINR model allows for more efficient algorithms for certain problems [20], as it has two significant capabilities, the **capture effect** and **power control**. The capture effect describes the capability of wireless networks to have multiple nodes broadcasting at the same time, with some of the messages received. Figure 1-1 demonstrates the capture effect. Nodes  $A$  and  $B$  are broadcasting using the same power, and nodes  $D$ ,  $E$  and  $C$  are listening. Due to the network topology, the proximity between  $A$  and  $D$  allows  $D$  to receive  $A$ 's message. Similarly,  $E$  receives

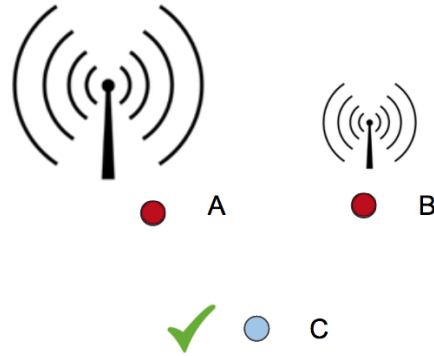


Figure 1-2: Demonstrating power control. Nodes  $A$  and  $B$  are broadcasting, and node  $C$  is listening. Even though  $C$  is approximately the same distance away from  $A$  and  $B$ , because  $A$  is broadcasting with a much higher power,  $C$  receives  $A$ 's message.

$B$ 's message. However, because  $C$  is approximately the same distance away from  $A$  and  $B$ , it receives no message. Note that if we didn't take interference into account, none of these messages could have been received.

The other important capability is power control. This allows the nodes to change their transmission power. Figures 1-2 and 1-3 demonstrate power control. In Figure 1-2, nodes  $A$  and  $B$  are broadcasting, and node  $C$  is listening. Even though  $C$  is approximately the same distance away from  $A$  and  $B$ , because  $A$  is broadcasting with a much higher power,  $C$  receives  $A$ 's message.

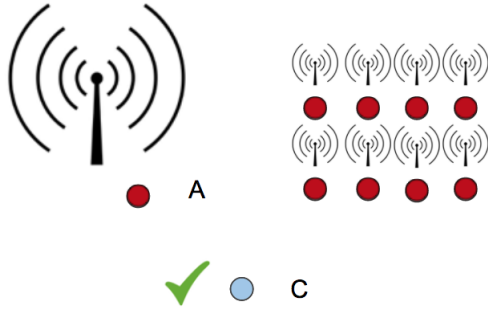


Figure 1-3: Generalizing power control. All the red nodes are broadcasting, and node  $C$  is listening. Even though  $C$  is approximately the same distance away from  $A$  and the cluster of nodes, because  $A$  is broadcasting with a much higher power,  $C$  receives  $A$ 's message.

In Figure 1-3, we generalize the above concept. All the red nodes are broadcasting, and node  $C$  is listening. Even though  $C$  is approximately the same distance away from  $A$  and the cluster of nodes, because  $A$  is broadcasting with a much higher power,  $C$  receives  $A$ 's message.

Note that again because we took interference into account by using the SINR model, and not the radio network model, communication occurred. Power control and the capture effect allow us to develop more efficient algorithms for the leader election problem in wireless networks.

## 1.2 Contributions

We worked on the leader election and the broadcast problems in single-hop wireless networks in the SINR model. Leader election is the problem of choosing one node out of  $n$ , with all nodes knowing if they are the chosen one. All-to-All Broadcast is the problem of each node communicating its message to every other node.

This thesis includes the following contributions:

1. We developed a 2-round leader election protocol. The previously best result was  $O(\log n)$  rounds.
2. We explored trade-offs between transmission power and communication complexity for the leader election problem. We showed that in order to elect a leader in  $t$  rounds, a power range  $\exp(n^{1/\Theta(t)})$  is sufficient and necessary.
3. We developed an all-to-all broadcast algorithm that uses power control, and solves the broadcast problem in  $2n$  rounds.

## 1.3 Publications

The work on the leader election problem first appeared as a brief announcement in the Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC) in 2017 [15]. The full paper on leader election appeared at SIROCCO 2017 [16], where it was awarded the **Best Paper Award**. The full paper was also invited to appear in a special issue of the Theoretical Computer Science journal dedicated to SIROCCO 2017.

## 1.4 Organization of the Thesis

Chapter 2 includes the model and related work. Chapter 3 presents the two round leader election protocol, and Chapter 4 presents upper bounds on the required power for a two round leader election protocol. In Chapter 5, we explore the trade-offs between transmission power and communication complexity for the leader election problem. In Chapter 6, we explore how much power is really necessary for leader

election. The broadcast algorithm can be found in Chapter 7. In Chapter 8, we summarize our contributions and suggest possible future work.

# Chapter 2

## Models and Related Work

In this chapter, we present the some preliminaries, and then the model and problem statements along with related work in this area.

### 2.1 Preliminaries

All networks contain  $n$  nodes. We define that an event happens with high probability (w.h.p.) if it happens with probability greater than  $1 - 1/n$ . The  $\tilde{O}$ -notation omits logarithmic factors. All logs are base 2.

### 2.2 Models

We present two models, the radio network model and the SINR model. Originally, the radio network model was used to describe wireless networks. The SINR model describes wireless networks more realistically, and allows for more efficient communication in certain cases, as proved by [20].

## 2.2.1 Radio Network Model

The radio network model was studied in the 1980s [2] in order to model and analyze algorithms in wireless networks.

Let  $V$  be a set of  $n$  nodes that represent wireless devices. Define an undirected graph  $G(V, E)$ , where an edge  $e \in E$  between nodes  $u$  and  $v$  denotes that if node  $u$  broadcasts a message, it will reach  $v$ . Time is divided into synchronous rounds. In each round, a node  $v$  can either transmit a message of size  $O(\log n)$ , or listen. Node  $v \in V$  receives a message transmitted by neighboring node  $u \in V$ , iff  $v$  is listening and none of the neighbors of  $v$  are broadcasting (other than  $u$ ), that is if only the message from  $u$  reaches  $v$ .

## 2.2.2 SINR Model

In the early 2000s came a renewed interest in the radio networks, along with the development of the SINR model in an effort to create more realistic models [14].

Let  $V$  be a set of  $n$  nodes, that represent wireless devices, deployed in a single-hop network located on a metric space, as in work by Magnús M. Halldórsson and Pradipta Mitra [17].

Time is divided into synchronous rounds. In each round, a node  $v$  can either transmit a message of size  $O(\log n)$  with some power  $P_v$ , or listen. Node  $v \in V$  receives a message transmitted by node  $u \in V$ , iff  $v$  is listening and

$$SINR(u, v, I) = \frac{\frac{P_u}{d(u,v)^\alpha}}{N + \sum_{w \in I} \frac{P_w}{d(w,v)^\alpha}} \geq \beta, \quad (2.1)$$

where  $I$  is the set of other nodes transmitting simultaneously,  $d(u, v)$  is the directed

distance from node  $u$  to node  $v$ , and  $\alpha \in \mathbb{R}_{\geq 1}, \beta \in \mathbb{R}_{>0}, N \in \mathbb{R}_{>0}$  are constants. Specifically,  $\alpha$  is the *path-loss exponent*,  $N$  is the *non-zero ambient noise*, and  $\beta$  is a *hardware-dependent minimum SINR threshold* required for a successful message reception.

The algorithms work for any  $\beta > 0$ , while the leader election lower bounds use  $\beta > 2$ .

We define  $d_{max} = \max_{(u,v)}(d(u,v))$ , and  $d_{min} = \min_{(u,v)}(d(u,v))$ . We define transmission power  $P_{min}$  as the minimum transmission power required such that any node  $u$  can communicate with any other node  $v$ , in absence of interference from other nodes, that is

$$P_{min} = \beta N (d_{max})^\alpha.$$

Denote by  $R = \frac{d_{max}}{d_{min}}$  the ratio of the longest to shortest distance between any two nodes in the network. Similarly to Fineman et al. [10], assume that  $R$  is bounded by a polynomial in  $n$ ,  $R \leq n^c$ , for some  $c \in \mathbb{N}$ .

Let  $\gamma$  be a constant such that

$$\gamma \geq \max(1, c\alpha + 1 + \log \beta).$$

Note that the max is required as  $\beta$  might be smaller than 1. We assume that the nodes know  $\gamma$ .

An algorithm uses *uniform power* if all nodes transmit messages with the same transmission power. If power control is used, nodes can transmit messages with varying transmission powers.



## 2.3 Leader Election Problem

The leader election problem is defined as:

**Problem 2.1** (Leader Election Problem). *Eventually elect exactly one node (called the leader), with all nodes knowing whether or not they are the leader.*

### 2.3.1 Prior Work on the Leader Election Problem

The leader election problem was first studied in the 1970s, when the ALOHA radio network system was built [1], and plenty of work considering this problem was published in the following decade. Gallager’s paper [11] contains a good survey of early work on leader election. Starting in the 1990s there was an increased interest in the *radio network model* [7]. In this model, the general leader election problem, where the nodes don’t have collision detection or know an upper bound for  $n$ , can be solved in  $O(\log^2 n)$  rounds w.h.p. [21] where  $n$  is the number of nodes in the network. This bound can be improved to  $O(\log n)$  w.h.p. assuming that nodes can detect collisions [21], and to  $O(\log n_u)$  expected rounds assuming that the nodes know an upper bound  $n_u$  on  $n$  [3].

In the beginning of the new millennium came a renewed interest in *fading radio networks*, captured with the SINR model, which is claimed to capture the real behavior of systems better than previous models, as they take interference into account in a more realistic way. Moscibroda and Wattenhofer [20] showed that algorithms on the fading radio networks model can achieve better runtimes than algorithms for the original radio networks model on certain problems, as SINR allows for better spatial reuse.

In the SINR model the most efficient currently published contention resolution protocol is by Fineman et al. [10]. Contention resolution is a similar problem to leader election as they both deal with breaking the symmetry in a wireless network. Contention resolution is solved in the first round when exactly one node transmits.

