The Energy Complexity of Las Vegas Leader Election

Yi-Jun Chang*
cyijun@nus.edu.sg
National University of Singapore
Singapore

Shunhua Jiang
sj3005@columbia.edu
Columbia University
New York, USA

ABSTRACT

We consider the time (number of communication rounds) and energy (number of non-idle communication rounds per device) complexities of randomized leader election in a multiple-access channel, where the number of devices $n \geq 2$ is *unknown*. It is well-known that for polynomial-time randomized leader election algorithms with success probability $1 - 1/\mathrm{poly}(n)$, the optimal energy complexity is $\Theta(\log\log^* n)$ if receivers can detect collisions, and it is $\Theta(\log^* n)$ otherwise.

Without collision detection, all existing randomized leader election algorithms using $o(\log\log n)$ energy are *Monte Carlo* in that they might fail with some small probability, and they might consume unbounded energy and never halt when they fail. Although the optimal energy complexity of leader election *appears* to have been settled, it is still an intriguing open question whether it is possible to attain the optimal $O(\log^* n)$ energy complexity by an efficient *Las Vegas* algorithm that never fails. In this paper we address this fundamental question.

- **A separation between Monte Carlo and Las Vegas algorithms:** Without collision detection, we prove that any Las Vegas leader election algorithm $\mathcal{A}$ with *finite* expected time complexity must use $\Omega(\log\log n)$ energy, establishing a large separation between Monte Carlo and Las Vegas algorithms. Our lower bound is *tight*, matching the energy complexity of an existing leader election algorithm that finishes in $O(\log n)$ time and uses $O(\log\log n)$ energy in expectation.
- **An exponential improvement with sender collision detection:** In the setting where transmitters can detect collisions, we design a new leader election algorithm that finishes in $O(\log^{1+\epsilon} n)$ time and uses $O(\epsilon^{-1}\log\log\log n)$ energy in expectation, showing that sender collision detection helps improve the energy complexity *exponentially*. Before this work, it was only known that sender collision detection is helpful for *deterministic* leader election.
- **An optimal deterministic leader election algorithm:** As a side result, via derandomization, we show a new deterministic leader election algorithm that takes $O\left(n\log\frac{N}{n}\right)$ time and $O\left(\log\frac{N}{n}\right)$ energy to elect a leader from n devices, where each device has a unique identifier in $[N]$. The algorithm is *simultaneously* time-optimal and energy-optimal, matching existing $\Omega\left(n\log\frac{N}{n}\right)$ time lower bound and $\Omega\left(\log\frac{N}{n}\right)$ energy lower bound.

CCS CONCEPTS

• **Theory of computation → Distributed algorithms**.

KEYWORDS

leader election, radio network, Las Vegas algorithms

ACM Reference Format:
Yi-Jun Chang and Shunhua Jiang. 2022. The Energy Complexity of Las Vegas Leader Election. In *Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '22), July 11–14, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3490148.3538586

1 INTRODUCTION

Leader election is one of the most central problems of distributed computing. In a network of an unknown number n of devices communicating via a shared communication channel, the goal of leader election is to have exactly one device in the network identify itself as the *leader*, and all other devices identify themselves as *non-leaders*.

Leader election has a wide range of applications, as it captures the classic *contention resolution* problem, where several processors need temporary and exclusive access to a shared resource. Leader election is also used to solve the *wake-up* problem [23, 34], whose the goal is to wake-up all processors in a completely connected broadcast system, in which an unknown number of processors are awake spontaneously and they have to wake-up the remaining sleeping processors.

We focus on *single-hop* networks (all devices communicating via a shared communication channel) in the *static* scenario (all devices start at the same time). Leader election protocols in single-hop networks are used as communication primitives in algorithms for more sophisticated distributed tasks in multi-hop networks [6, 13]. Leader election protocols in the static setting are useful building blocks in the design of contention resolution protocols in the *dynamic* setting where the devices have different starting time [8, 9] by batch processing.

1.1 The Multiple-access Channel Model

In our model, an unknown number n of devices connect to a *multiple-access channel*. The communication proceeds in synchronous rounds and all devices have an agreed-upon time zero. In each communication round, a device may choose to transmit a message, listen to the channel, or stay idle.

If more than one device simultaneously transmit a message in a round, then a *collision* occurs. Listeners only receive messages

*Supported by NUS ODPRT Grant No. R-252-000-C04-133.

This work is licensed under a Creative Commons Attribution International 4.0 License.

SPAA '22, July 11–14, 2022, Philadelphia, PA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9146-7/22/07.
https://doi.org/10.1145/3490148.3538586

from collision-free transmissions. There are four variants [12] of the model based on the collision detection ability of transmitters (distinguishing between successful transmission and collision) and listeners (distinguishing between silence and collision).

- Strong-CD. Transmitters and listeners receive one of the three feedback: (i) silence, if zero devices transmit, (ii) collision, if at least two devices transmit, or (iii) a message m, if exactly one device transmits.
- Sender-CD. Transmitters and listeners receive one of the two feedback: (i) silence, if zero or at least two devices transmit, or (ii) a message m, if exactly one device transmits.
- Receiver-CD. Transmitters receive no feedback. Listeners receive one of the three feedback: (i) silence, if zero devices transmit, (ii) collision, if at least two devices transmit, or (iii) a message m, if exactly one device transmits.
- No-CD. Transmitters receive no feedback. Listeners receive one of the two feedback: (i) silence, if zero or at least two devices transmit, or (ii) a message m, if exactly one device transmits.

We distinguish between randomized and deterministic models. In the randomized setting, the devices are *anonymous* in that they do not have unique identifiers and run the same algorithm, but they may break symmetry using their private random bits. In the deterministic setting, each device is initially equipped with a unique identifier from an ID space $[N]$, where N is *global knowledge*. Unless otherwise stated, we assume that the number of devices $n \geq 2$ is *unknown*.

The goal of leader election is to have exactly one device in the network identify itself as the leader, and all other devices identify themselves as non-leaders. We require that the communication protocol ends when the leader sends a message while every non-leader listens to the channel, so all devices terminate in the same round.

Complexity measures. Traditionally, the leader election problem has been studied from the context of optimizing the *time complexity*, which is defined as the number of communication rounds needed to solve the problem. More recently, there has been a growing interest [9, 11, 12, 15, 25, 26, 30, 32] in the *energy complexity* of leader election, which is defined as the maximum number of *non-idle* rounds per device, over all devices. That is, each transmitting or listening round costs one unit of energy. The study of energy complexity is motivated by the fact that many small mobile battery-powered devices are operated under a limited energy constraint. These devices may save energy by turning off their transceiver and entering a low-power sleep mode. As a large fraction of energy consumption of these devices are often spent on sending and receiving packets, the energy complexity of an algorithm approximates the actual energy usage of a device. In applied research, *idle listening* (transceiver is active but no data is received) has been identified as a major source of energy inefficiency in *wireless sensor networks*, and there is a large body of work studying strategies for minimizing the number of transmission and idle listening [31, 35, 37, 39].

1.2 Prior Work

For the time complexity of leader election, Willard [38] showed that expected $\Theta(\log\log n)$ time is necessary and sufficient for leader election in Receiver-CD. More generally, Nakano and Olariu [33] showed that the optimal time complexity of leader election in Receiver-CD is $\Theta(\log\log n + \log f^{-1})$ if the maximum allowed failure probability is f.

For the case an upper bound $n_{\max} \geq n$ on the unknown network size n is *known* to all devices, it is well-known that leader election can be solved using the *decay* algorithm of Bar-Yehuda, Goldreich, and Itai [6] in worst-case $O(\log n_{\max} \log f^{-1})$ time with success probability $1-f$ [6, 23, 27] in No-CD. The algorithm simply tries the transmission probability 2^{-i} for $O(f^{-1})$ times, for all integers $1 \leq i \leq \log n_{\max}$ until a successful transmission occurs. On the lower bound side, Jurdziński and Stachowiak [27] showed an $\Omega\left(\frac{\log n_{\max} \log f^{-1}}{\log\log n_{\max} + \log\log f^{-1}}\right)$ time lower bound. Later, Farach-Colton, Fernandes, and Mosteiro [21] showed the tight $\Omega(\log n_{\max} \log f^{-1})$ time lower bound for *oblivious* algorithms, in which there is a fixed sequence of transmission probabilities $(p_1, p_2, \ldots)$ such that if there has been no collision-free transmission, then all devices transmit with the same probability p_i in the ith round, using fresh randomness independently. Based on a technique of Alon, Bar-Noy, Linial, and Peleg [1], Newport [34] showed an $\Omega(\log^2 n_{\max})$ time lower bound for the case $f = 1/\text{poly}(n_{\max})$ that applied to all algorithms. Very recently, the time complexity of leader election in which the algorithm is provided an arbitrary distribution of the possible network sizes n was studied in [24].

For the energy complexity of leader election algorithms, Lavault, Marckert, and Ravelomanana [30] designed a leader election algorithm that finishes in expected $O(\log n)$ time and uses expected $O(\log\log n)$ energy in No-CD. Its expected time complexity matches the $\Omega(\log n)$ lower bound for expected time of Newport [34]. Subsequently, the algorithm of [30] was applied to finding an estimate $\widetilde{n}$ of n [36].

After a sequence of research [9, 11, 12, 25, 26], it is now known [12] that for polynomial-time randomized leader election algorithms with success probability $1 - 1/\text{poly}(n)$, the optimal energy complexity is $\Theta(\log\log^* n)$ if listeners can detect collisions (Strong-CD and Receiver-CD), and it is $\Theta(\log^* n)$ otherwise (Sender-CD and No-CD).

The energy complexity has also been studied in multi-hop networks [5, 7, 10, 13, 14, 17, 19, 20, 22, 29]. Optimization problems related to energy efficiency in multi-hop networks were considered in [3, 4, 28].

1.3 Monte Carlo and Las Vegas Algorithms

Although the optimal energy complexity of leader election *appears* to have been settled due to the work of [12], we observe that several existing randomized leader election protocols, including the ones in [12], are *Monte Carlo* in that they might fail with some small probability, and they might consume unbounded energy and never halt when they fail. In particular, without collision detection (No-CD), all existing randomized leader election algorithms [12, 26] using $o(\log\log n)$ energy have this issue, and it is not even known if these algorithms have *finite* expected time complexity.

It remains as an intriguing open question whether it is possible to attain the optimal $O(\log^* n)$ energy complexity proved in [12] by an efficient *Las Vegas* algorithm that never fail.

It is tempting to guess that we might be able to transform existing Monte Carlo leader election algorithms into Las Vegas algorithms without worsening the asymptotic time and energy complexities by too much, but designing such a transformation is actually very challenging when the number of devices n is *unknown*.

