Distributed Computing manuscript No.
(will be inserted by the editor)

The Complexity of Leader Election in Diameter-Two Networks

Soumyottam Chatterjee · Gopal Pandurangan · Peter Robinson

the date of receipt and acceptance should be inserted later

Abstract This paper focuses on studying the message complexity of implicit leader election in synchronous distributed networks of diameter two. Kutten et al. [JACM 2015] showed a fundamental lower bound of $\Omega(m)$ (m is the number of edges in the network) on the message complexity of (implicit) leader election that applied also to Monte Carlo randomized algorithms with constant success probability; this lower bound applies for graphs that have diameter at least three. On the other hand, for complete graphs (i.e., graphs with diameter one), Kutten et al. [TCS 2015] established a tight bound of $\tilde{\Theta}(\sqrt{n})$ on the message complexity of randomized leader election (n is the number of nodes in the network). For graphs of diameter two, the complexity was not known.

In this paper, we settle this complexity by showing a tight bound of $\tilde{\Theta}(n)$ on the message complexity of leader election in diameter-two networks. We first give a simple randomized Monte-Carlo leader election algorithm that with high probability (i.e., probability at least $1-n^{-c}$, for some fixed positive constant c) succeeds and uses $O(n\log^3 n)$ messages and runs in $O(1)$ rounds; this algorithm works without knowledge of n (and hence needs no global knowledge). We then show that any algorithm (even Monte Carlo randomized algorithms with large enough constant success probability) needs $\Omega(n)$ messages (even when n is known), regardless of the number of rounds. We also present an $O(n\log n)$ message deterministic algorithm that takes $O(\log n)$ rounds (but needs knowledge of n); we show that this message complexity is tight for deterministic algorithms.

Together with the two previous results of Kutten et al., our results fully characterize the message complexity of leader election vis-à-vis the graph diameter.

Keywords distributed algorithm · leader election · randomized algorithm · message complexity · time complexity · lower bound

1 Introduction

Leader election is a classical and fundamental problem in distributed computing. The leader election problem requires a group of processors in a distributed network to elect a unique leader among themselves, i.e., exactly one processor must output the decision that it is the leader, say, by changing a special *status* component of its state to the value *leader* [14]. All the rest of the nodes must change their status component to the value *non-leader*. These nodes need not be aware of the identity of the leader. This *implicit* variant of leader election is quite standard (cf. [14]), and has been extensively studied (see e.g., [11] and the references therein) and is sufficient in many applications, e.g., for token generation in a token ring environment [13]. In this paper, we focus on this implicit variant. [1]

A preliminary version of this work was nominated for the best paper award at the 19th International Conference on Distributed Computing and Networking (ICDCN 2018).

Gopal Pandurangan was supported, in part, by NSF grants CCF-1527867, CCF-1540512, IIS-1633720, and CCF-1717075 and by US-Israel BSF grant 2016419.

Soumyottam Chatterjee
University of Houston, Houston, TX 77204, USA
E-mail: schatterjee4@uh.edu

Gopal Pandurangan
University of Houston, Houston, TX 77204, USA
E-mail: gopal@cs.uh.edu

Peter Robinson
City University of Hong Kong, Kowloon, Hong Kong
E-mail: peter.robinson@cityu.edu.hk

[1] In another variant, called *explicit* leader election, all the non-leaders change their status component to the value *non-leader*, and moreover, every node must also know the identity of the unique leader. In this variant, $\Omega(n)$ messages is an obvious lower bound (throughout, n denotes the number of nodes in the network) since every node must be

The complexity of leader election, in particular, its message and time complexity, has been extensively studied both in general graphs as well as in special graph classes such as rings and complete networks, see e.g., [14, 17, 19, 20, 12, 11]. While much of the earlier work focused on deterministic algorithms, recent works have studied randomized algorithms (see e.g., [12, 11] and the references therein). Kutten et al. [11] showed a fundamental lower bound of $\Omega(m)$ (m is the number of edges in the network) on the message complexity of (implicit) leader election that applied even to Monte Carlo randomized algorithms with (large-enough) constant success probability; this lower bound applies for graphs *that have diameter at least three*. (They also showed that this bound is tight.) On the other hand, for complete graphs (i.e., graphs of diameter one), Kutten et al. [12] established a tight bound of $\tilde{\Theta}(\sqrt{n})$ on the message complexity of randomized leader election (n is the number of nodes in the network).