Fineman et al. present an algorithm that achieves a solution to the contention resolution problem in  $O(\log n + \log R)$  rounds w.h.p. in a single-hop network using uniform transmission power, where  $n$  is the number of nodes and  $R = O(\text{poly}(n))$  is the ratio between the longest and shortest link. Fineman et al. suggest that it may be possible to achieve better performance using power control.

Indeed, for problems like link scheduling and connectivity, power control has been shown to give much better performance [20]. Power control has also been used in the SINR setting to solve the link scheduling problem while conserving energy, e.g. [6], [9].

### 2.3.2 Contributions on the Leader Election Problem

We have two contributions on the leader election problem.

1. We have developed a 2-round leader election protocol (Thm. 3.5). The previously best result by Fineman et al. [10] was  $O(\log n)$  rounds.
2. We have explored trade-offs between transmission power and communication complexity for the leader election problem. Specifically, we have shown that in order to elect a leader in  $t$  rounds, a power range  $\exp(n^{1/\Theta(t)})$  is sufficient and necessary (Thm. 5.3).

## 2.4 Broadcast Problem

In this thesis, we focus on the All-to-All Broadcast problem in a single-hop network. We define a set  $\mathcal{M}_v$  to be the set of all possible messages for each node  $v \in V$ . For any two nodes  $u, v$ ,  $\mathcal{M}_u \cap \mathcal{M}_v = \emptyset$ . We define  $\mathcal{M} = \bigcup_v \mathcal{M}_v$  to be the set of all possible messages.

**Problem 2.2** (All-to-All Broadcast). *Given that each node  $v$  in the network starts with some arbitrary message in  $\mathcal{M}_v$ , each node  $v$  is to output a set that contains the messages of all nodes in  $V$ .*

### 2.4.1 Prior Work on the Broadcast Problem

There have been plenty of papers on the broadcast problem concerning both (i) *global broadcast* [4],[23] and (ii) *local broadcast* [12],[13],[19],[22].

Note that a key difference between the problem we are solving and these problems is that both global and local broadcast assume a multi-hop network, where nodes are not able to reach every other node in absence of interference.

The global broadcast problem, as defined by [23] is the following:

**Problem 2.3** (Global Broadcast). *Given  $k$  distinct messages stored at  $k$  arbitrary nodes, one message stored in each node, disseminate all messages to the entire network.*

An early paper on global broadcast on *radio networks* was by Bar-Yehuda et al. [4]. They described an algorithm that works in  $O(k \log n \log \Delta + (D+n/\log n) \log n \log \Delta)$  rounds in expectation, where  $k$  is the number of messages to be delivered to all  $n$  nodes,  $D$  is the network diameter and  $\Delta$  is the maximum node degree. In the

SINR model, Yu et al. [23] describe the best result known that achieves the dissemination of  $k$  messages, stored in  $k$  arbitrary nodes to the entire network in  $O((D + k) \log n + \log^2 n)$  rounds w.h.p., where  $D$  is the network diameter.

The local broadcast problem, as defined by [12] is the following:

**Problem 2.4** (Local Broadcast). *Given  $n$  distinct messages stored in  $n$  different nodes in a network, ensure that every node  $v$  has successfully received a message by all nodes in  $B_v$ .*

Note that in multi-hop networks, the broadcasting range  $B_v$  of node  $v$  is the maximum distance from which  $v$  can receive a clear transmission, assuming no other transmission occurs.

Local broadcast in the SINR model was first studied by Goussevskaia et al. in [12]. The paper presents a randomized asynchronous algorithm that achieves local broadcast in  $O(\Delta \log^3 n)$  rounds. More recently, in the SINR model, local broadcast was achieved in  $O(\Delta \log n)$  [13] with knowledge of  $\Delta$  and in  $O(\Delta \log n + \log^2 n)$  rounds without it [18]. Halldórsson et al. [19] present a randomized algorithm that achieves local broadcast within  $O(\Delta + \log n)$  rounds w.h.p. using collision detection and acknowledgments. Yu et al. [22] present a lower bound for local broadcast of  $\Omega(\Delta + \log n)$  w.h.p..

The All-to-All Broadcast problem in a single-hop network is similar to the above two problems. When  $k = n$  and the network is single-hop, the Global Broadcast problem is essentially the All-to-All Broadcast problem. In a single-hop network, Local Broadcast is essentially All-to-All Broadcast, because the broadcasting range

of a node contains the whole network.

## 2.4.2 Contributions on the Broadcast Problem

The currently best result in the area is by Halldórsson et al. [19], which achieves local broadcast within  $O(\Delta + \log n)$  rounds w.h.p. using collision detection and acknowledgments. We developed an all-to-all broadcast algorithm that achieves the same asymptotical complexity in a single-hop network using weaker assumptions:

We developed an all-to-all broadcast algorithm that uses power control, and terminates in  $2n$  rounds (Thm. 7.9).

# Chapter 3

## Achieving Leader Election in Two Rounds

In this Chapter, we present a 2-round leader election algorithm, Algorithm *2-LE*, that uses power control and requires no knowledge of  $n$ . First, we give some key ideas behind the algorithm. Then, we present a 2-round leader election algorithm followed by the analysis.

### 3.1 The Essence of the Algorithm

Below we present a high level description of the key ideas behind *2-LE*.

- (i) **Breaking symmetry:** Each node  $v$  computes a geometric random variable  $k$ . Then,  $v$  picks an ID uniformly at random from a range that depends on  $k$ . With high enough probability the largest ID is picked by exactly one node.
- (ii) **The loudest node wins:** Each broadcasting node  $v$  determines its trans-

mission power by evaluating power function  $f(ID) := P_{min} \cdot ID^{\gamma ID}$  using its identification number,  $ID_v$ . Any listening nodes receive the message sent by the node with the highest ID, with a high enough probability.

- (iii) **Feedback:** The set of nodes  $V$  is split into *listeners* and *competitors*. The competitors compete for the leader position during the first round of the two-round protocol. The listeners inform the competitors of the winner during the second round.

### 3.2 Leader Election Algorithm, 2-LE

Initially, node  $v$  flips a coin (a Bernoulli random variable) to determine its *role*, which is a *competitor* if heads are flipped, and *listener* if tails. It then computes a geometric random variable (r.v.)  $k_v$ , which counts the tails flipped in a sequence of coin flips before the first heads is flipped. The ID of the node,  $ID_v$ , is an integer selected uniformly at random from the range  $[J, 2 \cdot J]$ , where  $J = g(k_v) := 2^{k_v} k_v^4$ . The power  $P_v$  that  $v$  uses for broadcast is given by  $f(ID_v) := P_{min} \cdot ID_v^{\gamma ID_v}$ , where  $P_{min}$  is the minimum power needed to reach all nodes in the network (overcoming the ambient noise). As defined in Chapter 2,  $\gamma$  is a constant that the nodes know, and  $\gamma \geq \max(1, c\alpha + 1 + \log \beta)$ .

During round 1, each competitor  $v$  transmits its ID using power  $P_v$ , which is intended to be received by the listeners. In round 2, the roles are reversed, as each listener reports back the ID of the purported leader that it received.

We shall argue that, with high probability, a unique competitor succeeds in transmitting to all the listeners, and a unique listener succeeds in reporting back to all

the competitors. The leader is then that successful competitor.

### 3.2.1 Pseudocode for Algorithm 2- $LE_v$

---

**Algorithm 1** - 2- $LE_v$  : 2-Round Leader Election Algorithm for node  $v$

---

```

1: Preprocessing:
2: Role: a boolean Bernoulli( $\frac{1}{2}$ ) random variable {'competitor' if heads, 'listener'
   if tails}
3: k: a Geometric( $\frac{1}{2}$ ) random variable,  $k \in \mathbb{Z}_{\geq 0}$ 
4: ID: chosen uniformly at random from  $[J, 2 \cdot J]$ , where  $J = g(k) := 2^k k^4$ ,  $ID \in \mathbb{Z}_{\geq 0}$ 
5: P: the transmission power,  $P = f(ID) := P_{min} \cdot (ID)^{\gamma ID}$ ,  $P \in \mathbb{Z}_{\geq 0}$ 
6: Leader: a string denoting the identity of the leader, initially empty
7:
8: Round 1:
9: if Role = competitor then
10:   Broadcast ID using power P
11: else
12:   Listen
13:   if  $v$  receives message  $m$  then
14:     Set Leader =  $m$ 
15:
16: Round 2:
17: if Role = competitor then
18:   Listen
19:   if  $v$  receives message  $m$  then
20:     Set Leader =  $m$ 
21: else
22:   Broadcast Leader using power P

```

---



### 3.3 Analysis of Algorithm 2-LE

We show that the highest power used by a competitor is sufficient to overpower all the other competitors, ensuring that this competitor is heard by all the listeners. Identical arguments hold for the reporting back in round 2.

To this end, we first show that there is a competitor whose geometric r.v. is nearly  $\log n$ , and at most a logarithmic number of competitors have that large value. We then show that all the  $O(\log n)$  IDs at the high end of the spectrum are unique, i.e., selected by a single node. The difference in power used by nodes with different ID ensures that the competitor with highest ID will overpower all the other competitors and be heard by all the listeners.

The following version of Chernoff bounds is needed.

**Theorem 3.1** (Chernoff Bound). *Let  $X_1, X_2, \dots, X_n$  be independent Bernoulli random variables and  $X = \sum_{i=1}^n X_i$ . For  $R \geq 4\mathbb{E}[X]$ ,*

$$\Pr[X \geq R] \leq 2^{-0.55R} .$$

*Proof.* The standard Chernoff bound is that for any  $\delta > 0$ ,

$$\Pr[X \geq (1 + \delta)\mathbb{E}[X]] \leq \left( \frac{e^\delta}{(1 + \delta)^{1+\delta}} \right)^{\mathbb{E}[X]} .$$

Set  $\delta$  be such that  $R = (1 + \delta)\mathbb{E}[X]$ , so  $\delta \geq 3$ . Thus,

$$\Pr[X \geq R] \leq \left( \frac{e^\delta}{(1 + \delta)^{1+\delta}} \right)^{\mathbb{E}[X]} \leq \left( \frac{e}{1 + \delta} \right)^{(1+\delta)\mathbb{E}[X]} = 2^{-\lg((1+\delta)/e)R} ,$$

which is maximized when  $\delta$  is minimized. Finally, observe that  $\lg(4/e) \geq 0.55$ .