To explain the issue, consider the following simple Monte Carlo leader election protocol that finishes in $O(\log^2 n)$ time with probability $1 - 1/\text{poly}(n)$ when it is run on a network of n devices. For $i = 1, 2, \ldots$, the ith iteration consists of $C \cdot i$ rounds, where $C > 0$ is some constant. In each round each device v transmits with probability 2^{-i}. All devices that are not transmitters listen to the channel, so a leader is elected once the number of transmitting devices is exactly one in a round.

Let $i^* = \lfloor \log n \rfloor$. If the number of devices is n, then the success probability in each round in the i^*th iteration is $\Omega(1)$, implying that a leader is elected by the i^*th iteration with probability $1 - n^{-\Omega(C)}$. Hence the algorithm finishes in $\sum_{j=1}^{i^*} C \cdot i = O(\log^2 n)$ time with probability $1 - n^{-\Omega(C)}$. The *expected* time complexity of this protocol is however *infinite* because in an extremely unlucky event that a leader is not elected within the first $O(\log n)$ iterations, with high probability the protocol will run forever.

A natural attempt to fix the issue of infinite expected time complexity is to restart the protocol when it fails, but this strategy does not work as there is no mechanism for a device to detect that the algorithm has already failed! Since the number of devices n is unknown, we are not able to set a time limit $T(n)$ and restart the protocol once the number of rounds exceeds $T(n)$ does not work, as the devices cannot calculate $T(n)$ if n is not known.

The issue is even more serious if there is no collision detection. In the No-CD model, each transmitter does not know if the message is successfully transmitted and each listener cannot distinguish between collision and silence, so there is no way for a device to learn anything about the number of devices n given that no successful transmission (the number of transmitters is exactly one and the number of listeners is at least one) occurs.

1.4 New Results

The main objective of this paper is to better understand the strange gap between Monte Carlo and Las Vegas complexities of leader election discussed above. We focus on the following fundamental question: Is it possible to attain the optimal $O(\log^* n)$ energy bound proved in [12] by an efficient *Las Vegas* algorithm in the No-CD model?

A separation between Monte Carlo and Las Vegas algorithms. Surprisingly, we show that for any leader election algorithm $\mathcal{A}$ with a *finite* expected time complexity in the No-CD model, it is necessary that $\mathcal{A}$ uses $\Omega(\log\log n)$ energy, establishing a large separation between Monte Carlo and Las Vegas algorithms and answering the above question in the negative.

Theorem 1.1 (Energy lower bound for Las Vegas algorithms)**.** *Let $\mathcal{A}$ be a randomized leader election algorithm in the* No-CD *model. We write $T(n)$ and $E(n)$ to denote the expected time and energy complexities of $\mathcal{A}$. Suppose there is some integer $n^* \geq 2$ such the expected time $T(n^*)$ of the algorithm $\mathcal{A}$ when running on $n = n^*$ devices is finite. Then there exist infinite number of network sizes n such that $E(n) = \Omega(\log\log n)$.*

Our lower bound is very strong in that the energy lower bound $\Omega(\log\log n)$ holds even if there is just *one* network size n^* such that the algorithm $\mathcal{A}$ has *finite* expected time complexity when it is run on a network of n^* devices. Even allowing exponential time, our lower bound still rules out the possibility of having a Las Vegas leader election algorithm in No-CD that uses $o(\log\log n)$ energy.

Our lower bound is *tight* in that it matches the energy complexity of the existing Las Vegas leader election algorithm of [30]: There is a leader election algorithm that finishes in time $O(\log n)$ and energy $O(\log\log n)$ in expectation in the No-CD model. As the expected time complexity $O(\log n)$ is already optimal due to the lower bound $\Omega(\log n)$ in [34], our result implies that the algorithm of [30] is simultaneously time-optimal and energy-optimal.

An exponential improvement with sender collision detection. We design a new leader election algorithm in the Sender-CD model that finishes in $O(\log^{1+\epsilon} n)$ time and uses $O(\epsilon^{-1}\log\log\log n)$ energy in expectation, giving an *exponential improvement* over the previous Las Vegas algorithm of [30], at the cost of slightly increasing the time complexity from $O(\log n)$ to $O(\log^{1+\epsilon} n)$.

Theorem 1.2 (An exponential improvement in energy complexity)**.** *For any $0 < \epsilon < 1$, there is an algorithm $\mathcal{A}$ in the* Sender-CD *model that elects a leader in expected $O(\log^{1+\epsilon} n)$ time and using expected $O(\epsilon^{-1}\log\log\log n)$ energy.*

A fundamental problem in the study of multiple-access channels is to determine the value of collision detection. It is well-known that the ability for listeners to detect collision is very helpful in the design of randomized leader election algorithm in that the ability to distinguish between collision and silence allows the devices to perform an exponential search to estimate the network size n efficiently [38].

Prior to this work, existing results suggested that the ability for transmitters to detect collision does not seem to help in the randomized setting. Indeed, for polynomial-time Monte Carlo leader election algorithms, it was shown in [12] that $\Theta(\log^* n)$ is a tight energy bound in No-CD and Sender-CD and $\Theta(\log\log^* n)$ is a tight energy bound in Receiver-CD and Strong-CD, so it appears that the ability for transmitters to detect collision does not matter.

Our result shows that the ability for transmitters to detect collision helps improve the energy complexity exponentially for Las Vegas algorithms, giving the first example showing that the ability for transmitters to detect collision is valuable in the design of randomized algorithms.

We summarize our results for Las Vegas leader election algorithms in Table 1.

An optimal deterministic leader election algorithm. Recently, a systematic study of time-energy tradeoffs for deterministic leader election was done in [15]. Due to the result of [15], for the three models Strong-CD, Receiver-CD, and Sender-CD, tight or nearly tight time and energy bounds in terms of the number of devices n and the size of ID space N were known for deterministic leader election.

The last missing piece in the puzzle is the No-CD model, where the current best deterministic No-CD algorithm takes $O(N)$ time

Table 1: Old and new results on Las Vegas leader election algorithms. Here "Monte Carlo" indicates that the time and energy bounds hold with probability $1 - 1/\mathrm{poly}(n)$ and "Las Vegas" indicates that the time and energy bounds hold in expectation.

Model	Time	Energy	Type	Reference
No-CD/Sender-CD	$n^{o(1)}$	$O(\log^* n)$	Monte Carlo	[12]
	$n^{O(1)}$	$\Omega(\log^* n)$	Monte Carlo	
No-CD/Sender-CD	$O(\log n)$	$O(\log\log n)$	Las Vegas	[30]
No-CD	$\Omega(\log n)$	any	Las Vegas	[34]
No-CD	any	$\Omega(\log\log n)$	Las Vegas	Theorem 1.1
Sender-CD	$O(\log^{1+\epsilon} n)$	$O(\epsilon^{-1}\log\log\log n)$	Las Vegas	Theorem 1.2

and $O\left(\log \frac{N}{n}\right)$ energy [15] and the current best lower bounds are $\Omega\left(n \log \frac{N}{n}\right)$ time [18] and $\Omega\left(\log \frac{N}{n}\right)$ energy [15].

By a derandomization of a subroutine that we use in our randomized Las Vegas algorithms, we show an optimal deterministic No-CD algorithm that takes $O\left(n \log \frac{N}{n}\right)$ time [18] and $O\left(\log \frac{N}{n}\right)$ energy, settling the optimal complexities of leader election in the deterministic No-CD model.

Theorem 1.3 (Optimal deterministic leader election)**.** *Suppose that the size N of the ID space $[N]$ and an estimate $\widetilde{n}$ of the number of devices n such that $\widetilde{n}/2 < n \le \widetilde{n}$ are both known to all devices. There is a deterministic leader election algorithm in the* No-CD *model with time complexity $T = O\left(n \log \frac{N}{n}\right)$, energy complexity $E = O\left(\log \frac{N}{n}\right)$.*

Our deterministic algorithm requires that n is known or a constant-factor approximation of n is given. See Table 2 for a summary of results on deterministic algorithms in No-CD.

1.5 Technical Overview

In this section we overview of the key ideas behind the proofs of our results.

The $\Omega(\log\log n)$ energy lower bound. Recall from the discussion in Section 1.3 that the main source of difficulty of transforming a Monte Carlo algorithm a into a Las Vegas one in the No-CD model is that a device is unable to obtain any information from listening to the channel if no successful transmission occurs, as the feedback from the channel is always silence. The behavior of such a device depends only on its own private randomness, as it does not receive any other information. In particular, such a device cannot learn anything about n. A key idea behind our $\Omega(\log\log n)$ energy lower bound in Theorem 1.1 is to make use of this observation. The proof of Theorem 1.1 combines the following three ingredients.

- It was shown in [34] that with constant probability there is no collision-free transmission in the first $t = O(\log n)$ rounds. This means that to prove the $\Omega(\log\log n)$ energy lower bound, it suffices to show that under the condition that the channel feedback is always silence, the energy cost in the first t rounds is $\Omega(\log t) = \Omega(\log\log n)$ in expectation.
- The assumption that the given algorithm $\mathcal{A}$ has finite expected time complexity for some fixed $n = n^*$ implies that there are infinitely many t such that the probability that the algorithm does not finish by time t less than $f = 1/t$.
- By a derandomization, we may transform the $\Omega\left(\log \frac{N}{n}\right)$ deterministic energy lower bound in [15] into a randomized lower bound $\Omega\left(n^{-1}\log f^{-1}\right)$ for $f < 1/\binom{N}{n}$.

Setting $f = 1/t$ and $n = n^* = \Theta(1)$ in the randomized lower bound $\Omega\left(n^{-1}\log f^{-1}\right) = \Omega(\log t)$, we obtain that the energy cost in the first t rounds is $\Omega(\log t)$ when we run the algorithm $\mathcal{A}$ under the condition that the channel feedback is always silence. Combining this with the lower bound of [34], we obtain the desired $\Omega(\log\log n)$-energy lower bound. This lower bound argument still works even if the underlying network size n is not n^* because a device does not learn anything about n if the channel feedback is always silence.

Las Vegas Leader election algorithms. To prove Theorem 1.2, we will first design a basic subroutine that achieves the following. Given a network size estimate $\widetilde{n}$, the subroutine elects a leader in $O(\log f^{-1})$ time with $O(\widetilde{n}^{-1}\log f^{-1})$ energy in No-CD, if the number of devices n satisfies $\widetilde{n}/2 < n \le \widetilde{n}$.