For graphs of diameter two, the message complexity was not known. In this paper, we settle this complexity by showing a tight bound of $\tilde{\Theta}(n)$ on the message complexity of leader election in diameter-two networks. Together with the previous results [11, 12], our results fully characterize the message complexity of leader election vis-à-vis the graph diameter (see Table 1).

1.1 Our Results

This paper focuses on studying the message complexity of leader election (both randomized and deterministic) in synchronous distributed networks, in particular, in networks of *diameter two*.

For our algorithms, we assume that the communication is *synchronous* and follows the standard CONGEST model [18], where a node can send in each round at most one message of size $O(\log n)$ bits on a single edge. We assume that the nodes have unique IDs. We assume that all nodes wake up simultaneously at the beginning of the execution. (Additional details on our distributed computation model are given in Section 1.3.)

We show the following results:

1. **Algorithms:** We show that the message complexity of leader election in diameter-two graphs is $\tilde{O}(n)$, by presenting a randomized (implicit) leader election algorithm (cf. Section 2), that takes $O(n\log^3 n)$ messages and runs in $O(1)$ rounds with high probability (whp). [2] This algorithm works even without knowledge of n. While it is easy to design an $O(n\log n)$ messages randomized algorithm with knowledge of n (see Remark 1), not having knowledge of n makes the analysis more involved.
   We also present a *deterministic* algorithm that uses only $O(n\log n)$ messages, but takes $O(\log n)$ rounds. Also this algorithm needs knowledge of n (or at least a constant factor upper bound of $\log n$) (cf. Section 4).
   We note that all our algorithms will work seamlessly for complete networks as well.
2. **Lower Bounds:** We show that, in general, it is not possible to improve over our algorithm substantially, by presenting a lower bound for leader election that applies also to randomized (Monte Carlo) algorithms. We show that $\Omega(n)$ messages are needed for any leader election algorithm (regardless of the number of rounds) in a diameter-two network which succeeds with any constant probability that is strictly larger than $\frac{1}{2}$ (cf. Section 3). This lower bound holds even in the LOCAL model [18], where there is no restriction on the number of bits that can be sent on each edge in each round. To the best of our knowledge, this is the first non-trivial lower bound for randomized leader election in diameter-two networks.
   We also show a simple deterministic reduction that shows that any super-linear message lower bound for complete networks also applies to diameter-two networks as well (cf. Section 5). It can be shown that $\Omega(n\log n)$ messages is a lower bound for deterministic leader election in complete networks [1, 10] (under the assumption that the number of rounds is bounded by some function of n). [3] By our reduction this lower bound also applies to diameter-two networks. [4]

1.2 Technical Overview

All our algorithms exploit the following simple "neighborhood intersection" property of diameter-two graphs: Any two nodes (that are non-neighbors) have at least one neighbor in common (please refer to Observation 1).

Remark 1 Unlike complete networks (which have been extensively studied with respect to leader election — cf. Section 1.5), in diameter-two networks, nodes generally do not have knowledge of n, the network size (in a complete graph, this is trivially known by the degree). This complicates obtaining sublinear in m (where m is the number of edges) message algorithms that are fully localized (don't have knowledge of n).

Indeed, if n is known, the following is a simple randomized algorithm: each node becomes a candidate with probability $\Theta(\frac{\log n}{n})$ and sends its ID to all its neighbors; any node

informed of the leader's identity. Clearly, any lower bound for implicit leader election applies to explicit leader election as well.

[2] Throughout, "with high probability" means with probability at least $1-n^{-c}$, for some fixed positive constant c.

[3] Afek and Gafni[1] show the $\Omega(n\log n)$ message lower bound for complete networks under the non-simultaneous wakeup model in synchronous networks. The same message bound can be shown to hold in the simultaneous wake-up model as well under the restriction that the number of rounds is bounded by a function of n [10].

[4] We point out that lower bounds for complete networks do not directly translate to diameter-two networks.