□

We can now begin the analysis of Algorithm 2-LE.

**Lemma 3.2.** *Let  $k^* := \log n - \log \log n - 2$ . With probability greater than  $1 - \frac{1}{n^3}$ , for at least one and at most  $32 \log n$  competitors  $v$  in Algorithm 2-LE, it holds that  $k_v \geq k^*$ .*

*Proof.* Let  $t = \lceil k^* \rceil = \lceil \log n - \log \log n - 2 \rceil$ .

Let  $A_v$  be the event that a given node  $v$  is a competitor and has  $k_v \geq t$ . In order for  $v$  to have  $k_v \geq t$ , it must have flipped heads more than  $t$  times, that is  $\Pr[A_v] = \sum_{i=t}^{\infty} \frac{1}{2^i}$ . Let  $s = \sum_{i=t}^{\infty} \frac{1}{2^i}$ .

$$\begin{aligned} 2s &= \frac{2}{2^t} + \frac{2}{2^{t+1}} + \frac{2}{2^{t+2}} \dots \\ 2s &= \frac{1}{2^{t-1}} + s + 0 \\ s &= 2^{1-t} \end{aligned}$$

The probability of  $A_v$  is  $\Pr[A_v] = 2^{1-t}$ . As  $k^* \leq t \leq k^* + 1$ ,  $2^{-k^*} \leq 2^{1-t} \leq 2^{1-k^*}$ .

Thus,

$$2^{-\log n + \log \log n + 2} \leq 2^{-k^*} \leq \Pr[A_v] \leq 2^{1-k^*} \leq 2^{1-\log n + \log \log n + 2}$$

$$\frac{4 \log n}{n} \leq \Pr[A_v] \leq \frac{8 \log n}{n} .$$

The probability that no node satisfies  $A_v$  is then at most

$$\Pr \left[ \bigwedge_v \overline{A_v} \right] \leq \left( 1 - \frac{4 \log n}{n} \right)^n \leq e^{-4 \log n} \leq n^{-5.7} ,$$

for  $n$  sufficiently large, establishing the first part of the claim.

Let  $X$  be the number of nodes  $v$  for which  $A_v$  holds. Then  $\mathbb{E}[X] \leq 8 \log n$  and by Chernoff bound (Thm. 3.1) with  $R = 32 \log n$ ,

$$\Pr[X \geq 32 \log n] \leq 2^{-0.55 \cdot 32 \log n} = n^{-17.6} ,$$

for  $n$  large enough. I.e., at most  $32 \log n$  nodes satisfy  $A_v$ , with probability greater than  $1 - \frac{1}{n^{17.6}}$ .

Combined, with probability at least  $1 - \frac{1}{n^3}$ , both of these claimed events hold. □

We can now show that if some competitor  $v$  picked  $k_v > k^*$ , a sole competitor receives the highest ID with probability greater than  $1 - \frac{1}{8n}$ .

**Lemma 3.3.** *In the first round of Algorithm 2-LE, a sole competitor receives the highest ID with probability greater than  $1 - \frac{1}{8n}$ , given that at least one node calculated  $k_v \geq k^*$ .*

*Proof.* The ranges of IDs assigned to nodes of different  $k$  values are disjoint. The competitor receiving the highest ID will therefore necessarily be one with a highest  $k$  value, denoted by  $K$ . Let  $Z$  be the set of competitors with

$$k = K \geq k^* (= \log n - \log \log n - 2).$$

By Lemma 3.2,  $Z$  is non-empty and contains at most  $32 \log n$  nodes.

The range from which competitors in  $Z$  pick their IDs is  $[J, 2J]$ , for  $J \geq g(k^*) = 2^{k^*} (k^*)^4 \geq \frac{n \cdot \log^3 n}{8}$ .

The probability that a given pair of nodes in  $Z$  receive the same ID is inversely proportional to the range of IDs sampled from, or

$$1/J \leq \frac{1}{g(k^*)} \leq \frac{8}{n \cdot \log^3 n}.$$

The probability that some pair of nodes in  $Z$  are assigned the same ID is then, by the union bound, at most

$$\frac{\binom{|Z|}{2}}{J} \leq \frac{(32 \log n)^2}{\frac{n \cdot \log^3 n}{8}} = \frac{32^2}{n \log n} < \frac{1}{8n},$$

for large enough  $n$ . In particular, all nodes in  $Z$  receive different IDs with probability greater than  $1 - \frac{1}{8n}$ .

□

The highest ID received,  $ID_w$ , is at least  $g(k^*) \geq n$ , for sufficiently large values of  $n$  with probability at least  $1 - \frac{1}{n^3}$ . We can now proceed to show that if a sole competitor receives the highest ID, its transmission is received by all the listeners.

**Lemma 3.4.** *In the first round of Algorithm 2-LE, if a sole competitor receives the highest ID ( $ID > n$ ), its transmission is received by all the listeners.*

*Proof.* Let  $w$  be the sole competitor with the highest ID. For any other competitor  $v$  it then holds that

$$\frac{P_w}{P_v} \geq \frac{f(ID_w)}{f(ID_w - 1)} \geq ID_w^\gamma \geq n^\gamma \geq \beta n^{c\alpha+1}. \quad (3.1)$$

Let  $u$  be a listener. The noise and interference received by  $u$  can be bound in terms of the signal

$$S_u := P_w/d(w, u)^\alpha$$

it receives from  $w$ . Recall that  $d(w, u) \leq R \cdot d(v, u) \leq n^c \cdot d(v, u)$ , and thus

$$d(w, u)^\alpha \leq n^{c\alpha} \cdot d(v, u)^\alpha,$$

for any competitor  $v$ . Hence, applying (3.1), the interference received from a competitor  $v$  is bounded by

$$I_v := \frac{P_v}{d(v, u)^\alpha} \leq \frac{P_w \cdot n^{c\alpha}}{\beta n^{c\alpha+1} \cdot d(w, u)^\alpha} = \frac{S_u}{\beta n}. \quad (3.2)$$

The definition of minimum power  $P_{min}$  ensures that  $\frac{P_{min}/d(w, u)^\alpha}{N} \geq \beta$ . Thus, inequality (3.1) can be used to bound the noise term by

$$N \leq \frac{P_{min}}{d(w, u)^\alpha \cdot \beta} \leq \frac{P_w}{d(w, u)^\alpha \cdot n^\gamma \cdot \beta} = \frac{S_u}{\beta n^\gamma} \leq \frac{S_u}{\beta n}. \quad (3.3)$$

Combining (3.2) and (3.3), the SINR of  $w$ 's signal at receiver  $u$  is bounded below by

$$\frac{S_u}{N + \sum_{v \in X} I_v} \geq \frac{\beta n}{1 + |X|} \geq \beta,$$

where  $X$  is the set of competitors other than  $w$ . Thus,  $w$  overpowers all other competitors at all the listeners.  $\square$

**Theorem 3.5.** *Algorithm 2-LE terminates with all nodes agreeing on a common leader, w.h.p.*

*Proof.* Using a union bound, we add up the error probabilities of Lemmas 3.2 and

3.3, and find that a sole competitor  $w$  receives the highest ID, with probability at least  $1 - \frac{1}{4n}$ . By Lemma 3.4,  $w$  then successfully informs all the receivers.

All three lemmas work identically for the reporting process in round 2. Hence, with probability at least  $1 - \frac{1}{2n}$ , the algorithm succeeds.

We conclude that Algorithm 2-*LE* terminates successfully in 2 rounds with probability at least  $1 - \frac{1}{n}$ , and thus Algorithm 2-*LE* terminates with high probability. □

**Remark 3.6.** *Leader election can be achieved in a single round if simultaneous transmission and reception is possible. Such full-duplex radios operate by subtracting the transmitted signal from the received one. While they are still rare, being hard to implement, such technology has been progressing significantly in recent years ([5], [8]) and may well become a commodity feature. With full-duplex, the arguments apply unchanged to the success of reception by the other competitors, thus succeeding after only a single round.*

# Chapter 4

## Upper Bound on the Power Needed for a 2-Round Leader Election

### Algorithm

In this Chapter, we present upper bounds on the power needed for a 2-round leader election algorithm when the nodes don't know  $n$ , and when the nodes do.

We define  $P_{max} \in \mathbb{R}_{\geq 1}$  to be a *power bound* for an algorithm  $A$  provided that in all executions of  $A$ , the power with which any node transmits is at most  $P_{max}$ .

When the nodes don't know  $n$ , leader election can be achieved w.h.p. using

$$P_{max} \geq 2^{\tilde{O}(n^2)} P_{min}.$$

When the nodes do know  $n$ , leader election can be achieved w.h.p. using

$$P_{max} \geq 2^{\tilde{O}(n)} P_{min}.$$

## 4.1 Upper Bound on the Power Needed with no knowledge of $n$

Note that Algorithm  $2-LE$  does not have a finite power bound. In order to answer how the power bound must grow as a function of  $n$  for leader election to work correctly, we define algorithm  $2-LE(P_{max})$  which has a power bound  $P_{max}$ . We will show that algorithm  $2-LE(P_{max})$  works w.h.p. for

$$P_{max} \geq 2^{\tilde{O}(n^2)} P_{min}.$$

### 4.1.1 Algorithm $2-LE(P_{max})$

Below we present algorithm  $2-LE(P_{max})_v$ . The only difference between algorithms  $2-LE_v$  and  $2-LE(P_{max})_v$  is that in  $2-LE(P_{max})_v$  node  $v$  automatically truncates its assigned power to at most  $P_{max}$ .