Using this subroutine, we may re-establish the result of [30] that leader election can be solved in expected $O(\log n)$ time and using expected $O(\log\log n)$ energy. The leader election proceeds in iterations. During the ith iteration, we run our basic subroutine for all $\widetilde{n} = 2^1, 2^2, 2^3, \ldots, 2^{2^i-1}$ with $f = 1/4$, and then the algorithm terminates once a leader is elected. Intuitively, what the algorithm does in iteration i is that it goes over all network size estimates $\widetilde{n}$ from 2^1 to 2^{2^i-1} and spend $O(1)$ time for each $\widetilde{n}$. Once the number of devices n satisfies $\widetilde{n}/2 < n \le \widetilde{n}$, then a leader is elected with constant probability.

The main source of energy inefficiency of the above algorithm is the high energy cost of the basic subroutine with small $\widetilde{n}$-values. In particular, if $\widetilde{n} = O(1)$, Then the energy cost of achieve a success probability of $1 - f$ is $O(\log f^{-1})$ using our basic subroutine.

A key idea behind the proof of Theorem 1.2 is an observation that this energy complexity $O(\log f^{-1})$ can be improved exponentially in the Sender-CD model at the cost of increasing the time complexity. To see this, first consider the case when the number of devices is $n = 2$. We allocate an ID space of size $N = \lceil f^{-1} \rceil$, let each device choose an ID uniformly at random from $[N]$, and run the $O(N)$-time and $O(\log\log N)$-energy deterministic Sender-CD leader election algorithm of [12]. As long as the two devices select different IDs, the algorithm of [12] successfully elects a leader. Therefore, with probability at least $1 - f$, this procedure elects a leader in $O(N) = O(f^{-1})$ time and uses $O(\log\log N) = O(\log\log f^{-1})$ energy.

Table 2: Old and new results on deterministic leader election algorithms in No-CD, where N indicates the size of the ID space and n indicates the *known* number of devices.

Model	Time	Energy	Reference
No-CD	$O(N)$	$O(\log \frac{N}{n})$	[15]
No-CD	$\Omega(n \log \frac{N}{n})$	any	[18]
No-CD	any	$\Omega(\log \frac{N}{n})$	[15]
No-CD	$O(n \log \frac{N}{n})$	$O(\log \frac{N}{n})$	Theorem 1.3

By switching to a more energy-efficient Sender-CD algorithm when dealing with small $\widetilde{n}$-values, we are able to achieve an exponential improvement in the energy complexity. It is crucial that the increase in the time complexity is not too much when we switch from our basic subroutine to some other Sender-CD algorithm. To ensure that our final algorithm has a finite expected time complexity, the probability that the algorithm does not terminate by time t has to be $o(t^{-1})$ for all but a finite number of rounds t. To put it another way, for a given failure probability parameter f, the maximum allowed time complexity will be $o(f^{-1})$, meaning that the simple Sender-CD algorithm for $n = 2$ presented above is not suitable for our purpose.

To ensure that the time complexity is sufficiently small $o(f^{-1})$, we will employ a recent result [15] on time-energy tradeoffs of deterministic leader election in the Sender-CD model. By properly incorporating the Sender-CD algorithm of [15] into our framework to take care of small $\widetilde{n}$-values, we obtain a leader election algorithm $\mathcal{A}$ in the Sender-CD model that elects a leader in expected $O(\log^{1+\epsilon} n)$ time and using expected $O(\epsilon^{-1} \log\log\log n)$ energy, proving Theorem 1.2.

Derandomization. The proof of Theorem 1.3 follows from a derandomization of our basic subroutine, which elects a leader in $O(\log f^{-1})$ time with $O(\widetilde{n}^{-1} \log f^{-1})$ energy in No-CD. In Theorem 1.3, we consider the deterministic setting where the number of devices n satisfies $\widetilde{n}/2 < n \leq \widetilde{n}$ and each device has a unique identifier in $[N]$. To derandomize our basic subroutine, we set

$$f = 2^{-\Omega\left(\widetilde{n} \log \frac{N}{\widetilde{n}}\right)} < \frac{1}{\sum_{n \in (\widetilde{n}/2, \widetilde{n}]} \binom{N}{n}}$$

and apply a union bound over all size-n subsets of $[N]$ for $\widetilde{n}/2 < n \leq \widetilde{n}$. Our choice of f implies that the resulting deterministic algorithm has time complexity $O(\log f^{-1}) = O\left(\widetilde{n} \log \frac{N}{n}\right)$ and energy complexity $O(\widetilde{n}^{-1} \log f^{-1}) = O\left(\log \frac{N}{n}\right)$, as required.

1.6 Organization

In Section 2, we present an $\Omega(\log\log n)$ energy lower bound for Las Vegas leader election algorithms in No-CD, proving Theorem 1.1. In Section 3, we present our basic subroutine which we use in our randomized algorithms, and we derandomize it to obtain an optimal deterministic leader election algorithm in No-CD, proving Theorem 1.3. In Section 4, we present a framework for designing leader election algorithms and use it to reprove the result of [30] that in No-CD a leader can be elected in expected $O(\log n)$ time and using expected $O(\log\log n)$ energy.

Due to the page constraint, the proof of Theorem 1.2 is left to the full version of the paper [16]. Using our framework, in [16] we present a leader election algorithm in the Sender-CD model that finishes in expected $O(\log^{1+\epsilon} n)$ time and using expected $O(\epsilon^{-1} \log\log\log n)$ energy.

2 A TIGHT ENERGY LOWER BOUND FOR LAS VEGAS ALGORITHMS

In this section we prove Theorem 1.1, which gives an expected $\Omega(\log\log n)$ energy lower bound for any randomized algorithm $\mathcal{A}$ in the No-CD model that works for an unknown number of devices n and has a finite expected time complexity. In fact, we will prove the following stronger lower bound: With constant probability, the average energy cost among all n devices is $\Omega(\log\log n)$ in an execution of $\mathcal{A}$ on n devices.

Successful transmission. Instead of working with the leader election problem directly, we will consider the easier problem of having just one *successful transmission*, which is defined as follows. We say that a successful transmission occurs in a round if there is exactly one device transmitting in this round and there is at least one device listening in this round. We allow some devices to be idle when a successful transmission occurs. As we only consider the No-CD model in this section, the feedback from the communication channel is always silence if there is no successful transmission, so every device does not receive external information.

Description of an algorithm. Recall that in the deterministic setting, we assume each device v has a unique identifier $\mathsf{ID}(v) \in [N]$, where the size of ID space N is a global knowledge. As we do not care about the behavior of the algorithm after the first successful transmission, we may assume that a deterministic algorithm $\mathcal{A}$ is specified by a mapping ϕ from the ID space $x \in [N]$ to an infinite sequence of actions $(a_1(x), a_2(x), \ldots)$, where each $a_i(x) \in \{\mathsf{transmit}, \mathsf{listen}, \mathsf{idle}\}$ specifies the action of a device v with $\mathsf{ID}(v) = x$ in the ith round, assuming that the channel feedback is always silent whenever v listens in the first $i-1$ rounds. Similarly, a randomized algorithm $\mathcal{A}$ is specified by a distribution $\mathcal{D}$ of infinite sequences of actions $(a_1, a_2, \ldots)$. When a device v runs a randomized algorithm $\mathcal{A}$, it uses its private random bits to sample $(a_1, a_2, \ldots) \sim \mathcal{D}$ to determine its action $a_i \in \{\mathsf{transmit}, \mathsf{listen}, \mathsf{idle}\}$ in each round i if the channel feedback is always silent whenever v listens in the first $i-1$ rounds. Throughout this section, we use the above notation for describing a deterministic or a randomized algorithm $\mathcal{A}$.

Deterministic energy lower bound. Our proof of Theorem 1.1 relies on transforming deterministic lower bounds into randomized lower bounds. We first prove an $\Omega(\log \frac{N}{n})$ energy lower bound for deterministic algorithms in No-CD. This lower bound is a slightly stronger version of [15, Theorem 3] which considers the *average* energy cost.

Lemma 2.1 (Generalization of [15, Theorem 3]). *Let N be the size of the ID space, and let $2 \le n \le N/2$ be the number of devices. We allow both N and n to be global knowledge. Let $\mathcal{A}$ be any deterministic algorithm in the* No-CD *model that guarantees a successful transmission in the first t rounds for any choice of the size-n subset $V \subseteq [N]$ of devices. Let*

$$k_j = |\{a_i(j) \ : \ (i \in [t]) \wedge (a_i(j) \neq \text{idle})\}|$$

be the energy cost of a device with ID j in the first t rounds assuming that the channel feedback is always silent. Then we have

$$\frac{1}{N}\sum_{j=1}^{N} k_j = \Omega\left(\log \frac{N}{n}\right).$$

Proof. Re-order the N values $\{k_1, k_2, \ldots, k_N\}$ such that $k_1 \le k_2 \le \cdots \le k_N$. We will prove that $k_{N/2} = \Omega(\log \frac{N}{n})$, so the average value satisfies $\frac{1}{N}\sum_{j=1}^{N} k_j \ge \frac{1}{N}\cdot(\frac{N}{2}\cdot k_{N/2}) = \Omega(\log \frac{N}{n})$, as required.

We consider a random sequence $\{b_i\}_{i=1}^{t}$ where each b_i is uniformly randomly sampled from $\{\text{listen}, \text{transmit}\}$. We say $\{b_i\}_{i=1}^{t}$ matches a sequence $\{a_i(j)\}_{i=1}^{t}$ if for any i, either $a_i(j) = \text{idle}$, or $a_i(j) = b_i$. For each $j \in [\frac{N}{2}]$, since there are k_j listen or transmit actions in the sequence $\{a_i(j)\}_{i=1}^{t}$, it is easy to see that

$$\Pr[\{b_i\}_{i=1}^{t} \text{ matches } \{a_i(j)\}_{i=1}^{t}] = \frac{1}{2^{k_j}} \ge \frac{1}{2^{k_{N/2}}}.$$

Thus in expectation $\{b_i\}_{i=1}^{t}$ matches $N/2^{k_{N/2}+1}$ number of action sequences in $[\frac{N}{2}]$. This means there must exist some $\{b_i\}_{i=1}^{t}$ that matches at least $N/2^{k_{N/2}+1}$ number of action sequences in $[\frac{N}{2}]$. Let $V \subseteq [\frac{N}{2}]$ denote the set of devices that this $\{b_i\}_{i=1}^{t}$ matches with, and we have $|V| \ge N/2^{k_{N/2}+1}$.