	RANDOMIZED		DETERMINISTIC	
Diameter	Time	Messages	Time	Messages
$D = 1$: [12, 1]				
Upper Bound	$O(1)$	$O(\sqrt{n}\log^{\frac{3}{2}} n)$	$O(1)$†	$O(n\log n)$ †
Lower Bound	$\Omega(1)$	$\Omega(\sqrt{n})$	$\Omega(1)$	$\Omega(n\log n)$
$D \geq 3$: [11]				
Upper Bound	$O(D)$	$O(m\log\log n)$	$O(D\log n)$	$O(m\log n)$
Lower Bound	$\Omega(D)$	$\Omega(m)$	$\Omega(D)$	$\Omega(m)$
$D = 2$:		**Our Results**		
Upper Bound	$O(1)$	$O(n\log^3 n)$	$O(\log n)$††	$O(n\log n)$$
Lower Bound	$\Omega(1)$	$\Omega(n)$	$\Omega(1)$	$\Omega(n\log n)$

† Note that attaining $O(1)$ time requires $\Omega(n^{1+\Omega(1)})$ messages in cliques, whereas achieving $O(n\log n)$ messages requires $\Omega(\log n)$ rounds; see [1].

$ Needs knowledge of n.

†† Note that it is easy to show an $O(1)$ round deterministic algorithm that takes $O(m)$ messages.

Table 1 Message and time complexity of (implicit) leader election.

that gets one or more messages acts as a “referee" and notifies the candidate that has the smallest ID (among those it has received). The neighborhood intersection property implies that at least one candidate will be chosen uniquely as the leader with high probability.

If n is not known, the above idea does not work. However, we show that if each node v becomes a candidate with probability $\frac{1+\log(d_v)}{d_v}$, (where d_v is the degree of v) then the above idea can be made to work. The main technical difficulty is then showing that at least one candidate is present (cf. Section 2.1) and in bounding the message complexity (cf. Section 2.2). We use Lagrangian optimization to prove that on expectation at least $\Theta(\log n)$ candidates will be selected and then use a Chernoff bound to show a high probability result.

Our $\Omega(n)$ randomized lower bound is inspired by the *bridge crossing* argument of [11] and [16]. In [11], the authors construct a “dumbbell" graph G which is done by taking two identical regular graphs G_1 and G_2, removing an edge from each and adding them as bridge edges between G_1 and G_2 (so that regularity is preserved). The argument is that any leader election algorithm should send at least one message across one of the two bridge edges (bridge crossing); otherwise, it can be shown that the executions in G_1 and G_2 are identical leading to election of two leaders which is not valid. The argument in [11] shows that $\Omega(m)$ messages are needed for bridge crossing. As pointed out earlier in Section 1, this construction makes the diameter of G at least three and hence does not work for diameter-two graphs. To overcome this, we modify the construction that takes two complete graphs and add a set of bridge edges (as opposed to just two); see Fig 1. This creates a diameter-two graph; however, the large number of bridge edges requires a different style of argument and results in a bound different compared to [11]. We show that $\Omega(n)$ messages (in expectation) are needed to send a message across at least one bridge.

We also present a *deterministic* algorithm that requires $O(n\log n)$ messages, but takes $O(\log n)$ rounds. Note that, in a sense, this improves over the randomized algorithm that sends $O(n\log^3 n)$ messages (although, we did not strive to optimize the log factors). However, the deterministic algorithm is slower by a $\log(n)$-factor and is more involved compared to the very simple randomized algorithm (although its analysis is a bit more complicated). Our deterministic algorithm uses ideas similar to Afek and Gafni's [1] leader election algorithm for complete graphs; however, the algorithm is a bit more involved. Our algorithm assumes knowledge of n (this is trivially true in complete networks, since every node can infer n from its degree) which is needed for termination. It is not clear if one can design an $O(n\log n)$ messages algorithm (running in say $O(\log n)$ rounds) that does not need knowledge of n, which is an interesting open question (cf. Section 6).

Finally, we present a simple reduction that shows that superlinear (in n) lower bounds in complete networks also imply lower bounds for diameter-two networks, by showing how using only $O(n)$ messages and $O(1)$ rounds, a complete network can be converted to a diameter-two network in a distributed manner. This shows that our deterministic algorithm (cf. Section 4) is message optimal.