---

**Algorithm 2** -  $2\text{-}LE(P_{max})_v$  : 2-Round Leader Election Algorithm for node  $v$  given  $P_{max}$

---

- 1: **Preprocessing:**
  - 2:  $Role$ : a boolean  $Bernoulli(\frac{1}{2})$  random variable {‘*competitor*’ if heads, ‘*listener*’ if tails}
  - 3:  $k$ : a  $Geometric(\frac{1}{2})$  random variable,  $k \in \mathbb{Z}_{\geq 0}$
  - 4:  $ID$ : chosen uniformly at random from  $[J, 2 \cdot J]$ , where  $J = g(k) := 2^k k^4$ ,  $ID \in \mathbb{Z}_{\geq 0}$
  - 5:  $P$ : the transmission power,  $P = f(ID) := \mathbf{min}(P_{max}, P_{min} \cdot (ID)^{\gamma_{ID}})$ ,  $P \in \mathbb{Z}_{\geq 0}$
  - 6:  $Leader$ : a string denoting the identity of the leader, initially empty
  - 7:
  - 8: **Round 1:**
  - 9: **if**  $Role = competitor$  **then**
  - 10:   Broadcast  $ID$  using power  $P$
  - 11: **else**
  - 12:   Listen
  - 13:   **if**  $v$  receives message  $m$  **then**
  - 14:     Set  $Leader = m$
  - 15:
  - 16: **Round 2:**
  - 17: **if**  $Role = competitor$  **then**
  - 18:   Listen
  - 19:   **if**  $v$  receives message  $m$  **then**
  - 20:     Set  $Leader = m$
  - 21: **else**
  - 22:   Broadcast  $Leader$  using power  $P$
-

### 4.1.2 Analysis of Algorithm 2-LE( $P_{max}$ )

**Theorem 4.1.** *Assume power bound  $P_{max} \geq 2^{10n^2(\log n)^2} P_{min}$ . Then, Algorithm 2-LE( $P_{max}$ ) elects a leader w.h.p..*

*Proof.* Fix  $P_{max} \geq 2^{10n^2(\log n)^2} P_{min}$ .

The probability  $p$  that Algorithm 2-LE( $P_{max}$ ) elects a leader is greater than the probability  $p'$  that the Algorithm 2-LE( $P_{max}$ ) elects a leader *and* no node truncates its transmission power. Now,  $p'$  is essentially the probability that the non-truncating Algorithm 2-LE from Section 3.2 elects a leader *and* no node picks a transmission power that exceeds  $P_{max}$ .

The probability that Algorithm 2-LE( $P_{max}$ ) elects a leader,  $p$ , is then greater or equal to the probability that Algorithm 2-LE elects a leader,  $p_{LE}$ , minus the probability that some node picks a transmission power that exceeds  $P_{max}$ ,  $p_P$ :

$$p \geq p_{LE} - p_P$$

By the proof of Theorem 3.5, we know that Algorithm 2-LE terminates with all nodes agreeing on a common leader with probability greater than  $1 - 1/2n$ . Thus,

$$p_{LE} \geq 1 - 1/2n.$$

It remains to determine  $p_P$ . A node  $v$  can only pick a transmission power greater than  $P_{max}$  if it picks  $k_v \geq 2 \log n + 2$ . The probability that  $v$  picks such a transmission power is  $2^{1-2 \log n - 2} = \frac{1}{2n^2}$ . The probability that no node picks  $k \geq 2 \log n + 2$  is  $(1 - \frac{1}{2n^2})^n$ . By Bernoulli's inequality,  $(1 - \frac{1}{2n^2})^n \geq 1 - \frac{1}{2n}$ . Thus,  $p_P \leq \frac{1}{2n}$ .

Thus,

$$p \geq p_{LE} - p_P \geq 1 - \frac{1}{2n} - \frac{1}{2n} \geq 1 - \frac{1}{n}$$

We conclude that Algorithm 2- $LE(P_{max})$  elects a leader w.h.p..

□

## 4.2 Assuming $n$ , a Smaller Power Bound is Achievable

Assuming the nodes know  $n$ , leader election can be achieved with a smaller power bound,

$$P_{max} \geq 2^{\tilde{O}(n)} P_{min}.$$

### 4.2.1 Algorithm 2- $LEn$

Below we present a leader election algorithm, 2- $LEn_v$ , that achieves leader election in two rounds w.h.p. using  $P_{max} \geq 2^{\tilde{O}(n)} P_{min}$ .

The algorithm begins with each node  $v$  flipping an unfair coin to determine if it will participate in the election. Specifically, node  $v$  participates in the election with probability  $4 \log n/n$ . Then, each participating node flips a fair coin to determine if it will be a competitor or a listener, and then selects an  $ID$  uniformly at random from the range  $[J, 2J]$ , where  $J = n \log^2 n$ . The power used is  $f(ID_v)$  as in Algorithm 2- $LE$ , and the algorithm rounds are the same as well.

The highest possible ID is  $2n \log^2 n$ . This results in  $P_{max} \geq (2n \log^2 n)^{2n \log^2 n} P_{min} \geq 2^{\tilde{O}(n)} P_{min}$ .

---

**Algorithm 3** -  $2-LEn_v$  2-Round Leader Election Algorithm for node  $v$  with knowledge of  $n$

---

1: *Role*: a string  
2: *ID*: an identification number,  $ID \in \mathbb{Z}_{\geq 0}$   
3: *P*: the transmission power,  $P \in \mathbb{Z}_{\geq 0}$   
4: *Leader*: a positive integer number denoting the identity of the leader, initially 0  
5:  
6: *Participating*: a boolean  $Bernoulli(\frac{4 \log n}{n})$  random variable  
7:  
8: **if** *Participating* = 1 **then**  
9:     *Role* = a boolean  $Bernoulli(\frac{1}{2})$  random variable {‘*competitor*’ if heads, ‘*listener*’ if tails}  
10:     *ID*: choose uniformly at random from  $[J, 2 \cdot J]$ , where  $J = n \log^2 n$   
11:      $P = f(ID) = P_{min} \cdot (ID)^{\gamma ID}$   
12:  
13:     **Round 1:**  
14:     **if** *Role* = *competitor* **then**  
15:         Broadcast *ID* using power *P*  
16:     **else**  
17:         Listen  
18:         **if**  $v$  receives message  $m$  **then**  
19:             Set *Leader* =  $m$   
20:  
21:     **Round 2:**  
22:     **if** *Role* = *competitor* **then**  
23:         Listen  
24:         **if**  $v$  receives message  $m$  **then**  
25:             Set *Leader* =  $m$   
26:     **else**  
27:         Broadcast *Leader* using power *P*

---

### 4.2.2 Analysis of Algorithm 2-LEn

**Theorem 4.2.** *Algorithm 2-LEn elects a leader w.h.p..*

*Proof.* The probability that two nodes participate in the algorithm and they pick the same ID is

$$\left(\frac{4 \log n}{n} \cdot \frac{1}{J}\right)^2 = \left(\frac{4 \log n}{n} \cdot \frac{1}{n \log^2 n}\right)^2 = \left(\frac{4}{n^2 \log n}\right)^2.$$

The probability that any two nodes participate in the algorithm and they pick the same ID is

$$\binom{n}{2} \left(\frac{4}{n^2 \log n}\right)^2 \leq \frac{16}{n^2 \log^2 n}.$$

We need to show that there will be at least one competitor and at least one listener with a high enough probability. The probability that there are no competitors is

$$\left(1 - \frac{4 \log n}{2n}\right)^n \leq e^{-2 \log n} \leq \frac{1}{n^2}.$$

Similarly, the probability that there are no listeners is at most  $\frac{1}{n^2}$ .

We can now use arguments similar to Lemma 3.4, to show that in the first round of Algorithm 2-LEn, when a sole competitor has the highest ID, its transmission is received by all the listeners.

Similarly for the second round of Algorithm 2-LEn, when a sole listener has the highest ID, its transmission is received by all the competitors.

Thus, Algorithm 2-*LEn* elects a leader with probability greater than

$$1 - \frac{16}{n^2 \log^2 n} - \frac{1}{n^2} - \frac{1}{n^2} \geq 1 - \frac{1}{n}.$$

□

# Chapter 5

## Leader Election: Trading Time for Power

In this chapter, we explore how much the power can be reduced by increasing the round complexity, while still achieving a successful leader election protocol. When only a smaller power bound  $P_{max} < 2^{\tilde{O}(n^2)} P_{min}$  is available, we can still elect a leader using a larger number of rounds with Algorithm  $2t\text{-}LE(P_{max})$ .

### 5.1 Algorithm $2t\text{-}LE(P_{max})$

Our multi-round algorithm  $2t\text{-}LE(P_{max})_v$  simply repeats the 2-round algorithm  $t$  times, for a given  $t \geq 1$ , but uses a slower-growing power function. Namely, node  $v$  changes its ID-selection function to  $g_t(k_v) = 2^{k_v} k_v^{3t+1}$ , and its power function to  $f_t(ID_v) = \min(P_{max}, P_{min} \cdot ID_v^{\gamma(ID_v)^{1/t}})$ . After each round-pair, each competitor  $v$  updates its  $leader_v$  value to the *largest* among those heard so far. After some round-pair when all nodes have the same *leader* value, *leader* will remain the same.

---

**Algorithm 4** -  $2t$ -LE( $P_{max}$ ) $_v$  :  $2t$ -Round Leader Election Algorithm for node  $v$ 

---

1: *Role*: a string  
2:  $k \in \mathbb{Z}_{\geq 0}$   
3: *ID*: an identification number,  $ID \in \mathbb{Z}_{\geq 0}$   
4:  $P$ : the transmission power,  $P \in \mathbb{Z}_{\geq 0}$   
5: *leader*: a positive integer number denoting the identity of the leader, initially 0  
6:  
7: **for**  $r$  in range  $[1, \dots, t]$  **do**  
8:   *Role* = a boolean *Bernoulli*( $\frac{1}{2}$ ) random variable {‘*competitor*’ if heads, ‘*listener*’ if tails}  
9:    $k$  = a *Geometric*( $\frac{1}{2}$ ) random variable  
10:   Choose *ID* uniformly at random from  $[J, 2 \cdot J]$ , where  $J = g_t(k) = 2^k k^{3t+1}$   
11:    $P = f_t(ID) = \min(P_{max}, P_{min} \cdot ID^{\gamma(ID)^{1/t}})$   
12:  
13:   **Round 1:**  
14:   **if** *Role* = *competitor* **then**  
15:     **if** *leader* is 0 **then**  
16:       Broadcast *ID* using power  $P$   
17:     **else**  
18:       Broadcast *leader* using power  $P$   
19:     **else**  
20:       Listen  
21:       **if**  $v$  receives message  $m$  **then**  
22:         Set *leader* =  $m$   
23:  
24:     **Round 2:**  
25:     **if** *Role* = *competitor* **then**  
26:       Listen  
27:       **if**  $v$  receives message  $m$  **then**  
28:         Set *leader* =  $\max(m, leader)$   
29:     **else**  
30:       Broadcast *leader* using power  $P$

---



## 5.2 Analysis of Algorithm $2t$ -LE( $P_{max}$ )

First, observe that it suffices to succeed in one of the round-pairs. A round-pair is successful if by the end of it all nodes have the same value in the *leader* variable.