The algorithm $\mathcal{A}$ cannot be correct on any set $V' \subseteq V$, because in any round all the devices in V' either all perform actions in $\{\text{listen}, \text{idle}\}$, or all perform actions in $\{\text{transmit}, \text{idle}\}$, so there does not exist a round where exactly one device transmits and at least one device listens. Since the algorithm $\mathcal{A}$ is correct on all sets of size n, we must have $n > |V| \ge N/2^{k_{N/2}+1}$, which gives $k_{N/2} = \Omega(\log \frac{N}{n})$. □

Randomized energy lower bound. Next, we prove the following energy lower bound for algorithms with success probability f using a reduction from the previous $\Omega(\log \frac{N}{n})$ energy lower bound for deterministic algorithms in No-CD.

Note that if $f \ge (8e)^{-n}$, then $\frac{1}{n}\log\frac{1}{f} = \log(8e) \le 5$, so $\Omega(n^{-1}\log f^{-1})$ becomes $\Omega(1)$, which is a trivial lower bound for leader election as well as many non-trivial tasks. In this sense the assumption $f < (8e)^{-n}$ in the following lemma is justified.

Lemma 2.2 (Energy lower bound for algorithms with error probability f). *Let $0 < f < (8e)^{-n}$. Let $\mathcal{A}$ be a randomized algorithm in the* No-CD *model that satisfies the following. When $\mathcal{A}$ is executed on a network of n devices, with probability at least $1 - f$, a successful transmission occurs in the first t rounds. We allow both parameters n and f to be global knowledge.*

Then there exists an integer $N = \Theta\left(n \cdot (1/f)^{1/n}\right)$ such that the following is satisfied. Define $k_1, k_2, \ldots, k_N$ as independent random variables

$$k_j = |\{a_i(j) \ : \ (i \in [t]) \wedge (a_i(j) \neq \text{idle})\}|,$$
$$\text{where } (a_1(j), a_2(j), \ldots) \sim \mathcal{D},$$

As a result, the expected value of each k_j is $\Omega(n^{-1}\log f^{-1})$.

Proof. Let N be the largest integer that satisfies $\binom{N}{n} < \frac{1}{4f}$. Since $f < (8e)^{-n}$ and $\binom{2n}{n} \le (2e)^n$, we must have $N \ge 2n$. By definition we have

$$\frac{1}{4f} \le \binom{N+1}{n} = \binom{N}{n}\cdot\frac{N+1}{N-n+1} \le 2\cdot\binom{N}{n}.$$

Thus we have $\frac{1}{f} = \Theta\left(\binom{N}{n}\right)$, and hence $\log\frac{1}{f} = \Theta\left(n\cdot\log\frac{N}{n}\right)$ and $N = \Theta\left(n\cdot(1/f)^{1/n}\right)$.

Consider the ID space $[N]$. For each $j \in [N]$, we fix the private random bits for the device v with $\mathsf{ID}(v)$ independently, and let

$$(a_1(j), a_2(j), \ldots) \sim \mathcal{D}$$

be the actions of v for each round i when running $\mathcal{A}$ using its private random bits, assuming that the channel feedback is always silent whenever v listens in the first $i-1$ rounds. We also define

$$k_j = |\{a_i(j) \ : \ (i \in [t]) \wedge (a_i(j) \neq \text{idle})\}|.$$

For a fixed subset S of n devices of $[N]$, the algorithm $\mathcal{A}$ executed on S guarantees a successful transmission within the first t rounds with probability at least $1 - f$. We define X to be the event that for all subsets S of n devices of $[N]$, $\mathcal{A}$ successfully elects a leader. Note that X depends on the private random bits used in sampling $(a_1(j), a_2(j), \ldots) \sim \mathcal{D}$ for all $j \in [N]$. Using a union bound, and since $\binom{N}{n} < \frac{1}{4f}$,

$$\Pr[X] \ge 1 - f\cdot\binom{N}{n} \ge \frac{3}{4}.$$

If for some fixed random bits the event X happens, then the algorithm $\mathcal{A}$ with those fixed random bits yields a deterministic algorithm for ID space $[N]$. The $\Omega(\log\frac{N}{n})$ energy lower bound of Lemma 2.1 then implies that the lower bound $\frac{1}{N}\sum_{j=1}^{N} k_j = \Omega\left(\log\frac{N}{n}\right)$ holds whenever X happens. Thus we have

$$\Pr\left[\frac{1}{N}\sum_{j=1}^{N} k_j = \Omega\left(\frac{1}{n}\log\frac{1}{f}\right)\right] = \Pr\left[\frac{1}{N}\sum_{j=1}^{N} k_j = \Omega\left(\log\frac{N}{n}\right)\right] \ge \frac{3}{4}.$$

□

In the following, we apply Lemma 2.2 to analyze the energy complexity of any algorithm $\mathcal{A}$ meeting the conditions specified in Theorem 1.1.

Lemma 2.3 ($\Omega(\log t)$ energy in t rounds). *Let $\mathcal{A}$ be a randomized leader election algorithm in the* No-CD *model that works for any unknown number of devices $n \ge 2$, and it has a finite expected time complexity for some $n = n^*$.*

Then there exists an infinite set S of positive integers such that for any $t \in S$, there exists an integer $N = \Theta\left(t^{1/n^*}\right)$ such that the following is satisfied. Define $k_1, k_2, \ldots, k_N$ as independent random variables

$$k_j = |\{a_i(j) \ : \ (i \in [t]) \wedge (a_i(j) \neq \text{idle})\}|, \quad \text{where } (a_1(j), a_2(j), \ldots) \sim \mathcal{D},$$

representing the energy cost of $\mathcal{A}$ for a device v with $\mathsf{ID}(v) = j$ in the first t rounds assuming that the channel feedback is always silent. Then we have

$$\Pr\left[\frac{1}{N}\sum_{j=1}^{N} k_j = \Omega(\log t)\right] \geq \frac{3}{4}.$$

In particular, the expected value of each k_j is $\Omega(\log t)$.

Proof. The expected time of $\mathcal{A}$ when running on n^* devices is

$$\sum_{t=1}^{\infty} \Pr[\mathcal{A} \text{ doesn't finish in } t \text{ rounds when running on } n^* \text{ devices}].$$

Since we assume this expected time is finite, and $\sum_{t=1}^{\infty} \frac{1}{t} = \infty$, there must exist an infinite set S of positive integers such that for any $t \in S$,

$$\Pr[\mathcal{A} \text{ doesn't finish in } t \text{ rounds when running on } n^* \text{ devices}] \leq \frac{1}{t}.$$

Fix any $t \in S$. If we run $\mathcal{A}$ on n^* devices, then $\mathcal{A}$ successfully elects a leader within the first t rounds with probability at least $1 - \frac{1}{t}$. In particular, it guarantees a successful transmission within the first t rounds with probability at least $1 - \frac{1}{t}$.

Therefore, applying Lemma 2.2 with $f = 1/t$, $n = n^*$, and since n^* is a fixed constant, there exists an integer $N = \Theta\left(n^* \cdot t^{1/n^*}\right) = \Theta\left(t^{1/n^*}\right)$ such that the conclusion of this lemma is satisfied. Although the criterion $0 < f < (8e)^{-n}$ needed for applying Lemma 2.2 might not be met when t is small, the set S excluding those small t is still infinite. □

The last missing piece is the following lemma that shows an $\Omega(\log n)$ time lower bound when the algorithm succeeds with constant probability. The proof of this lemma follows from the same proof argument of [34, Theorem 5.2], which shows an $\Omega(\log n)$ *expected* time lower bound, so here we only include a proof sketch. We note that the purpose of the technical condition $\sqrt{\widetilde{n}} \leq n \leq \widetilde{n}$ in the lemma is to ensure that the $\Omega(\log\log n)$ energy lower bound in Theorem 1.1 applies to infinitely many n.

Lemma 2.4 ($\Omega(\log n)$ time for constant success probability [34])**.** *Let $\mathcal{A}$ be a randomized algorithm in the* Sender-CD *model such that for any known integer $\widetilde{n} \geq 2$, for any unknown number of devices $\sqrt{\widetilde{n}} \leq n \leq \widetilde{n}$, the probability that a collision-free transmission occurs by time $T(\widetilde{n})$ in an execution of $\mathcal{A}$ on n devices is at least $1/4$. Then we have $T(\widetilde{n}) = \Omega(\log \widetilde{n})$.*

Proof sketch. Following the terminology of [34], for any two sets $F, H \subseteq [N]$, we say that F hits H if $|F \cap H| = 1$. In Theorem 3.1 of [2], it was shown that given any integer $N \geq 2$, there exists a multiset $\mathcal{H}$ of subsets $H \subseteq [N]$ such that every subset $F \subseteq [N]$ hits at most $O\left(\frac{1}{\log N}\right)$ fraction of $\mathcal{H}$. In fact this result also holds with the additional requirement that each subset H in the multiset $\mathcal{H}$ has size $\sqrt{N} \leq |H| \leq N$. This follows from a straightforward extension of the original proof of [2], and for simplicity we omit it.

We follow the same proof strategy as Theorem 5.2 of [34] which proves an $\Omega(\log n)$ expected time lower bound using Theorem 3.1 of [2]. Construct the multiset $\mathcal{H}$ with $N = \widetilde{n}$. For each set $H \subseteq [\widetilde{n}]$ in the multiset $\mathcal{H}$, consider running the algorithm $\mathcal{A}$ on the devices in H. In each round, a collision-free transmission occurs if and only if the set of the transmitting devices $F \subseteq [\widetilde{n}]$ hits the set H. Since any set F hits at most $O\left(\frac{1}{\log \widetilde{n}}\right)$ fraction of $\mathcal{H}$, in order to achieve a success probability of $1/4$, $\mathcal{A}$ needs to run for at least $\Omega(\log \widetilde{n})$ rounds. □

Now we are ready to prove Theorem 1.1 by combining the $\Omega(\log n)$ time lower bound in Lemma 2.4 with the $\Omega(\log t)$ energy lower bound in Lemma 2.3.

Theorem 1.1 (Energy lower bound for Las Vegas algorithms)**.** *Let $\mathcal{A}$ be a randomized leader election algorithm in the* No-CD *model. We write $T(n)$ and $E(n)$ to denote the expected time and energy complexities of $\mathcal{A}$. Suppose there is some integer $n^* \geq 2$ such the expected time $T(n^*)$ of the algorithm $\mathcal{A}$ when running on $n = n^*$ devices is finite. Then there exist infinite number of network sizes n such that $E(n) = \Omega(\log\log n)$.*

Proof. From Lemma 2.4 we have that there exists a constant $c > 0$ such that for each integer $\widetilde{n} \geq 2$, there exists an integer $\sqrt{\widetilde{n}} \leq n \leq \widetilde{n}$ such that when running $\mathcal{A}$ on n devices, we have

$$\Pr[\text{ no successful transmission in the first } c\log n \text{ rounds }] \geq 3/4. \tag{1}$$

Let S be the infinite set obtained from applying Lemma 2.3 with the algorithm $\mathcal{A}$. Consider any $t \in S$. Choose $\widetilde{n} = 2^{2t/c}$, and then pick an integer $\sqrt{\widetilde{n}} \leq n \leq \widetilde{n}$ such that when running $\mathcal{A}$ on n devices, Eq. (1) holds. The existence of such an integer $\widetilde{n}$ is guaranteed by Lemma 2.4.