1.3 Distributed Computing Model

The model we consider is similar to the models of [1, 4, 6, 8, 9], with the main addition of giving processors access to a private unbiased coin. We consider a system of n nodes, represented as an undirected graph $G = (V, E)$. In this paper, we focus on graphs with diameter $D(G) = 2$, where $D(G)$ is the diameter of $G = (V, E)$. An obvious consequence of this is that G is connected, therefore $m \geq n - 1$, where $m = |E|$ and $n = |V|$. Also, since G is not a complete graph, $m < \frac{n(n-1)}{2}$.

Each node has a unique identifier (ID) of $O(\log n)$ bits and runs an instance of a distributed algorithm. The computation advances in synchronous rounds where, in every round, nodes can send messages, receive messages that were sent in the same round by neighbors in G, and perform some local computation. Every node has access to the outcome of unbiased private coin flips (for randomized algorithms). Messages are the only means of communication; in particular, nodes cannot access the coin flips of other nodes, and do not share any memory. Throughout this paper, we assume that all nodes are awake initially and simultaneously start executing the algorithm. We note that initially nodes have knowledge only of themselves, in other words we assume the *clean network model* — also called the *KT0 model* [18] which is standard and most commonly used. [5]

1.4 Leader Election: Problem Definition

We formally define the leader election problem here.

Every node u has a special variable $\texttt{status}_u$ that it can set to a value in

$$\{\perp, \text{NON-ELECTED}, \text{ELECTED}\};$$

initially we assume $\texttt{status}_u = \perp$.

An *algorithm A solves leader election in T rounds* if, from round T onwards, exactly one node has its status set to ELECTED while all other nodes are in state NON-ELECTED. This is the requirement for standard (implicit) leader election. For *explicit* leader election, we further require that all non-leader nodes should know the identity of the leader.

1.5 Other Related Works

The complexity of the leader election problem and algorithms for it, especially deterministic algorithms (guaranteed to always succeed), have been well-studied. Various algorithms and lower bounds are known in different models with synchronous (as well as asynchronous) communication and in networks of varying topologies such as a cycle, a complete graph, or some arbitrary topology (e.g., see [5, 14, 17, 19, 20, 12, 11] and the references therein).

The study of leader election algorithms is usually concerned with both message and time complexity. We discuss two sets of results, one for complete graphs and the other for general graphs. As mentioned earlier, for complete graphs, Kutten et al. [12] showed that $\tilde{\Theta}(\sqrt{n})$ is the tight message complexity bound for randomized (implicit) leader election. In particular, they presented an $O(\sqrt{n}\log^{3/2} n)$ messages algorithm that ran in $O(1)$ rounds; they also showed an almost matching lower bound for randomized leader election, showing that $\Omega(\sqrt{n})$ messages are needed for any leader election algorithm that succeeds with a sufficiently large constant probability.

For deterministic algorithms on complete graphs, it is known that $\Theta(n \log n)$ is a tight bound on the message complexity [1, 10]. In particular, Afek and Gafni [1] presented an $O(n \log n)$ messages algorithm for complete graphs that ran in $O(\log n)$ rounds. For complete graphs, Korach et al. [7] and Humblet [4] also presented $O(n \log n)$ message algorithms. Afek and Gafni [1] presented asynchronous and synchronous algorithms, as well as a tradeoff between the message and the time complexity of synchronous *deterministic* algorithms for complete graphs: the results varied from a $O(1)$-time, $O(n^2)$-messages algorithm to a $O(\log n)$-time, $O(n \log n)$-messages algorithm. Afek and Gafni [1], as well as [7, 9] showed a lower bound of $\Omega(n \log n)$ messages for *deterministic* algorithms in the general case. [6]

For general graphs, the best known bounds are as follows. Kutten et al. [11] showed that $\Omega(m)$ is a very general lower bound on the number of messages and $\Omega(D)$ is a lower bound on the number of rounds for any leader election algorithm. It is important to point out that their lower bounds applied for graphs with *diameter at least three*. Note that these lower bounds hold even for randomized Monte Carlo algorithms that succeed even with (some large enough, but) constant success probability and apply even for implicit leader election. Earlier results, showed such lower bounds only for deterministic algorithms and only for the restricted case of comparison algorithms, where it was also required that nodes may not wake up spontaneously and that D and n were not known. The $\Omega(m)$ and $\Omega(D)$ lower bounds are *uni-*

[5] If one assumes the *KT1 model*, where nodes have an initial knowledge of the IDs of their neighbors, there exists a trivial algorithm for leader election in a diameter-two graph that uses only $O(n)$ messages.