**Lemma 5.1.** *Algorithm  $2t$ -LE( $P_{max}$ ) elects a leader, if, after some round-pair, all nodes have the same value in the leader variable.*

*Proof.* Suppose that after some round-pair, all nodes have *leader's* value set as  $w$ .

In any future round-pairs, all broadcasts use  $w$  for the value of *leader*. Thus, there can't be any inconsistencies in any future round-pairs that lead to different nodes having different values in *leader*. The algorithm will conclude will all nodes having *leader's* value set as  $w$ , and thus  $w$  will be the leader. □

For a single round-pair to be successful, it is required that the highest transmission power is unique and sufficiently larger than the second highest. In Algorithm  $2$ -LE it sufficed to show that the highest ID was unique to satisfy this condition. Because here we use a slower growing power function, we need to show that the  $t^{\text{th}}$  root of the highest ID is larger than the  $t^{\text{th}}$  root of the second highest ID + 1 with a high enough probability.

**Lemma 5.2.** *In a given round of Algorithm  $2t$ -LE( $P_{max}$ ), with probability at least  $1 - \frac{1}{3n^{1/t}}$ , some unique node  $w$  has an ID such that  $(ID_w)^{1/t} - (ID_v)^{1/t} > 1$ , for all other nodes  $v$ .*

*Proof.* Let  $Z$  be the set of broadcasting nodes with the largest  $k$  value. The node  $v$  with the highest transmission power will be in  $Z$ . Using a similar argument to

Lemma 3.2, we can show that  $k_v \geq k^* = \log n - \log \log n - 2$ , and that  $Z$  has at least one and at most  $32 \log n$  nodes with probability greater than  $1 - 1/n^3$ .

Recall that IDs are allocated uniformly at random, and for nodes in  $Z$ , the range is of size at least

$$\begin{aligned} g_t(k^*) &= 2^{\log n - \log \log n - 2} (\log n - \log \log n - 2)^{3t+1} \\ &\geq \frac{n}{8 \log n} \left( \frac{\log n}{2} \right)^{3t+1} \end{aligned}$$

for large enough  $n$ . Thus,  $g_t(k^*) \geq \frac{1}{2^{3t+1}} n \log^{3t} n$ .

The probability that a given pair of nodes  $u, v$  in  $Z$  receive nearly equivalent IDs, with  $|(ID_u)^{1/t} - (ID_v)^{1/t}| \leq 1$ , is at most

$$g_t(k^*)^{-1/t} \leq \frac{2}{n^{1/t} \log^3 n}.$$

Thus, the probability that some two nodes in  $Z$  receive nearly equivalent IDs ( $|(ID_u)^{1/t} - (ID_v)^{1/t}| \leq 1$ ) is at most

$$\frac{\binom{|Z|}{2}}{g_t(k^*)^{1/t}} \leq \frac{2 \cdot 32^2 \log^2 n}{n^{1/t} \log^3 n} < \frac{1}{3n^{1/t}},$$

for sufficiently large  $n$ .

Thus, with probability at least  $1 - \frac{1}{3n^{1/t}}$ , some unique node  $w$  has an ID such that  $(ID_w)^{1/t} - (ID_v)^{1/t} > 1$ , for all other nodes  $v$ .

□

We can now show that Algorithm  $2t$ -LE( $P_{max}$ ) elects a leader w.h.p..

**Theorem 5.3.** *Algorithm  $2t$ -LE( $P_{max}$ ), with  $P_{max} \geq 2^{10n^{2/t}(\log n)^2} P_{min}$ , elects a leader w.h.p., for each number  $t = O(\log n / \log \log n)$ .*

*Proof.* Fix  $P_{max} \geq 2^{10n^{2/t}(\log n)^2} P_{min}$ .

It suffices to show that at least one round-pair is successful w.h.p. (Lemma 5.1).

Let's focus on one round pair  $r$ .

First, we can show that no truncation occurs with probability greater than  $1 - \frac{1}{4n}$ . In order for a node  $v$  to pick  $P_{max}$ , it needs to pick some  $k_v \geq 2 \log n + 3$ . This occurs with probability  $\frac{1}{4n^2}$ . The probability that no node picks  $k \geq 2 \log n + 3$  is

$$\left(1 - \frac{1}{4n^2}\right)^n \geq 1 - \frac{1}{4n}.$$

Now, we can calculate the probability that round pair  $r$  is successful.

Let's look at the first round of  $r$ . In Lemma 5.2 we showed that in a given round, with probability at least  $1 - \frac{1}{3n^{1/t}}$ , some unique node  $w$  has an ID such that

$$(ID_w)^{1/t} > (ID_v)^{1/t} + 1,$$

for all other nodes  $v$ .

As a unique node  $w$  has an ID such that  $(ID_w)^{1/t} > (ID_v)^{1/t} + 1$ , this node also has the highest transmission power and using similar arguments to Lemma 3.4, we can show that with probability at least  $1 - \frac{1}{3n^{1/t}}$ , node  $w$ 's message is received by all listening nodes.

A similar argument holds for the second round of  $r$ . We can use Lemmas 5.2 and 3.4 to show that in this round with probability at least  $1 - \frac{1}{3n^{1/t}}$ , some node  $w$ 's

message is received by all listening nodes.

Thus, round-pair  $r$  is successful with probability greater than  $1 - \frac{1}{3n^{1/t}} - \frac{1}{3n^{1/t}} - \frac{1}{4n} \geq 1 - \frac{1}{n^{1/t}}$ .

We conclude that the probability that *all*  $t$  round-pairs are unsuccessful is

$$\left(\frac{1}{n^{1/t}}\right)^t = 1/n,$$

as desired. □

# Chapter 6

## Lower Bound on the Power Needed for a $t$ -Round Leader Election

### Algorithm

In this chapter we show that an exponential-size power bound is necessary for any leader election protocol running in  $t$  rounds. More specifically, any  $t$ -round leader election algorithm in the SINR model running correctly w.h.p. requires

$$P_{max} \geq 2^{\sqrt[t]{n}/(12e)} P_{min},$$

$t \geq 1$ . Our proofs holds for  $\beta > 2$ .

First, we determine how much power is needed to break the symmetry of the network, and then we prove a lower bound on the power needed by any  $t$ -round leader election algorithm.

## 6.1 Power Needed to Break the Symmetry of a Network

First, we need the definition of a uniform metric space.

**Definition 6.1** (Uniform Metric Space). *A uniform metric space is a metric space, where for all nodes  $(u, v) \in V$ ,  $d(u, v) = d$ , for some  $d \in \mathbb{R}_{>0}$ .*

Uniform metric spaces have the following convenient property:

**Lemma 6.2.** *Assume an SINR network is located in a uniform metric space. In a given round either a single message is received by all the listeners or none of them hear anything.*

*Proof.* Define  $T$  as the set of all transmitting nodes. If  $T$  is empty, the lemma is clearly true. Otherwise, we can define  $w$  to be the transmitting node with the highest transmission power.

Let  $L = T \cap V$  be the set of listeners. Suppose that node  $v \in L$ . Let  $I = T - \{w\}$ . The SINR equation for  $v$  given a signal from  $w$  is

$$SINR(w, v, I) = \frac{\frac{P_w}{d(w,v)^\alpha}}{N + \sum_{u \in I} \frac{P_u}{d(u,v)^\alpha}} = \frac{\frac{P_w}{d^\alpha}}{N + \sum_{u \in I} \frac{P_u}{d^\alpha}}.$$

As no node in  $L$  can be in  $I$ , the SINR equation  $SINR(w, v, I)$  is the same for all nodes  $v \in L$ . Thus, if one listener receives a message, all listeners receive a message.  $\square$

We can now prove the following claim.

**Lemma 6.3.** *Let  $A$  be an algorithm with a power bound  $P_{max} = 2^q P_{min}$ , running on  $n$  nodes located in a uniform metric space in a single-hop SINR network, where  $q$  is a nonnegative integer. Given a round when all nodes are in identical states at the start of the round, the nodes will remain in identical states with probability at least  $\frac{1}{24e \cdot q}$ , for integer  $q \geq 1$ , and with at least constant probability for  $q = 0$ .*

*Proof.* By Lemma 6.2, in a given round of  $A$  either a single message is received by all the listeners or none of them hear anything.

In order to break the symmetry of the network, a message has to be successfully received.

We divide the available range of power into subranges, each within factor 2. Specifically, the  $i$ -th highest subrange is

$$(P_{max}/2^i, P_{max}/2^{i-1}],$$

where  $i \in [1, \dots, q]$ .

Let  $X_i^v$  be the event that node  $v$  transmits using the  $i$ -th highest subrange. For example,  $X_2^v$  is the event that node  $v$  transmits using the second highest subrange, that is node  $v$  transmits with some power in  $(P_{max}/4, P_{max}/2]$ .

Since the nodes are identical, the same probability holds for them all, so let  $p_i = \Pr[X_i^v]$ . Let  $q \geq 0$  be the largest number such that

$$\sum_{i=1}^q p_i \leq \frac{1}{2n} . \tag{6.1}$$

Now, we have two cases to consider, either  $q = 0$ , or  $q \geq 1$ .