Our choice of n ensures that $t \leq c\log n$, so Eq. (1) implies that when running $\mathcal{A}$ on n devices, with probability at least $3/4$, no successful transmission occurs within the first $t = \Theta(\log n)$ rounds.

Consider the integer integer $N = \Theta\left(t^{1/n^*}\right)$ in Lemma 2.3. Let k denote the average energy used by the first N devices in t rounds. Lemma 2.3 implies that

$$\Pr[k \geq \Omega(\log t)] \geq 3/4. \tag{2}$$

Combining Eq. (1) and (2) with a union bound, we obtain that with probability at least $1/2$, when running $\mathcal{A}$ on n devices, there is at least one device that uses at least $\Omega(\log t) = \Omega(\log\log n)$ energy. Since there are infinitely many $t \in S$, we are able to select infinitely many n such that the $\Omega(\log\log n)$ lower bound holds.

For the rest of the proof, we extend the above argument to show that in fact with probability at least $1/2$, when running $\mathcal{A}$ on n devices, the average energy cost per device is $\Omega(\log t) = \Omega(\log\log n)$, so the expected energy cost per device is indeed $\Omega(\log\log n)$.

Let $v_1, v_2, \ldots, v_n$ denote the n devices. For each $j \in [n]$, let k_j be the random variable representing the energy cost of $\mathcal{A}$ for v_j in the first t rounds assuming that the channel feedback is always silent.

It is clear that $k_1, k_2, \ldots, k_n$ are independent, and each k_j depends only on the private random bits in v_j. It suffices to show that

$$\Pr\left[\frac{1}{n}\sum_{j=1}^{n} k_j = \Omega\left(\log t\right)\right] \geq \frac{3}{4}.$$

To prove this bound, we partition $\{k_1, k_2, \ldots, k_n\}$ into $\lfloor n/N \rfloor$ disjoint groups of size exactly N and at most one leftover group. For each group, with probability at least 3/4, the average value is $\Omega(\log t)$. Applying a Chernoff bound over all $\lfloor n/N \rfloor$ disjoint groups of size N, we obtain that the average value of $\{k_1, k_2, \ldots, k_n\}$ is $\Omega(\log t)$ with probability $1 - e^{-\Omega(n/N)} = 1 - e^{-\Omega(n/\log n)} \gg 3/4$, as long as n is sufficiently large, as $N = O\left(t^{1/n^*}\right) = o(\log n)$. □

3 LEADER ELECTION WITH A NETWORK SIZE ESTIMATE

For upper bounds, we begin with the case where an estimate $\widetilde{n}$ of the actual number of devices n is given. Given a parameter $0 < f < 1$, our task is to elect a leader with probability $1 - f$ when the estimate $\widetilde{n}$ satisfies $\widetilde{n}/2 < n \leq \widetilde{n}$.

In this section, we design an algorithm solving this task with $O(\log f^{-1})$ time and $O(\widetilde{n}^{-1} \log f^{-1})$ energy in the No-CD model. In Section 3.1 we extend this algorithm to deal with multiple pairs of $(\widetilde{n}, f)$, and the resulting algorithm will later be used as a subroutine in our leader election algorithms for the more challenging scenario where the number of devices n is completely unknown. In Section 3.2, we derandomize our algorithm to give an optimal deterministic leader election algorithm.

Balls-into-bins. We need the following balls-into-bins lemma. In this lemma we care about a subset of the bins, which we call "good" bins, and the rest of the bins are called "bad" bins. The goal of this lemma is to analyze the number of good bins that contain exactly one ball.

In Lemma 3.1, the numbers αN and γN are not required to be integers, but $n = \beta N$ must be an integer, as it specifies the number of balls in a balls-into-bins experiment.

Lemma 3.1 (Number of good bins with exactly one ball)**.** *Let N be an integer. Let $\alpha, \beta, \gamma \in (0, 1)$ be three numbers satisfying $\alpha > 2\beta + \gamma$. Let t be any parameter such that $t < \alpha - 2\beta - \gamma$. There are $n = \beta N$ balls and N bins. Among the bins there are at least αN good bins, and the rest of the bins are called bad bins.*

Consider the following balls-into-bins experiment. For each ball, there is an arbitrary subset of at least $(1 - \gamma)N$ bins such that the ball is uniformly randomly thrown into one of the bins in this subset. With probability at least $1 - e^{-\frac{t^2 n}{2}}$, there exist at least $(\alpha - 2\beta - \gamma - t)n$ good bins that contain exactly one ball.

Proof. We use a random variable Y to denote the total number of good bins that contain exactly one ball.

We define n random variables $X_1, X_2, \ldots, X_n \in \{-1, 0, 1\}$ as follows. Before throwing the ith ball into a random bin, we first reorder the bins in such a way that the good bins that are empty are in the front, so the first $(\alpha - \beta)N$ bins are always empty good bins. We define

$$X_i = \begin{cases} 1 & \text{if the } i\text{th ball is thrown into the first } (\alpha - \beta)N \text{ bins,} \\ 0 & \text{if the } i\text{th ball is thrown into a bad bin,} \\ -1 & \text{otherwise.} \end{cases}$$

We have $Y \geq \sum_{i=1}^{n} X_i$. This is because for each i, if $X_i = 1$ (the ball is thrown into an empty good bin), Y increases by 1, and if $X_i = -1$ (the ball is thrown into a possibly non-empty good bin), Y decreases by at most 1.

Observe that the X_i's are independent, and $\forall i \in [n]$, we have $\Pr[X_i = 1] \geq (\alpha - \beta - \gamma)/(1 - \gamma)$ and $\Pr[X_i = -1] \leq \beta/(1 - \gamma)$, so we have

$$\mathbb{E}[X_i] \geq 1 \cdot (\alpha - \beta - \gamma)/(1 - \gamma) + (-1) \cdot \beta/(1 - \gamma) \geq \alpha - 2\beta - \gamma.$$

Using Hoeffding's inequality, together with the fact that X_i is bounded within the interval $[a_i, b_i]$ with $a_i = -1$ and $b_i = 1$, we have

$$\begin{aligned} \Pr\left[Y \leq (\alpha - 2\beta - \gamma - t)n\right] &\leq \Pr\left[\sum_{i=1}^{n} X_i \leq (\alpha - 2\beta - \gamma - t)n\right] \\ &\leq \Pr\left[\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}[X_i]) \leq -t\right] \\ &\leq \exp\left(-\frac{2n^2 t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) \\ &\leq e^{-\frac{t^2 n}{2}}. \end{aligned}$$ □

Using the balls-into-bins procedure of Lemma 3.1 and the ID assignment procedure from the proof of [12, Lemma 18], we design the following basic leader election algorithm.

Lemma 3.2 (Basic leader election algorithm)**.** *Let S be a set of devices, where all devices in S agree on an integer $n' \geq 2$. There is an algorithm in the* No-CD *model achieving the following goals.*

- *The algorithm costs $O(n')$ time and $O(1)$ energy in the worst case.*
- *The algorithm elects a leader with probability $1 - 2^{-\Omega(n')}$ if $n'/3 \leq |S| \leq n'$.*

Proof. We define $N' = 100n'$. The algorithm has two phases.

ID assignment. In Phase I, the devices in S contend for the IDs in the ID space $[N']$. Each device v first picks a single *transmitting ID* uniformly at random from $[N']$ and then picks 30 *listening IDs* uniformly at random from the remaining $N' - 1$ IDs with duplicates.

For each $i \in [N']$, there are two rounds, and each device has three possible actions:

(1) If i is a transmitting ID for the device v, then v transmits in the first round and listens in the second round.
(2) If i is a listening ID for the device v, then v listens in the first round, and v transmits in the second round if v received a message in the first round, otherwise v stays idle in the second round.
(3) If i is neither a transmitting ID nor a listening ID for the device v, then v stays idle in both two rounds.

If i is a transmitting ID for a device v and v receives a message in the second round, then the ID i is assigned to this device v. It is

straightforward to verify that each ID is assigned to at most one device. The algorithm costs $O(n')$ time and $O(1)$ energy. We write $S' \subseteq S$ to denote the set of the devices that are assigned IDs.

Leader election. In Phase II, the devices in S' elect a leader by running the deterministic algorithm of [15, Theorem 2] over the space $[N']$. The algorithm costs $O(N') = O(n')$ time and $O\left(\log \frac{N'}{|S'|}\right)$ energy.

For the case $n'/3 \le |S| \le n'$, we will later show that with probability $1 - 2^{-\Omega(n')}$ we have $|S'| = \Theta(N')$, so the leader election algorithm of [15, Theorem 2] costs $O(1)$ energy, as required. To ensure that the energy usage of our algorithm never exceeds $O(1)$, we let each device stop participating in the leader algorithm once its energy usage exceeds the required upper bound $O(1)$.

Analysis. For the rest of the proof, we show that with probability $1 - 2^{-\Omega(n')}$, the number of IDs that are assigned is $\Theta(N')$. Recall that an ID $i \in [N']$ is assigned if it is a listening ID for exactly one device and a transmitting ID for exactly one device.

First, consider the listening IDs. We use Lemma 3.1 with $N = N'$, $\alpha = 1$, $\beta = 30|S|/N'$, $\gamma = 1/N'$, and $t = 1/200$. Here we interpret the ID space $[N']$ as $N = N'$ bins, where all of them are good. The $30|S| = \beta N'$ random choices of listening IDs are seen as $\beta N'$ balls. Each ball is thrown to a random bin in a subset of $N' - 1 = (1-\gamma)N$ bins. We have $0.1 \le \beta \le 0.3$ since $n'/3 \le |S| \le n'$ and $N' = 100n'$. We have $\gamma \le 1/200$ since $n' \ge 2$ and $N' = 100n'$. Hence $\alpha - 2\beta - \gamma - t \ge 1 - 0.6 - 0.005 - 0.005 > 1/3$, so the probability that the number of IDs in $[N]$ that are assigned as a listening ID to exactly one device is at least $(\alpha - 2\beta - \gamma - t)\beta N \ge \beta N/3 \ge N/30$ with probability at least $1 - 2^{-\Omega(n')}$, by Lemma 3.1.