[6] This lower bound assumes non-simultaneous wakeup though. If nodes are assured to wake up at the same time in synchronous complete networks, there exists a trivial algorithm: if a node's identity is some i, it waits i time before it sends any message then leader election could be solved (deterministically) in $O(n)$ messages on complete graphs in synchronous networks. Recently Kutten [10] shows that the $\Omega(n \log n)$ lower bound holds for simulataneous wakeup as well, if the number of rounds is bounded.

versal in the sense that they hold for all universal algorithms (namely, algorithms that work for all graphs), apply to every $D \geq 3$, m, and n, and hold even if D, m, and n are known, all the nodes wake up simultaneously, and the algorithms can make any use of node's identities. To show that these bounds are tight, they also present an $O(m)$ messages algorithm (this algorithm is not time-optimal). An $O(D)$ time leader election algorithm is known [17] (this algorithm is not message-optimal). They also presented an $O(m \log\log n)$ messages randomized algorithm that ran in $O(D)$ rounds (where D is the network diameter) that is simultaneously almost optimal with respect to both messages and time. They also presented an $O(m \log n)$ and $O(D \log n)$ deterministic leader election algorithm for general graphs.

Whereas most previous works characterized the message complexity of leader election in terms of a graph's density, the recent work of [2] showed bounds for leader election in terms of the graph conductance. In particular, they provide a class of graphs with conductance ϕ, and show that $\Omega(\sqrt{n} \cdot \phi^{-\frac{3}{4}})$ messages are required for solving leader election with high probability. Note that this does not contradict the message complexity bound of $\Omega(m)$ of [11], since the latter was shown by leveraging a bottleneck construction that has very small conductance. [2] also provides a leader election algorithm that uses $O(\sqrt{n} \log^{\frac{7}{2}} n \cdot t_{\text{mix}})$ messages and $O(t_{\text{mix}} \log^2 n)$ time in any graph, even when nodes do not have any knowledge of the mixing time t_{mix}.

2 A Randomized Algorithm

In this section, we present a simple randomized Monte Carlo algorithm that works in a constant number of rounds. Algorithm 1 is entirely local, as nodes do not require any knowledge of n. Nevertheless, we show that we can sub-sample a small number of candidates (using only local knowledge) that then attempt to become leader. In the remainder of this section, we prove the following result.

Theorem 1 *There exists a Monte Carlo randomized leader election algorithm that, with high probability, succeeds in n-node networks of diameter at most two in $O(1)$ rounds, while sending $O(n \log^3 n)$ messages.*

2.1 Proof of Correctness: Analyzing the number of candidates selected

We use the following property of diameter-two graphs crucially in our algorithm.

Observation 1 Let $G = (V, E)$ be a graph of diameter two. Then for any $u, v \in V$, either $(u, v) \in E$ or $\exists w \in V$ such that $(u, w) \in E$ and $(v, w) \in E$, i.e., u and v have at least one common neighbor w (say).

We note that if one or more candidates are selected, then only the candidate node with the minimum ID is selected as the leader. That is, the leader is unique, and therefore the algorithm produces the correct output. The only case when the algorithm may be wrong is if no candidates are selected to begin with, in which case no leader is selected. In this section, we show that, with high probability, at least two candidates are selected. We note that having only one candidate is sufficient for our purposes, so having two (guaranteed by Lemma 3) or more candidates is actually even better, informally speaking.

We make use of the following fact in order to show that.