**Case 1:**  $q = 0$

Suppose that  $q = 0$ . Then,  $p_1 > \frac{1}{2n}$ . That is, any node  $u$  chooses to transmit using a transmission power in the highest subrange with probability greater than  $\frac{1}{2n}$ .

Thus, the probability that any two particular nodes  $u, w$  choose to transmit using a transmission power in the highest subrange  $(P_{max}/2, P_{max}]$  is

$$\binom{n}{2} p_1^2 > \binom{n}{2} \frac{1}{4n^2} = \frac{n(n-1)}{8n^2} \approx \frac{1}{8}.$$

Suppose node  $v$  is listening and nodes  $u, w$  are transmitting with some power from the highest subrange. Let  $P_w > P_u$ . Note that  $P_u > P_w/2$ . Then,

$$SINR(w, v, I) = \frac{\frac{P_w}{d(w,v)^\alpha}}{N + \sum_{u \in I} \frac{P_u}{d(u,v)^\alpha}} < \frac{\frac{P_w}{d(w,v)^\alpha}}{\frac{P_u}{d(u,v)^\alpha}} < \frac{P_w}{P_u} < 2 < \beta.$$

As  $SINR(w, v, I) < \beta$ , no message is received. Thus, with at least constant probability, no message is received, which results in the nodes remaining in identical states.

**Case 2:**  $q \geq 1$

Now, let's look at the case when  $q \geq 1$ .

Let  $A_i$  be the event that at least two nodes use the  $i$ -th highest subrange  $(P_{max}/2^i, P_{max}/2^{i-1}]$ ,  $B_i$  be the event that no node transmits at subranges  $1, 2, \dots, i-1$ , and  $C_i = A_i \cap B_i$  be the event that both  $A_i$  and  $B_i$  occur, for  $i = 1, 2, \dots$



A subrange  $i$  is in use if at least one node broadcasts with transmission  $P \in (P_{max}/2^i, P_{max}/2^{i-1}]$ . The subrange  $i$  is highest subrange in use if no subrange  $j$  is in use, for  $j \in 1, 2, \dots, i-1$ .

Then,  $C = \bigcup_i C_i$  is the event that at least two nodes use the highest subrange in use.

Observe that  $\Pr[A_i|B_i] \geq \Pr[A_i]$ , since the non-use of the  $i-1$  highest subranges only makes the event  $A_i$  more likely. Then,

$$\Pr[C_i] = \Pr[A_i \cap B_i] = \Pr[A_i|B_i] \Pr[B_i] \geq \Pr[A_i] \Pr[B_i] .$$

The probability of  $A_i$ ,  $i \leq q$ , can be bounded by the first term of the binomial expansion:

$$\Pr[A_i] > \binom{n}{2} p_i^2 (1-p_i)^{n-2} > \frac{n^2}{3} p_i^2 \left(1 - \frac{1}{2n}\right)^{n-2} > \frac{n^2}{3e} p_i^2 .$$

Also, applying (6.1),

$$\Pr[B_i] \geq 1 - n \sum_{j=1}^{i-1} p_j \geq \frac{1}{2} .$$

Observe that the  $C_i$ 's are mutually exclusive and apply the Cauchy-Schwarz inequality followed by (6.1) to obtain:

$$\Pr[C] \geq \sum_{i=1}^q \Pr[C_i] \geq \frac{n^2}{3e} \sum_{i=1}^q p_i^2 \cdot \frac{1}{2} \geq \frac{n^2}{6e} \frac{(\sum_{i=1}^q p_i)^2}{q} \geq \frac{1}{24e \cdot q} .$$

When  $C$  holds, at least two nodes pick a transmission from the highest subrange in use. Using a similar argument to case 1, we can show that in this case the symmetry of the network does not break.

We conclude that with probability at least  $\frac{1}{24e \cdot q}$ , the nodes remain in identical states.  $\square$

## 6.2 Lower Bound on Power Needed for a $t$ -Round Leader Election Algorithm

We can now show a lower bound on the power using Lemma 6.3.

**Theorem 6.4.** *Any  $t$ -round leader election algorithm in the SINR model running correctly w.h.p. requires  $P_{max} \geq 2^{\sqrt[t]{n}/(24e)} P_{min}$ ,  $t \geq 1$ .*

*Proof.* For the sake of contradiction, let  $A$  be a  $t$ -round leader election algorithm in the SINR model that runs correctly with probability greater than  $1 - 1/n$  with a power bound strictly smaller than  $2^{\sqrt[t]{n}/(24e)} P_{min}$ .

We first argue by induction on the number of rounds  $r$ ,  $1 \leq r \leq t$ , that the probability that all nodes are in identical states at the end of round  $r$ , is at least

$$\frac{1}{(24e \cdot q)^r},$$

for  $q < \sqrt[t]{n}/(24e)$ .

1. Base Case:

We're in the first round of the algorithm, where all nodes are initialized in identical states. By Lemma 6.3, the probability that the nodes are in identical states at the end of round 1 is at least  $\frac{1}{24e \cdot q}$ .

2. Inductive Step:  $r \geq 2$ :

Suppose that the nodes remain in identical states for the first  $r - 1$  rounds with probability at least

$$\frac{1}{(24e \cdot q)^{r-1}}.$$

We must prove that the nodes remain in identical states for  $r$  rounds with probability at least

$$\frac{1}{(24e \cdot q)^r}.$$

Given that the nodes are in identical states in the beginning of round  $r$ , by Lemma 6.3, the nodes remain in identical states at the end of round  $r$  with probability at least  $\frac{1}{24e \cdot q}$ . Thus, the nodes are in identical states after the first  $r$  rounds with probability  $\frac{1}{(24e \cdot q)^r}$ .

This concludes the induction. For  $r = t$ , all nodes remain in identical states for  $t$  rounds with probability at least

$$\frac{1}{(24e \cdot q)^t}.$$

Algorithm  $A$  elects a leader with probability at least  $1 - 1/n$ . For Algorithm  $A$  to work correctly, it's required that the nodes are not in identical states by the end of round  $t$ .

Thus, it's necessary that  $1/n \geq \frac{1}{(24e \cdot q)^t}$ . However, as  $q < \sqrt[t]{n}/(24e)$ , we reach a contradiction:

$$\frac{1}{n} \geq \frac{1}{(24e \cdot q)^t} > \left( \frac{1}{24e \cdot \frac{\sqrt[t]{n}}{24e}} \right)^t > \frac{1}{n}$$

Thus, we have shown that such an algorithm  $A$  cannot exist. It follows that any algorithm that solves the leader election problem in  $t$  rounds must have a power bound of at least  $2^{\sqrt[n]{t}/(24e)} P_{min}$ .

□

**Corollary 6.5.** *Any 2-round leader election algorithm in the SINR model running correctly w.h.p. requires  $P_{max} \geq 2^{\sqrt[n]{t}/(24e)} P_{min}$ .*

# Chapter 7

## All-to-All Broadcast with Power Control

Here we present an All-to-All Broadcast algorithm, Algorithm LeCirio<sup>1</sup>, that works with power control and takes  $2n$  rounds.

This chapter consists of two Sections. In Section 7.1, we present a variant of leader election algorithm *2-LE*, from Chapter 3, that works with probability greater than  $1 - \frac{2048}{n^2 \log n} - \frac{2}{n^3}$ , Algorithm *2-LEp*.

In Section 7.2, we present an All-to-All broadcast algorithm, Algorithm LeCirio, that uses it as a subroutine.

### 7.1 Augmented Leader Election

Algorithm *2-LE* works with probability greater than  $1 - \frac{1}{n}$ . We need an algorithm that works with higher probability to use as a subroutine in Algorithm LeCirio.

---

<sup>1</sup>This algorithm gets its name from a restaurant in Brussels that has a wonderful orange cat.

In addition, the broadcasting algorithm, Algorithm LeCirio, requires a high-probability upper bound for  $n$ . We need an algorithm that returns some value that can be used to calculate an upper bound of  $n$ .

To this end, we define the following problem.

**Problem 7.1** (Augmented Leader Election Problem). *Eventually elect exactly one node (called the leader), with all nodes knowing (i) whether or not they are the leader and (ii) some value  $k$ , such that  $k \geq \log n - \log \log n - 3$ .*

We now present Algorithm 2-LEP that solves Problem 7.1 with probability greater than  $1 - \frac{2048}{n^2 \log n} - \frac{2}{n^3}$ .

Algorithm 2-LEP differs from 2-LE in only two places, the ID selection helper function  $g$ , and the transmission of  $k$ . The nodes pick IDs uniformly at random from  $[J, 2 \cdot J]$ , where  $J = g(k) := 2^{k^3} k^5$ . We will show that this change ensures that Algorithm 2-LEP elects a leader with a much higher probability than 2-LE.

In the first round, the competing nodes broadcast their estimate of  $k$ , along with their ID, and in the second round, the listeners repeat the received estimate for  $k$ , along with the ID of the leader. This change ensures that the nodes will be able to calculate an upper bound for  $n$  during LeCirio with a high enough probability.

### 7.1.1 Pseudocode for Algorithm 2-LEP<sub>v</sub>

---

**Algorithm 5** *2-LEp<sub>v</sub>*: 2-Round Leader Election Algorithm with higher success rate

---

- 1: **Preprocessing:**
  - 2: *Role*: a boolean *Bernoulli*( $\frac{1}{2}$ ) random variable {‘*competitor*’ if heads, ‘*listener*’ if tails}
  - 3: *k*: a *Geometric*( $\frac{1}{2}$ ) random variable,  $k \in \mathbb{Z}_{\geq 0}$
  - 4: *k<sub>l</sub>*: the leader’s *k* value, initially 0,  $k_l \in \mathbb{Z}_{\geq 0}$
  - 5: *ID*: chosen uniformly at random from  $[J, 2 \cdot J]$ , where  $J = g(k) := 2^{k^3} k^5$ ,  $ID \in \mathbb{Z}_{\geq 0}$
  - 6: *P*: the transmission power,  $P = f(ID) := P_{min} \cdot (ID)^{\gamma ID}$ ,  $P \in \mathbb{Z}_{\geq 0}$
  - 7: *Leader*: a string denoting the identity of the leader, initially empty
  - 8:
  - 9: **Round 1:**
  - 10: **if** *Role* = *competitor* **then**
  - 11:   Broadcast *ID*, and *k* using power *P*
  - 12: **else**
  - 13:   If *v* receives message *m*, set *Leader* to the received *ID*, set *k<sub>l</sub>* to the received *k*
  - 14:
  - 15: **Round 2:**
  - 16: **if** *Role* = *competitor* **then**
  - 17:   If *v* receives message *m*, set *Leader* to the received *ID*, set *k<sub>l</sub>* to the received *k*
  - 18: **else**
  - 19:   Broadcast *Leader*, and the received *k* using power *P*
-

### 7.1.2 Analysis of Algorithm 2-LEp

Let  $k^* := \log n - \log \log n - 2$ . Using a similar argument to Lemma 3.2, we can show that for at least one and at most  $32 \log n$  competitors  $v$ , it holds that  $k_v \geq k^*$  with a high enough probability.