Next, consider the transmitting IDs. We use Lemma 3.1 with $N = N'$, $\alpha = 1/30$, $\beta = |S|/N'$, $\gamma = 0$, and $t = 0.01$. Again, we interpret the ID space $[N']$ as $N = N'$ bins, but only the ones assigned as a listening ID to exactly one device are considered good. In the following analysis, we condition on the event that the number of IDs assigned as a listening ID to exactly one device is at least $N/30 = \alpha N'$, which occurs with probability $1 - 2^{-\Omega(n')}$. The $|S| = \beta N'$ random choices transmitting IDs are seen as $\beta N'$ balls. Each ball is thrown to a bin uniformly at random from the set of all $N' = (1-\gamma)N$ bins. We have $0.01/3 \le \beta \le 0.01$ since $n'/3 \le |S| \le n'$ and $N' = 100n'$. Hence $\alpha - 2\beta - \gamma - t \ge 1/30 - 0.02 - 0 - 0.01 = 1/300$, and the probability that the number of IDs in $[N]$ that are assigned as a transmitting ID to exactly one device and assigned as a listening ID to exactly one device is at least $(\alpha - 2\beta - \gamma - t)\beta N \ge \beta N/300 \ge N/90000$ with probability at least $1 - 2^{-\Omega(n')}$, as required. □

Main algorithm. We extend Lemma 3.2 to cope with a general failure probability parameter $0 < f < 1$. We show that leader election can be done in worst-case $O(\log f^{-1})$ time and expected $O(\widetilde{n}^{-1} \log f^{-1})$ energy, see Theorem 3.1 for the precise specification of our algorithm. Later in Section 3.2 we will show that both the time and energy complexities of our algorithm are optimal.

Since the expected energy cost $O(\widetilde{n}^{-1} \log f^{-1})$ can be much smaller than one, it is implicit in the statement of Theorem 3.1 that each device participates in the algorithm with probability $p = \min\{1, O(\widetilde{n}^{-1} \log f^{-1})\}$ independently. If a device v chooses to not participate in the algorithm, then v stays idle throughout the algorithm and its energy usage is zero. If a device v chooses to participate in the algorithm, then the energy cost of v is at most $O(1 + \widetilde{n}^{-1} \log f^{-1})$.

In Theorem 3.1, we do not require all devices to be informed whether a leader is elected by the end of the algorithm. Indeed, if $f = 2^{-o(\widetilde{n})}$, then the expected energy cost is much less than one, so a majority of the devices do not participate in the algorithm at all, and these devices cannot know the outcome of the algorithm as they remain idle throughout the algorithm.

We can allocate one additional round after running the algorithm of Theorem 3.1 to let the elected leader speak to the rest of the devices. This costs one unit of energy for all devices. We choose to not include this step in the algorithm of Theorem 3.1 because this will break the bound $O(\widetilde{n}^{-1} \log f^{-1})$ of the expected energy cost. Of course, for the case $\widetilde{n}^{-1} \log f^{-1} = \Omega(1)$, we can assume that the outcome of the leader election is known to all devices.

Theorem 3.1 (Leader election given a network size estimate $\widetilde{n}$)**.** *Given a number $0 < f < 1$ and an integer $\widetilde{n} \ge 2$, there is an algorithm $\mathcal{A}$ achieving the following goals.*

- **Expected energy:** *The expected energy cost for a device is $O(\widetilde{n}^{-1} \log f^{-1})$.*
- **Worst-case energy:** *The worst-case energy cost for a device is $O(1 + \widetilde{n}^{-1} \log f^{-1})$.*
- **Time:** *The time complexity of the algorithm is $O(\log f^{-1})$, which is a fixed number independent of the random bits used by the algorithm.*
- **Leader election:** *By the end of the algorithm, at most one device identifies itself as a leader. If $\widetilde{n}/2 < n \le \widetilde{n}$, then the probability that no leader is elected is at most f.*

Proof. For each integer $2 \le n' \le \widetilde{n}$, we define $f_1 = 2^{-\Omega(n')}$ to be the failure probability of leader election in Lemma 3.2 with parameter n', and we define $f_2 = 2^{-\Omega(n')}$ to be the probability defined as follows. Assuming that the actual network size n satisfies $\widetilde{n}/2 < n \le \widetilde{n}$, pick a subset of devices S by including each device with probability $0.9 \cdot (n'/\widetilde{n})$ independently. Let f_2 be the maximum probability that the inequality $n'/3 \le |S| \le n'$ does not hold, where the maximum ranges over all n satisfying $\widetilde{n}/2 < n \le \widetilde{n}$. By a Chernoff bound, we have $f_2 = 2^{-\Omega(n')}$.

Suppose that there exists a number $2 \le n' \le \widetilde{n}$ satisfying $f_1 + f_2 \le f$, then we simply pick a subset of devices S by including each device with probability $0.9 \cdot (n'/\widetilde{n})$ and run the algorithm of Lemma 3.2 with n' and S. Such a number n' must satisfies $n' = O(\log f^{-1})$, so each device participates in the algorithm with probability $0.9 \cdot (n'/\widetilde{n}) = O(\widetilde{n}^{-1} \log f^{-1})$ independently. Since the energy cost in Lemma 3.2 is $O(1)$, the expected and worst-case energy cost of our algorithm are $O(\widetilde{n}^{-1} \log f^{-1})$ and $O(1)$. By Lemma 3.2, the time complexity of our algorithm is $O(n') = O(\log f^{-1})$. A leader is guaranteed to be elected if $n'/3 \le |S| \le n'$ and the algorithm of Lemma 3.2 succeeds. This occurs with probability at least $1 - f_1 - f_2 = 1 - f$ whenever $\widetilde{n}/2 < n \le \widetilde{n}$.

Suppose that there does not exist a number $2 \le n' \le \widetilde{n}$ with $f_1 + f_2 \le f$. Then we must have $f = 2^{-\Omega(\widetilde{n})}$ and so $\widetilde{n}^{-1} \log f^{-1} = \Omega(1)$. In this case, we simply let S be the set of all devices and run the algorithm of Lemma 3.2 with $n' = \widetilde{n}$ for $C = \Theta(\widetilde{n}^{-1} \log f^{-1})$ times. For the case $\widetilde{n}/2 < n \le \widetilde{n}$, the probability that no leader is elected

in all C iterations is $2^{-\Omega(C\widetilde{n})} = 2^{-\Omega(\log f^{-1})}$, which can be made at most f by selecting $C = \Theta(\widetilde{n}^{-1} \log f^{-1})$ to be sufficiently large. From the time and energy complexities specified in Lemma 3.2, it is clear the our algorithm satisfies all the requirements in the statement of Theorem 3.1. □

3.1 Multiple Instances

Suppose that we are given k pairs $(\widetilde{n}_1, c_1), (\widetilde{n}_2, c_2), \ldots, (\widetilde{n}_k, c_k)$ such that we need to invoke the algorithm of Theorem 3.1 with parameters $\widetilde{n}_i$ and $f = 2^{-c_i}$ for each $1 \le i \le k$. A simple calculation shows that the worst-case energy complexity of the combined algorithm is $O\left(k + \sum_{j=1}^{k} c_i \widetilde{n}_i^{-1}\right)$. This bound can be improved to $O\left(1 + \sum_{j=1}^{k} c_i \widetilde{n}_i^{-1}\right)$ at the cost of allowing the random bits used in different executions of the algorithm of Theorem 3.1 to correlate.

Consider the probability $p_i = \min\{1, O(\widetilde{n}_i^{-1} c_i)\}$ that a device participates in an execution of the algorithm of Theorem 3.1 with parameters $\widetilde{n} = \widetilde{n}_i$ and $f = 2^{-c_i}$. We partition the indices $\{1, 2, \ldots, k\}$ into groups $[k] = B_1 \cup B_2 \cup \cdots \cup B_s$ such that $1/2 \le \sum_{i \in B_j} p_i \le 1$ for each $1 \le j < s$ and $\sum_{i \in B_j} p_i \le 1$ for $j = s$. It is clear that such a partition exists, and the number s of groups is at most $1 + \sum_{j=1}^{k} 2p_j = O\left(1 + \sum_{j=1}^{k} c_j \widetilde{n}_j^{-1}\right)$.

Consider any group B_j. Since $\sum_{i \in B_j} p_i \le 1$, we can ensure that each device only participate in at most one execution of the algorithm of Theorem 3.1 associated with the indices in B_j. Specifically, we let each device v samples a random variable x such that $x = i$ with probability p_i for each $i \in B_j$. Then each device v participates in the execution of the algorithm of Theorem 3.1 associated with the index $i \in B_j$ if $x = i$. Hence the worst-case energy cost for invoking the algorithm of Theorem 3.1 for all indices $i \in B_j$ is $\max_{i \in B_j} O(1 + \widetilde{n}_i^{-1} c_i) = O\left(1 + \sum_{i \in B_j} c_i \widetilde{n}_i^{-1}\right)$ instead of the bound $O\left(\sum_{i \in B_j} \left(1 + c_i \widetilde{n}_i^{-1}\right)\right) = O\left(|B_j| + \sum_{i \in B_j} c_i \widetilde{n}_i^{-1}\right)$ given by the straightforward summation.

Going over all groups $B_1, B_2, \ldots, B_s$, the overall worst-case energy cost for invoking the algorithm of Theorem 3.1 with parameters $(\widetilde{n}_1, c_1), (\widetilde{n}_2, c_2), \ldots, (\widetilde{n}_k, c_k)$ is

$$O\left(\sum_{j=1}^{s}\left(1 + \sum_{i \in B_j} c_i \widetilde{n}_i^{-1}\right)\right) = O\left(s + \sum_{i=1}^{k} c_i \widetilde{n}_i^{-1}\right) = O\left(1 + \sum_{i=1}^{k} c_i \widetilde{n}_i^{-1}\right).$$

We summarize the discussion as a lemma. This lemma will be used in the algorithms of Section 4.