Lemma 1 *Let $f(x_1, x_2, \ldots, x_n)$ be a function of n variables $x_1, x_2, \ldots, x_n$, where $x_1, x_2, \ldots, x_n$ are positive reals. f is defined as*

$$f(x_1, x_2, \ldots, x_n) \overset{\text{def}}{=} \sum_{i=1}^{n} \frac{1 + \log(x_i)}{x_i}.$$

Let C be a real number $\geq n\sqrt{2}$. Then $f(x_1, x_2, \ldots, x_n)$ is minimized, subject to the constraint $\sum_{i=1}^{n} x_i = C$, when $x_i = \frac{C}{n}$, for all $1 \leq i \leq n$. The minimum value that $f(x_1, x_2, \ldots, x_n)$ takes is at the point

$(\frac{C}{n}, \frac{C}{n}, \ldots, \frac{C}{n})$, *and is given by*

$$f^{min} = f(\frac{C}{n}, \frac{C}{n}, \ldots, \frac{C}{n}) = \frac{n^2}{C}(1 + \log(\frac{C}{n})).$$

Proof We use standard Lagrangian optimization techniques to show this. Please refer to the appendix for the full proof.

Lemma 2 *Let X be a random variable that denotes the total number of candidates selected in Algorithm 1. Then for any $n \geq 119$, the expected number of selected candidates is lower-bounded by*

$$E[X] > 2 + \tfrac{1}{2} \log n.$$

Proof Let X_v be an indicator random variable that takes the value 1 if and only if v becomes a candidate. Then $\mathrm{E}[X_v] = \Pr[X_v = 1] = \frac{1+\log(d_v)}{d_v}$. Thus if X denotes the total number of candidates selected, then

$$\mathrm{E}[X] = \sum_{v \in V} \mathrm{E}[X_v] = \sum_{v \in V} \frac{1 + \log(d_v)}{d_v}.$$

Since G is connected, $m \geq n - 1 \implies 2m \geq 2n - 2 > n\sqrt{2}$, i.e., the precondition for the applicability of Lemma 1 is satisfied.

Thus by By Lemma 1, $\mathrm{E}[X]$ is minimized subject to the constraint

$$\sum_{v \in V(G)} d_v = 2m,$$

when $d_v = \frac{2m}{n}$ for all $v \in V(G)$, i.e., when G is regular.

Algorithm 1 Randomized leader election in $O(1)$ rounds and $O(n\log^3 n)$ message complexity

```
1: Each node $v \in V$ selects itself to be a "candidate" with probability $\frac{1+\log(d_v)}{d_v}$, where $d_v$ is the degree of $v$.
2: if $v$ becomes a candidate then $v$ sends its ID to all its neighbors.
3: Each node acts as a "referee node" for all its candidate neighbors (including, possibly itself).
4: If a node $w$ receives ID's from its neighbors $v_1, v_2, \ldots, v_j$ (say), then $w$ computes the minimum ID of those and sends it back to those neighbors. That is, $w$ sends $\min\{ID(v_1), ID(v_2), \ldots, ID(v_j)\}$ back to each of $v_1, v_2, \ldots, v_j$.
5: A node $v$ decides that it is the leader if and only if it receives its own ID from all its neighbors. Otherwise $v$ decides that it is not the leader.
```

Case 1 ($n-1 \le m \le n^{\frac{3}{2}}$): The minimum value that $\mathrm{E}[X]$ takes is given by

$$\begin{aligned}\mathrm{E}[X]|_{\min} &= \frac{n^2}{2m}(1+\log(\frac{2m}{n})) \\ &> \frac{n^2}{2m} && (\text{since } 1+\log(\tfrac{2m}{n}) > 1) \\ &\ge \frac{\sqrt{n}}{2} && (\text{since } m \le n^{\frac{3}{2}})\end{aligned}$$

Case 2 ($n^{\frac{3}{2}} < m \le \binom{n}{2}$): The minimum value that $\mathrm{E}[X]$ takes is given by

$$\begin{aligned}\mathrm{E}[X]|_{\min} &= \frac{n^2}{2m}(1+\log(\frac{2m}{n})) \\ &> 1+\log(\frac{2n^{\frac{3}{2}}}{n}) && (\text{since } \tfrac{n^2}{2m} > 1 \text{ and } m > n^{\frac{3}{2}}) \\ &= 1+\log 2+\log(n^{\frac{1}{2}}) = 2+\frac{1}{2}\log n.\end{aligned}$$

Finally, we note that $\frac{\sqrt{n}}{2} > 2+\frac{1}{2}\log n$ for all $n \ge 119$, and this completes the proof.

Lemma 3 *If X denotes the number of candidates selected, then* $\Pr[X \le 1] < n^{-\frac{1}{4}}$.