**Lemma 7.2.** *Let  $k^* := \log n - \log \log n - 2$ . With probability greater than  $1 - \frac{1}{n^3}$ , for at least one and at most  $32 \log n$  competitors  $v$  in Algorithm 2-LEp, it holds that  $k_v \geq k^*$ .*

The proof of Lemma 7.2 is the same as the proof of Lemma 3.2, as a node  $v$  calculates  $k_v$  the in the exact same way whether it is running Algorithm 2-LEp or 2-LE.

We show that exactly one node receives the highest ID with probability greater than  $1 - \frac{1024}{n^2 \log n}$  (similarly to Lemma 3.3).

**Lemma 7.3.** *In the first round of Algorithm 2-LEp, a sole competitor receives the highest ID with probability greater than  $1 - \frac{1024}{n^2 \log n}$ , given that at least one node calculated  $k_v \geq k^*$ .*

*Proof.* The ranges of IDs assigned to nodes of different  $k_v$  values are disjoint. The competitor receiving the highest ID will therefore necessarily be one with a highest  $k_v$  value, which we denote by  $K$ .

Let  $Z$  be the set of competitors with

$$k_v = K \geq k^* (= \log n - \log \log n - 2).$$



By Lemma 7.2,  $Z$  is non-empty and contains at most  $32 \log n$  nodes. The probability that a given pair of nodes in  $Z$  receive the same ID is inversely proportional to the range of IDs sampled from, or  $1/J$ . The probability that some pair of nodes in  $Z$  are assigned the same ID is then, by the union bound, at most

$$\frac{\binom{|Z|}{2}}{J} \leq \frac{(32 \log n)^2}{(2^{\log n})^2 \log^3 n} = \frac{1024}{n^2 \log n},$$

for large enough  $n$ .

In particular, all nodes in  $Z$  receive different IDs with probability greater than  $1 - \frac{1024}{n^2 \log n}$ .

□

Given that a sole competitor  $v$  received the highest ID, we can show that if  $v$  broadcasts, all listeners will receive its message.

**Lemma 7.4.** *In the first round of Algorithm 2-LEP, if a sole competitor receives the highest ID ( $ID > n$ ), its transmission is received by all the listeners.*

We can prove Lemma 7.4 using similar arguments to the ones in the proof of Lemma 3.4.

All that is left to show is that the whole algorithm works with probability greater than  $1 - \frac{2048}{n^2 \log n} - \frac{2}{n^3}$ .

**Theorem 7.5.** *Algorithm 2-LEP terminates with all nodes agreeing on a common leader and knowing some value  $k \geq \log n - \log \log n - 2$ , with probability greater than  $1 - \frac{2048}{n^2 \log n} - \frac{2}{n^3}$ .*

*Proof.* Using a union bound, we add up the error probabilities of Lemmas 7.2 and 7.3, and find that a sole competitor  $w$ , calculates  $k_w \geq \log n - \log \log n - 2$  and receives the highest ID, with probability at least  $1 - \frac{1024}{n^2 \log n} - \frac{1}{n^3}$ . Then, by Lemma 7.4,  $w$  successfully informs all the receivers of its ID and  $k_w$ . All three lemmas work identically round 2. Hence, with probability at least  $1 - \frac{2048}{n^2 \log n} - \frac{2}{n^3}$ , the algorithm succeeds. □

## 7.2 All-to-All Broadcast Algorithm

In this Section, we present our all-to-all broadcast algorithm LeCirio, which concludes successfully in  $2n$  rounds w.h.p., that is, with probability greater than  $1 - 1/n$ .

### 7.2.1 The LeCirio Algorithm

Algorithm  $LeCirio_v$  has two phases.

The *first phase* consists of the two rounds needed for the nodes to run Algorithm 2-LEP, which elects a leader, and to pick transmission powers. This leader acquires the *Leader* position and stays around for the whole algorithm.

The *second phase* consists of  $(n - 1)$  two-round timeslots, which is  $2n - 2$  rounds. In the first round of each timeslot, the leader node listens and all other competing nodes broadcast. If no message is received, the leader assumes that no one else wants to broadcast, and drops out.

In the second round, the leader broadcasts the message it received in the last round, acknowledging it. All other nodes listen. If a node receives its own message, it drops out. The messages are distinct, so a node drops out only if it receives its

own message. Each node  $v$  keeps a set  $out_v$  that contains all the messages it has received and its own message.

Note that in the above algorithm, all nodes but the *leader* node  $v$  have a chance to broadcast. In order to ensure that  $v$  also has a chance to broadcast, it will append its message to the message it repeats in the second round of Phase 2, so that all nodes receive it.

More specifically:

**Phase 1** This phase takes two rounds.

In Phase 1, each node  $v$  runs Algorithm 2-*LEP*, to determine whether it is the leader node, and uses the leader's  $k$  value,  $k_l$ , to calculate  $N = 2^{2k_l - 6}$ . Node  $v$  picks its  $ID_v$  uniformly at random from  $[N, N^4 + N]$ , and its transmission power  $P = f(ID_v) = P_{min} \cdot (ID_v)^{\gamma ID_v}$ .

**Phase 2** The second phase takes  $2n - 2$  rounds. Let *rounds* denote how many rounds have passed in Phase 2, and  $msg_v$  be what node  $v$  wants to broadcast.

If *rounds* is odd:

If  $v$  is the leader node, it listens. If it receives a message, it sets variable  $lastmsg_v$  to the received message. Otherwise, it drops out.

If  $v$  is not the leader node, it broadcasts  $msg_v$ .

If *rounds* is even:

If  $v$  is the leader node and  $rounds = 2$ , it broadcasts  $lastmsg_v$  and  $msg_v$ . Otherwise, it broadcasts just  $lastmsg_v$ .

If  $v$  is not the leader node, it listens. If it receives a packet that is  $msg_v$  or contains  $msg_v$ , it drops out.

---

**Algorithm 6** *LeCirio<sub>v</sub>*: All-to-All Broadcast Algorithm with Power Control

---

- 1:  $ID$ , initially 0,  $ID \in \mathbb{Z}_{\geq 0}$
- 2:  $P$ : the transmission power,  $P = P_{min}$ ,  $P \in \mathbb{Z}_{\geq 0}$
- 3: *Leader*: a string denoting the identity of the leader, initially empty
- 4:  $N$ , initially 1,  $N \in \mathbb{Z}_{\geq 1}$
- 5:  $k_l$ : the leader's  $k$  value, initially 0,  $k_l \in \mathbb{Z}_{\geq 0}$
- 6:  $lastmsg \in \mathcal{M} \cup \perp$ , initially  $\perp$
- 7:  $msg \in \mathcal{M}_v$
- 8: *out*, a set, initially contains just  $msg$
- 9:
- 10: **Phase 1:**
- 11: Run Algorithm 2-*LEp*, setting the *Leader* and  $k_l$  variables
- 12: Choose  $ID$  uniformly at random from  $[N, N^4 + N]$ , where  $N = 2^{2k_l - 6}$
- 13: Set  $P = f(ID) = P_{min} \cdot (ID)^{\gamma ID}$
- 14:
- 15: **Phase 2:**
- 16: **for** *round* in  $[1, 2, \dots, 2n - 2]$  **do**
- 17:   **if**  $v = Leader$  **then**
- 18:     **if** *round* mod 2=0 **then**
- 19:       **if** *round* = 2 **then**
- 20:          Broadcast a packet containing both *lastmsg* and *msg*
- 21:       **else**
- 22:          Broadcast *lastmsg*
- 23:     **else**
- 24:       Listen
- 25:       **if** a packet is received **then**
- 26:          Set *lastmsg* to that packet, add *lastmsg* to *out*
- 27:       **else**
- 28:          Drop out
- 29:     **if**  $v \neq Leader$  **then**
- 30:       **if** *round* mod 2=0 **then**
- 31:          Listen, adding any received messages to *out*
- 32:       **if** a packet is received that contains *msg* **then**
- 33:          Drop out and keep listening, adding any received messages to *out*
- 34:       **else**
- 35:          Broadcast *msg*

---

## 7.2.2 Analysis of LeCirio

We show that LeCirio terminates after each node has communicated its information to everyone in  $2n$  rounds w.h.p..

To this end, we first show that by the end of Phase 1 a leader is elected and all nodes have unique IDs larger than  $n$ , and then show that all nodes broadcast their messages successfully in Phase 2, with a high enough probability.

By Lemma 7.5 we know that Algorithm 2-LEP terminates with all nodes agreeing on a common leader and knowing some value  $k \geq \log n - \log \log n - 2$ , with probability greater than  $1 - \frac{2048}{n^2 \log n} - \frac{2}{n^3}$ .

We now show that by the end of Phase 1, the nodes have unique IDs greater than  $n$  with a high enough probability.