Lemma 3.3 (Multiple instances)**.** *Given parameters $(\widetilde{n}_1, c_1), (\widetilde{n}_2, c_2), \ldots, (\widetilde{n}_k, c_k)$, there is an algorithm $\mathcal{A}$ achieving the following goals in the* No-CD *model.*

- **Energy:** *The worst-case energy cost for a device is $O\left(1 + \sum_{j=1}^{k} c_j \widetilde{n}_j^{-1}\right)$.*
- **Time:** *The time complexity of the algorithm is $O\left(\sum_{j=1}^{k} c_j\right)$, which is a fixed number independent of the random bits used by the algorithm.*
- **Leader election:** *For each $1 \le j \le k$, at most one device identifies itself as the jth leader. The probability that the jth leader is not elected is at most 2^{-c_j} if $\widetilde{n}_j/2 < n \le \widetilde{n}_j$.*

3.2 Derandomization

By derandomizing Theorem 3.1, we obtain an optimal deterministic leader election algorithm for the No-CD model with $O(n \log \frac{N}{n})$ time and $O(\log \frac{N}{n})$ energy, where the size N of the ID space $[N]$ and an estimate $\widetilde{n}$ of the number of devices n such that $\widetilde{n}/2 < n \le \widetilde{n}$ are both known to all devices. As we will later see, the optimality of the deterministic algorithm implies the optimality of Theorem 3.1.

Theorem 1.3 (Optimal deterministic leader election)**.** *Suppose that the size N of the ID space $[N]$ and an estimate $\widetilde{n}$ of the number of devices n such that $\widetilde{n}/2 < n \le \widetilde{n}$ are both known to all devices. There is a deterministic leader election algorithm in the* No-CD *model with time complexity $T = O\left(n \log \frac{N}{n}\right)$, energy complexity $E = O\left(\log \frac{N}{n}\right)$.*

Proof. In Theorem 3.1 we pick $f = \frac{1}{1+\sum_{\widetilde{n}/2<n\le\widetilde{n}} \binom{N}{n}}$. Consider an assignment ϕ from $[N]$ to an infinite sequence of random bits. When we run the randomized algorithm of Theorem 3.1 on a device v with identifier i, v uses the random bits $\phi(i)$. For each fixed size-n subset of $[N]$ with $\widetilde{n}/2 < n \le \widetilde{n}$, the simulation of the randomized algorithm of Theorem 3.1 successfully elects a leader with probability at least $1 - f$. By a union bound, there is a non-zero probability that the simulation of the randomized algorithm of Theorem 3.1 succeeds for all size-n subsets of $[N]$ such that $\widetilde{n}/2 < n \le \widetilde{n}$. This non-zero probability implies the existence of a deterministic algorithm that works for all size-n subsets of $[N]$ such that $\widetilde{n}/2 < n \le \widetilde{n}$. Since $\widetilde{n}/2 < n \le \widetilde{n}$ and $\log f^{-1} = O\left(n \log \frac{N}{n}\right)$, this algorithm has time complexity $T = O(\log f^{-1}) = O\left(n \log \frac{N}{n}\right)$ and energy complexity $E = O(\widetilde{n}^{-1} \log f^{-1}) = O\left(\log \frac{N}{n}\right)$, by Theorem 3.1. □

The algorithm of Theorem 1.3 is both energy-optimal and time-optimal in No-CD, due to the energy lower bound of $\Omega\left(\log \frac{N}{n}\right)$ in [15, Theorem 3] and the time lower bound of $\Omega\left(n \log \frac{N}{n}\right)$ in [18, Theorem 3.3].

The optimality of Theorem 1.3 implies that Theorem 3.1 the time $O(\log f^{-1})$ and energy $O(\widetilde{n}^{-1} \log f^{-1})$ complexities of Theorem 3.1 are optimal. If the time complexity of Theorem 1.3 can be made $o(\log f^{-1})$, then we can derandomize it to give a deterministic leader election algorithm whose time complexity is $o\left(n \log \frac{N}{n}\right)$, violating the $\Omega\left(n \log \frac{N}{n}\right)$ lower bound. Similarly, if the energy complexity of Theorem 1.3 can be made $o(\widetilde{n}^{-1} \log f^{-1})$, then we can derandomize it to give a deterministic leader election algorithm whose energy complexity is $o\left(\log \frac{N}{n}\right)$, violating the $\Omega\left(\log \frac{N}{n}\right)$ lower bound.

4 LEADER ELECTION WITH AN UNKNOWN NUMBER OF DEVICES

In this section, we design randomized leader election algorithms in the model in the scenario where the number of devices n is completely unknown. In Section 4.1 we describe the framework for our leader election algorithms. To illustrate the use of our framework. in Section 4.2, we reprove a result of [30] using our framework that

in No-CD a leader can be elected in expected $O(\log n)$ time and using expected $O(\log\log n)$ energy.

4.1 Basic Framework

Our randomized leader election algorithms proceeds in iterations. We will specify an infinite sequence of deadlines $(d_1, d_2, \ldots)$ such that the ith iteration finishes by time $O(d_i)$. Alternatively, we require the algorithm of the ith iteration to take at most $O(d_i - d_{i-1})$ time, and we let $d_0 = 0$ for convenience.

One iteration. Each iteration of the algorithm is specified by a list of pairs of positive integers $(\widetilde{n}_1, c_1), (\widetilde{n}_2, c_2), \ldots, (\widetilde{n}_k, c_k)$. Given such a list, the algorithm of one iteration has the following three parts.

- **Using the basic subroutine:** In the first part, we run the algorithm of Lemma 3.3 with parameters $(\widetilde{n}_1, c_1), (\widetilde{n}_2, c_2), \ldots, (\widetilde{n}_k, c_k)$.
- **Leader election:** In the second part, we allocate $O(k)$ rounds to elect one leader among the at most k leaders elected during the execution of the algorithm of Lemma 3.3. More specifically, we allocate an ID space $[N]$ with $N = k$. For each $1 \le j \le k$, if a device v is the jth leader elected in Lemma 3.3, then v grabs the ID j. Then we run the well-known $O(N)$-time and $O(\log N)$-energy deterministic No-CD algorithm [12] to elect a single leader among these devices. It is possible that a device v grabs multiple IDs. In this case, the ID of v in the deterministic algorithm is set to be the smallest identifier among all identifiers that v grabs.
- **Announcing the result:** In the third part, we allocate one round to let the leader elected during the second part of the algorithm speak, while all other devices listen. If a leader is elected, the algorithm terminates. Otherwise, the algorithm moves on to the next iteration.

It is possible that no leader is elected, as there is no guarantee that the algorithm of Lemma 3.3 must elect a leader.

Time. The time complexity of the algorithm is dominated by the round complexity of the first part, which is $O\left(\sum_{j=1}^{k} c_j\right)$ according to Lemma 3.3. Here we assume that each c_j is a positive integer, so $\sum_{j=1}^{k} c_j \ge k$. The number of rounds used by the algorithm is fixed independent of the randomness, so the time complexity upper bound holds in the worst case.

Energy. For the first part of the algorithm, the energy complexity is $O\left(1 + \sum_{j=1}^{k} c_i \widetilde{n}_i^{-1}\right)$ by Lemma 3.3. For the second part of the algorithm, the energy cost is $O(\log k)$ among those devices participating in the leader election algorithm, and it is zero for the rest of the devices. For the third part of the algorithm, the energy cost is one unit for all devices. All these upper bounds hold in the worst case.

4.2 An $O(\log n)$-time and $O(\log\log n)$-energy Algorithm in No-CD

Using the framework of algorithm design in Section 4.1, we reprove a result of [30] that in the No-CD model a leader among an unknown number $n \ge 2$ of devices can be elected using $O(\log n)$ time and $O(\log\log n)$ energy in expectation. Our algorithm is described by the following choice of parameters.

- **The deadline sequence:** We set $d_i = 2^i$ for each $i \ge 1$.
- **The parameters for each iteration:** The algorithm of the ith iteration is specified by the list of pairs $(\widetilde{n}, 2)$ for all $\widetilde{n} = 2^1, 2^2, \ldots, 2^{d_i - 1}$.

With the above choice of parameters, the algorithm for the ith iteration takes $O\left(\sum_{j=1}^{d_i-1} 2\right) = O(d_i) = O(d_i - d_{i-1})$ time, so the choice of the deadline sequence is valid.

Failure probability. Suppose that we run the algorithm on a network of $n \ge 2$ devices. We pick $\widetilde{n} = 2^{j^*}$ to be the number such that $\widetilde{n}/2 < n \le \widetilde{n}$ and j^* is an integer. Let i^* be the smallest index i such that $\log \widetilde{n} < d_i$, so the pair $(\widetilde{n}, 2)$ is included in the specification for each iteration $i \ge i^*$. Therefore, for each $i \ge i^*$, if we run the ith iteration of the algorithm on a network of $n \ge 2$ devices, then the probability that no leader is elected in this iteration is at most $1/4$, as $\widetilde{n}/2 < n \le \widetilde{n}$ and the pair $(\widetilde{n}, 2)$ is considered in the ith iteration. We define

$$f_i = \begin{cases} 1 & i < i^*, \\ 2^{-2(i-i^*+1)} & i \ge i^*, \end{cases}$$

so f_i is an upper bound on the probability that no leader is elected by the end of the ith iteration. In other words, the probability that the algorithm enters the $(i+1)$th iteration is at most f_i.

Expected time complexity. Since the time spent on the $(i+1)$th iteration is $O(d_{i+1} - d_i)$, the expected time complexity $T(n)$ of the algorithm can be upper bounded as follows.

$$\begin{aligned} T(n) &= O(d_{i^*}) + \sum_{i=i^*}^{\infty} f_i \cdot O(d_{i+1} - d_i) \\ &= O(2^{i^*}) + \sum_{i=i^*}^{\infty} 2^{-2(i-i^*+1)} \cdot O(d_i) \\ &= O(2^{i^*}) + \sum_{i=i^*}^{\infty} O\left(2^{i-2(i-i^*+1)}\right) \\ &= O(2^{i^*}) \cdot O\left(1 + \sum_{i=i^*}^{\infty} 2^{(i-i^*)-2(i-i^*+1)}\right) \\ &= O(2^{i^*}) \cdot O\left(1 + \sum_{j=0}^{\infty} 2^{-2-j}\right) \\ &= O(2^{i^*}) \\ &= O(\log n). \end{aligned}$$

Here we use the fact that $d_{i^*} = 2^{i^*} = O(\log n)$.

Expected energy complexity. The energy cost for the algorithm of one iteration depends on whether a leader is elected in this iteration. Given the parameters $(\widetilde{n}_1, c_1), (\widetilde{n}_2, c_2), \ldots, (\widetilde{n}_k, c_k)$, the energy cost is $O\left(1 + \sum_{j=1}^{k} c_j \widetilde{n}_j^{-1}\right)$ if no leader is elected, and it is $O\left(1 + \sum_{j=1}^{k} c_j \widetilde{n}_j^{-1}\right) + O(\log k)$ if a leader is elected. These bounds hold in the worst case.