Proof We set $\delta = \frac{2+\log n}{4+\log n}$. Then clearly $0 < \delta < 1$, and $1-\delta = \frac{2}{4+\log n}$. Also, substituting the lower bound for μ from Lemma 2, we have that

$$(1-\delta)\mu > 1. \tag{1}$$

We want to apply Chernoff bound, so we first compute a lower bound for the quantity $\frac{\mu\delta^2}{2}$:

$$\begin{aligned}\frac{\mu\delta^2}{2} &= (\frac{\mu}{2})\cdot\delta^2 \\ &> (\frac{4+\log n}{2})\cdot(\frac{2+\log n}{4+\log n})^2 \\ &\qquad (\text{since } \mu > \tfrac{4+\log n}{2} \text{ from Lemma 2}) \\ &= \frac{(2+\log n)^2}{4(4+\log n)} = \frac{4+\log n(4+\log n)}{4(4+\log n)} \\ &= \frac{1}{4+\log n}+\frac{\log n}{4} > \frac{\log n}{4}\end{aligned} \tag{2}$$

Hence from Equation 1, we have that

$$\begin{aligned}\Pr[X \le 1] &\le \Pr[X \le (1-\delta)\mu] \\ &\le \exp(-\frac{\mu\delta^2}{2}) \\ &\qquad (\text{by Chernoff bound [15, Theorem 4.5(2)]}) \\ &< \exp(-\frac{\log n}{4}) \\ &\qquad (\text{since } \tfrac{\mu\delta^2}{2} > \tfrac{\log n}{4} \text{ from Equation (2)}) \\ &< 2^{-\frac{\log n}{4}} = n^{-\frac{1}{4}}.\end{aligned}$$

2.2 Computing the message complexity

We use the following variant of Chernoff Bound [15, Theorem 4.4(3)] in the following analysis.

Theorem 2 (Chernoff Bound) *Let $X_1, X_2, \ldots, X_n$ be independent indicator random variables, and let $X = \sum_{i=1}^{n} X_i$. Then the following Chernoff bound holds: for $R \ge 6E[X]$,* $\Pr[X \ge R] \le 2^{-R}$.

Now on to the analysis.

Computing the expected total message complexity of the algorithm. Note that the expected total message complexity of the algorithm can be bounded as follows. Let M^{entire} be a random variable that denotes the total messages sent during the course of the algorithm. Let M_v be the number of messages sent by node v. Thus

$$M^{\text{entire}} = \sum_{v \in V} M_v.$$

A node v becomes a candidate with probability $\frac{1+\log(d_v)}{d_v}$ and, if it does, it sends d_v messages (the referee nodes reply to these, but that increases the total number of messages by only a factor of 2). Hence by linearity of expectation, it follows that

$$\begin{aligned}\mathrm{E}[M^{\text{entire}}] &= \sum_{v \in V} \mathrm{E}[M_v] = \sum_{v \in V} 2\frac{1+\log(d_v)}{d_v} d_v \\ &= 2\sum_{v \in V}(1+\log(d_v)) \le 2\sum_{v \in V}(1+\log n) \\ &\le 2n + 2n\log n.\end{aligned}$$

Showing concentration by introducing the idea of "buckets". To show concentration, we cannot directly apply a standard Chernoff bound, since that works for 0-1 random variables only, whereas the M_vs are not 0-1 random variables (they take values either 0 or d_v). To handle this, we "bucket" the degrees into different categories, called "buckets", based on their value; we then use a Chernoff bound as detailed below.

Definition 1 Let k be a positive integer such that $2^{k-1} < n \le 2^k$. For $0 \le j \le k$, let $V_j \subset V$ be the set of vertices with degree in $(2^{j-1}, 2^j]$, i.e., if $v \in V_j$, then $2^{j-1} < d_v \le 2^j$. Thus

$$\begin{aligned}V_0 &\overset{\text{def}}{=} \{v \in V \mid d_v = 1\}, \\ V_1 &\overset{\text{def}}{=} \{v \in V \mid d_v = 2\}, \\ V_2 &\overset{\text{def}}{=} \{v \in V \mid d_v = 3 \text{ or } d_v = 4\}, \\ V_3 &\overset{\text{def}}{=} \{v \in V \mid d_v \in \{5,6,7,8\}\}, \text{ and so on.}\end{aligned}$$

Remark 2 We note that

$$\sum_{j=0}^{k} n_j = n, \text{ where } n_j = |V_j| \text{ for } 0 \le j \le k.$$

In particular, $n_j \le n$ for all $j \in [0,k]$.