**Lemma 7.6.** *Algorithm LeCirio's Phase 1 terminates with all nodes having unique IDs greater than  $n$ , with probability greater than  $1 - 1/n^2$ , given that Algorithm 2-LEP terminated successfully.*

*Proof.* If Algorithm 2-LEP terminated successfully, a leader  $l$  was elected, and all nodes know  $k_l \geq \log n - \log \log n - 2$ . Each node  $v$  calculates  $N = 2^{2k_l - 6}$ , which is a good upper bound for  $n$ :

$$N = 2^{2k_l - 6} > 2^{2 \log n - 2 \log \log n} = \left( \frac{n}{\log n} \right)^2.$$

Given such  $N$ , there is a collision when picking IDs with probability

$$\binom{n}{2} \frac{1}{N^4} \leq \frac{n^2}{N^4} \leq \frac{1}{n^2}.$$

Thus, the nodes have unique IDs with probability greater than  $1 - 1/n^2$ , and all IDs are greater than  $n$ .

□

We can now combine Lemmas 7.5 and 7.6 to show that LeCirio's Phase 1 is successful with a high enough probability.

**Lemma 7.7.** *By the end of Algorithm LeCirio's Phase 1, a leader is elected and all nodes have unique IDs larger than  $n$  with probability greater than  $1 - 2/n^2$ .*

*Proof.* By Lemma 7.5 we know that Algorithm 2-LEP terminates with all nodes agreeing on a common leader and knowing some value  $k \geq \log n - \log \log n - 2$ , with probability greater than  $1 - \frac{2048}{n^2 \log n} - \frac{2}{n^3}$ .

By Lemma 7.6, we know that given the successful termination of Algorithm 2-LEP, all nodes pick unique IDs greater than  $n$  with probability greater than  $1 - 1/n^2$ .

Using a union bound, we can conclude that by the end of Algorithm LeCirio's Phase 1, a leader is elected and all nodes have unique IDs larger than  $n$  with probability greater than  $1 - 2/n^2$ .

□

Now, we show that in any round of Phase 2 of Algorithm LeCirio, all listeners receive the message sent by the broadcasting node with the highest ID.

**Lemma 7.8.** *In any round of Phase 2 of Algorithm LeCirio, if a sole broadcasting node has the highest ID ( $ID > n$ ), then its transmission is received by all the listeners.*

*Proof.* Suppose that the nodes are executing Algorithm LeCirio, and they are on round  $i$  of Phase 2. If  $i$  is even, only the leader node is broadcasting, thus the claim holds.

It remains to show that the claim holds when  $i$  is odd. We assume that some broadcasting node has the highest ID ( $ID > n$ ). Because the nodes calculate their transmission power using the same functions as in Algorithm *2-LEP*, we can use similar arguments as the ones in the proof of Lemma 7.4 to show that all listeners receive the message sent by the node with the highest ID.

Thus, in any round of Phase 2 of Algorithm LeCirio, if a sole broadcasting node has the highest ID ( $ID > n$ ), its transmission is received by all the listeners.

□

It remains to show that Algorithm LeCirio completes an all-to-all broadcast in  $2n$  rounds with probability greater than  $1 - 1/n$ .

**Theorem 7.9.** *Algorithm LeCirio completes an all-to-all broadcast in  $2n$  rounds with probability greater than  $1 - 1/n$ .*

*Proof.* According to Lemma 7.7, Phase 1 terminates with the election of a leader, and with all nodes having unique IDs greater than  $n$ , with probability greater than  $1 - 2/n^2$ .

No messages in  $\mathcal{M}$  are sent during Phase 1, so all messages are sent during Phase 2. We ensure that both the leader node and the non-leader nodes broadcast their message successfully.

### **Leader node**

The leader node sends its message during the second round of Phase 2, and since it's the only one broadcasting, its message is received by all nodes.

### **Non-leader nodes**

Let's assume that all nodes have unique IDs greater than  $n$ .

The non-leader nodes attempt to broadcast their messages in odd rounds during Phase 2. In any odd round, according to Lemma 7.8, some node  $v$ 's message is received by all listeners, which includes the leader node. In the next round the leader node repeats  $v$ 's message ensuring that all nodes receive it. After a node receives a message it sent, it drops out. The messages are distinct, and no node can drop out as a result of receiving someone else's message.

Every two rounds of Phase 2, some node drops out as every other node has received its message. Since there are only  $n - 1$  nodes, after  $2n - 2$  rounds of Phase 2, each non-leader node has successfully sent its message to every other node.

We conclude that Algorithm LeCirio takes  $2n$  rounds, and terminates after all nodes have sent their messages to every other node with probability greater than



$1 - 2/n^2 \geq 1 - 1/n$ . Thus, every node  $v$  has in its  $out_v$  set a message from every node in  $V$  w.h.p..

□

**Remark 7.10.** *It might be possible to achieve an even more efficient broadcast algorithm than LeCirio by piggybacking messages. In Algorithm LeCirio, every other round a message is simply acknowledged. Instead of just acknowledging a message, it might possible to send a new message as well. This is an interesting topic for future work.*

# Chapter 8

## Contributions

In this Masters thesis, we showed how the SINR model allows for more efficient algorithms in certain problems than the traditional radio network model. This is due to the new capabilities that the SINR model gives us, power control and the capture effect.

On the leader election problem, we are able to achieve a two communication round solution, where  $O(\log n)$  rounds were the previously best known result. We explored some trade-offs between communication complexity and transmission power for the leader election algorithm.

We achieved an efficient algorithm for the all-to-all broadcast problem. We developed an all-to-all broadcast algorithm that uses power control, and terminates in  $2n$  rounds w.h.p..

### 8.1 Future Work

Some interesting directions for the future are the following:

1. It would be interesting to explore how to make efficient algorithms using power control in other problems, like coloring, or MIS.
2. The presented leader election algorithm only works in a single-hop network. It would be exciting to see what speed-up power control might be able to give in a multi-hop setting.
3. There is a gap between our upper bound on the power range (Theorem 4.1), and our lower bound (Theorem 6.4). It would be interesting to find a more efficient algorithm and/or a tighter lower bound.
4. One could explore how to make a more efficient broadcast algorithm than LeCirio by piggybacking messages.

# Bibliography

- [1] Norman Abramson. The ALOHA system: Another alternative for computer communications. In *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference, AFIPS '70 (Fall)*, pages 281–285, New York, NY, USA, 1970. ACM.
- [2] Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast in radio networks: An exponential gap between determinism randomization. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC '87*, pages 98–108, New York, NY, USA, 1987. ACM.
- [3] Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast in radio networks: an exponential gap between determinism randomization. In *PODC*, pages 98–108. ACM, 1987.
- [4] Reuven Bar-Yehuda, Amos Israeli, and Alon Itai. Multiple communication in multihop radio networks. *SIAM J. Comput.*, 22(4):875–887, August 1993.
- [5] Dinesh Bharadia, Emily McMillin, and Sachin Katti. Full duplex radios. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, SIGCOMM '13*, pages 375–386, New York, NY, USA, 2013. ACM.
- [6] S. A. Borbash and A. Ephremides. Wireless link scheduling with power control and sinr constraints. *IEEE Transactions on Information Theory*, 52(11):5106–5111, Nov 2006.
- [7] Imrich Chlamtac and Shay Kutten. On broadcasting in radio networks—problem analysis and protocol design. *IEEE Transactions on Communications*, 33(12):1240–1246, 1985.
- [8] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. Achieving single channel, full duplex wireless communication. In *Proceedings*

of the Sixteenth Annual International Conference on Mobile Computing and Networking, MobiCom '10, pages 1–12, New York, NY, USA, 2010. ACM.

- [9] R. L. Cruz and A. V. Santhanam. Optimal routing, link scheduling and power control in multihop wireless networks. In *INFOCOM*, pages 702–711, vol.1, 2003.
- [10] Jeremy T Fineman, Seth Gilbert, Fabian Kuhn, and Calvin Newport. Contention resolution on a fading channel. In *PODC*, pages 155–164. ACM, 2016.
- [11] Robert G Gallager. A perspective on multiaccess channels. Technical report, DTIC Document, 1985.
- [12] Olga Goussevskaia, Thomas Moscibroda, and Roger Wattenhofer. Local broadcasting in the physical interference model. In *Proceedings of the Fifth International Workshop on Foundations of Mobile Computing*, DIALM-POMC '08, pages 35–44, New York, NY, USA, 2008. ACM.
- [13] Olga Goussevskaia, Thomas Moscibroda, and Roger Wattenhofer. Local broadcasting in the physical interference model. In *Proceedings of the Fifth International Workshop on Foundations of Mobile Computing*, pages 35–44. ACM, 2008.
- [14] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.
- [15] Magnús M. Halldórsson, Stephan Holzer, and Evangelia Anna Markatou. Brief announcement: Leader election in SINR model with arbitrary power control. In *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 203–205, 2017.
- [16] Magnús M. Halldórsson, Stephan Holzer, and Evangelia Anna Markatou. Leader election in SINR model with arbitrary power control. In *Structural Information and Communication Complexity - 24th International Colloquium, SIROCCO 2017, Porquerolles, France, June 19-22, 2017, Revised Selected Papers*, pages 3–14, 2017.
- [17] Magnús M Halldórsson and Pradipta Mitra. Wireless capacity with oblivious power in general metrics. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1538–1548. SIAM, 2011.

- [18] Magnus M Halldorsson and Pradipta Mitra. Towards tight bounds for local broadcasting. In *Proceedings of the 8th International Workshop on Foundations of Mobile Computing*, page 2. ACM, 2012.
- [19] Magnus M. Halldórsson, Tigran Tonoyan, Yuexuan Wang, and Dongxiao Yu. Brief announcement: Data dissemination in unified dynamic wireless networks. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 199–201, New York, NY, USA, 2016. ACM.
- [20] Thomas Moscibroda and Roger Wattenhofer. The complexity of connectivity in wireless networks. In *INFOCOM*, 2006.
- [21] Calvin C. Newport. Radio network lower bounds made easy. In *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, pages 258–272, 2014.
- [22] D. Yu, Q. S. Hua, Y. Wang, and F. C. M. Lau. An  $o(\log n)$  distributed approximation algorithm for local broadcasting in unstructured wireless networks. In *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*, pages 132–139, May 2012.
- [23] Dongxiao Yu, Qiang-Sheng Hua, Yuexuan Wang, Jiguo Yu, and Francis CM Lau. Efficient distributed multiple-message broadcasting in unstructured wireless networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 2427–2435. IEEE, 2013.