With our parameters, the energy cost for the ith iteration is $O\left(1 + \sum_{j=1}^{d_i-1} 2 \cdot 2^{-j}\right) = O(1)$ if no leader is elected in this iteration,

and it is $O\left(1+\sum_{j=1}^{d_i-1} 2\cdot 2^{-j}\right)+O(\log d_i) = O(i)$ if a leader is elected in this iteration. Once a leader is elected, the algorithm is terminated, so the energy cost of the entire algorithm is $O(i)$ if the algorithm is terminated by the end of the ith iteration as a leader is elected in the ith iteration. Therefore, the expected energy cost $E(n)$ of the algorithm can be upper bounded by $O\left(\sum_{i=0}^{\infty} f_i\right)$, and we have

$$E(n) = O\left(\sum_{i=0}^{\infty} f_i\right) = O(i^*)+O\left(\sum_{j=0}^{\infty} 2^{-2-2j}\right) = O(i^*) = O(\log\log n).$$

We summarize the discussion as a theorem.

Theorem 4.1 (Leader election with an unknown number of devices [30]). *There is an algorithm in the* No-CD *model that elects a leader from an unknown number $n \geq 2$ of devices using expected $O(\log n)$ time and expected $O(\log\log n)$ energy.*

REFERENCES

[1] Noga Alon, Amotz Bar-Noy, Nathan Linial, and David Peleg. 1991. A lower bound for radio broadcast. *J. Comput. System Sci.* 43, 2 (1991), 290–298.

[2] Noga Alon, Mohsen Ghaffari, Bernhard Haeupler, and Majid Khabbazian. 2014. Broadcast throughput in radio networks: routing vs. network coding. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms (SODA)*. SIAM, 1831–1843.

[3] Christoph Ambühl. 2005. An optimal bound for the MST algorithm to compute energy efficient broadcast trees in wireless networks. In *International Colloquium on Automata, Languages, and Programming (ICALP)*. Springer, 1139–1150.

[4] Christoph Ambühl. 2008. Minimum Energy Broadcasting in Wireless Geometric Networks. In *Encyclopedia of Algorithms*, Ming-Yang Kao (Ed.). Springer US, Boston, MA, 526–528. https://doi.org/10.1007/978-0-387-30162-4_233

[5] John Augustine, William K Moses Jr, and Gopal Pandurangan. 2022. Distributed MST Computation in the Sleeping Model: Awake-Optimal Algorithms and Lower Bounds. *arXiv preprint arXiv:2204.08385* (2022).

[6] Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. 1992. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *J. Comput. System Sci.* 45, 1 (1992), 104–126.

[7] Leonid Barenboim and Tzalik Maimon. 2021. Deterministic Logarithmic Completeness in the Distributed Sleeping Model. In *35th International Symposium on Distributed Computing (DISC) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 209)*, Seth Gilbert (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 10:1–10:19. https://doi.org/10.4230/LIPIcs.DISC.2021.10

[8] Michael A. Bender, Jeremy T. Fineman, Seth Gilbert, and Maxwell Young. 2018. Scaling Exponential Backoff: Constant Throughput, Polylogarithmic Channel-Access Attempts, and Robustness. *J. ACM* 66, 1, Article 6 (dec 2018). https://doi.org/10.1145/3276769

[9] Michael A. Bender, Tsvi Kopelowitz, Seth Pettie, and Maxwell Young. 2018. Contention Resolution with Constant Throughput and Log-Logstar Channel Accesses. *SIAM J. Comput.* 47, 5 (2018), 1735–1754. https://doi.org/10.1137/17M1158604

[10] Petra Berenbrink, Colin Cooper, and Zengjian Hu. 2009. Energy efficient randomised communication in unknown adhoc networks. *Theoretical Computer Science* 410, 27-29 (2009), 2549–2561.

[11] I. Caragiannis, C. Galdi, and C. Kaklamanis. 2005. Basic computations in wireless networks. In *International Symposium on Algorithms and Computation*. Springer, 533–542.

[12] Yi-Jun Chang, Tsvi Kopelowitz, Seth Pettie, Ruosong Wang, and Wei Zhan. 2019. Exponential Separations in the Energy Complexity of Leader Election. *ACM Transactions on Algorithms* 15, 4, Article 49 (2019). https://doi.org/10.1145/3341111

[13] Yi-Jun Chang, Varsha Dani, Thomas P. Hayes, Qizheng He, Wenzheng Li, and Seth Pettie. 2018. The Energy Complexity of Broadcast. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC)*. ACM, 95–104. https://doi.org/10.1145/3212734.3212774

[14] Yi-Jun Chang, Varsha Dani, Thomas P. Hayes, and Seth Pettie. 2020. The Energy Complexity of BFS in Radio Networks. In *Proceedings of the 39th Symposium on Principles of Distributed Computing (PODC)*. ACM, 273–282. https://doi.org/10.1145/3382734.3405713

[15] Yi-Jun Chang, Ran Duan, and Shunhua Jiang. 2021. Near-Optimal Time-Energy Trade-Offs for Deterministic Leader Election. In *Proceedings of the 33th annual ACM symposium on Parallelism in algorithms and architectures (SPAA)*. ACM.

[16] Yi-Jun Chang and Shunhua Jiang. 2022. The Energy Complexity of Las Vegas Leader Election. *arXiv preprint arXiv:2205.08642* (2022).

[17] Soumyottam Chatterjee, Robert Gmyr, and Gopal Pandurangan. 2020. Sleeping is Efficient: MIS in $O(1)$-Rounds Node-Averaged Awake Complexity. In *Proceedings of the 39th Symposium on Principles of Distributed Computing (PODC)*. ACM, 99–108. https://doi.org/10.1145/3382734.3405718

[18] A. E. F. Clementi, A. Monti, and R. Silvestri. 2003. Distributed broadcast in radio networks of unknown topology. *Theoretical Computer Science* 302, 1 (2003), 337–364.

[19] Varsha Dani, Aayush Gupta, Thomas P. Hayes, and Seth Pettie. 2021. Wake Up and Join Me! An Energy-Efficient Algorithm for Maximal Matching in Radio Networks. *arXiv preprint arXiv:2104.09096* (2021).

[20] Fabien Dufoulon, William K Moses Jr, and Gopal Pandurangan. 2022. Sleeping is Superefficient: MIS in Exponentially Better Awake Complexity. *arXiv preprint arXiv:2204.08359* (2022).

[21] Martín Farach-Colton, Rohan J. Fernandes, and Miguel A. Mosteiro. 2006. Lower Bounds for Clear Transmissions in Radio Networks. In *Proceedings of the 7th Latin American Symposium on Theoretical Informatics (LATIN)*. 447–454. https://doi.org/10.1007/11682462_42

[22] Leszek Gasieniec, Erez Kantor, Dariusz R Kowalski, David Peleg, and Chang Su. 2007. Energy and time efficient broadcasting in known topology radio networks. In *International Symposium on Distributed Computing (DISC)*. Springer, 253–267.

[23] Leszek Gasieniec, Andrzej Pelc, and David Peleg. 2001. The Wakeup Problem in Synchronous Broadcast Systems. *SIAM Journal on Discrete Mathematics* 14, 2 (2001), 207–222. https://doi.org/10.1137/S0895480100376022

[24] Seth Gilbert, Calvin Newport, Nitin Vaidya, and Alex Weaver. 2021. Contention Resolution with Predictions. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing (PODC)*. Association for Computing Machinery, New York, NY, USA, 127–137. https://doi.org/10.1145/3465084.3467911

[25] Tomasz Jurdziński, Mirosław Kutyłowski, and Jan Zatopiański. 2002. Efficient algorithms for leader election in radio networks. In *Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing (PODC)*. 51–57. https://doi.org/10.1145/571825.571833

[26] T. Jurdzinski, M. Kutylowski, and J. Zatopianski. 2002. Energy-Efficient Size Approximation of Radio Networks with No Collision Detection. In *Proceedings of the 8th Annual International Conference on Computing and Combinatorics (COCOON)*. 279–289. https://doi.org/10.1007/3-540-45655-4_31

[27] Tomasz Jurdziński and Grzegorz Stachowiak. 2005. Probabilistic algorithms for the wake-up problem in single-hop radio networks. *Theory of Computing Systems* 38, 3 (2005), 347–367.

[28] Lefteris M Kirousis, Evangelos Kranakis, Danny Krizanc, and Andrzej Pelc. 2000. Power consumption in packet radio networks. *Theoretical Computer Science* 243, 1-2 (2000), 289–305.

[29] Marek Klonowski and Dominik Pajak. 2018. Brief Announcement: Broadcast in Radio Networks, Time vs. Energy Tradeoffs. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PDOC)*. Association for Computing Machinery, New York, NY, USA, 115–117. https://doi.org/10.1145/3212734.3212786

[30] C. Lavault, J.-F. Marckert, and V. Ravelomanana. 2007. Quasi-optimal energy-efficient leader election algorithms in radio networks. *Information and Computation* 205, 5 (2007), 679–693.

[31] Matthew J Miller and Nitin H Vaidya. 2005. A MAC protocol to reduce sensor network energy consumption using a wakeup radio. *IEEE Transactions on mobile Computing* 4, 3 (2005), 228–242.

[32] Koji Nakano and Stephan Olariu. 2000. Randomized leader election protocols in radio networks with no collision detection. In *International Symposium on Algorithms and Computation (ISAAC)*. Springer, 362–373.

[33] Koji Nakano and Stephan Olariu. 2002. Uniform leader election protocols for radio networks. *IEEE transactions on parallel and distributed systems* 13, 5 (2002), 516–526.

[34] Calvin Newport. 2014. Radio Network Lower Bounds Made Easy. In *Proceedings of the 28th International Symposium on Distributed Computing (DISC)*. 258–272. https://doi.org/10.1007/978-3-662-45174-8_18

[35] Brian M Sadler. 2005. Fundamentals of energy-constrained sensor network systems. *IEEE Aerospace and Electronic Systems Magazine* 20, 8 (2005), 17–35.

[36] Ravelomanana Vlady. 2016. Time-optimal and energy-efficient size approximation of radio networks. In *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 233–237.

[37] Yongqiang Wang, Felipe Nunez, and Francis J Doyle. 2012. Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening. *IEEE Transactions on Signal Processing* 60, 10 (2012), 5293–5306.

[38] Dan E. Willard. 1986. Log-Logarithmic Selection Resolution Protocols in a Multiple Access Channel. *SIAM J. Comput.* 15, 2 (1986), 468–477. https://doi.org/10.1137/0215032

[39] Xinyu Zhang and Kang G Shin. 2012. E-MiLi: Energy-minimizing idle listening in wireless networks. *IEEE Transactions on Mobile Computing* 11, 9 (2012), 1441–1454.