Intuition behind this idea of "bucketing".

- Nodes are bucketed according to the logarithm of their respective degree (ceiling of the logarithm, to be more precise) (cf. Definition 1). Thus nodes in the same bucket send approximately the same amount of messages.
- Nodes in a higher-degree bucket will send a higher number of messages *each*, than nodes in a lower-degree bucket. The greater effect of a higher-degree bucket, however, is counterbalanced by the the smaller number (in expectation) of candidates in that bucket.
- Thus the number of messages contributed by each bucket — whether high-degree or low-degree — is approximately the same (cf. Lemma 4).
- We can, therefore, easily "add" the contributions made by each bucket — there are at most $(\lceil \log n \rceil + 1)$ such buckets — and obtain the combined (total) message complexity (cf. Lemma 5).

Counting the number of messages sent in the first round by each individual node.

1. **Analyzing vertices with degree ≤ 2:** We recall that X_v is an indicator random variable that takes the value 1 if and only if v becomes a candidate. Then $\Pr[X_v = 1] = 1$ if $v \in V_0 \cup V_1$, i.e., every vertex with degree 1 or degree 2 selects itself to be a candidate, deterministically.

   For $v \in V$, let m_v denote the number of messages that v sends. So $m_v = d_v$ if v becomes a candidate, and $m_v = 0$ otherwise. Let M_j be the total number of messages that members of V_j send, i.e.,

   $$\begin{aligned}M_j &\overset{\text{def}}{=} \sum_{v \in V_j} m_v \le \sum_{v \in V_j} d_v \\ &\le \sum_{v \in V_j} 2^j = n_j.2^j \le n.2^j \\ &\implies M_0 \le n \text{ and } M_1 \le 2n.\end{aligned}$$

2. **Analyzing vertices with degree > 2:** We recall that for $v \in V$, X_v is an indicator random variable that takes the value 1 if and only if v becomes a candidate. Let i be an integer in $[2,k]$ and let $v \in V_i$. Then

   Observation 2 $\frac{i}{2^i} < \mathrm{E}[X_v] < \frac{3i}{2^i}$.

   Proof For $v \in V_i$, $2^{i-1} < d_v \le 2^i$. So

   $$\begin{aligned}\mathrm{E}[X_v] &= \Pr[X_v = 1] \\ &\qquad \text{(since } X_v \text{ is an indicator random variable)} \\ &= \frac{1+\log(d_v)}{d_v} \\ \implies & \frac{1+\log(2^{i-1})}{2^i} < \mathrm{E}[X_v] < \frac{1+\log(2^i)}{2^{i-1}} \\ &\qquad \text{(since } 2^{i-1} < d_v \le 2^i\text{)} \\ \text{or, } & \frac{i}{2^i} < \mathrm{E}[X_v] < \frac{i+1}{2^{i-1}} \le \frac{3i}{2^i} \\ &\qquad \text{(since } i \ge 2 \implies \tfrac{3i}{2} \ge i+1\text{)}\end{aligned}$$

   For $0 \le j \le k$, let Y_j be a random variable that denotes the total number of candidates selected from V_j.

Observation 3 For $2 \leq i \leq k$, $\frac{in_i}{2^i} < \mathrm{E}[Y_i] < \frac{3in_i}{2^i}$.

Proof

$$Y_i = \sum_{v \in V_i} X_v \implies \mathrm{E}[Y_i] = \mathrm{E}[\sum_{v \in V_i} X_v]$$
$$= \sum_{v \in V_i} \mathrm{E}[X_v] \qquad \text{(by linearity of expectation)}$$
$$\implies \sum_{v \in V_i} \frac{i}{2^i} < \mathrm{E}[Y_i] < \sum_{v \in V_i} \frac{3i}{2^i}$$
$$\implies \frac{in_i}{2^i} < \mathrm{E}[Y_i] < \frac{3in_i}{2^i}.$$

Remark 3 $\forall u, v \in V(G)$, $u \neq v$, X_u and X_v are independent, and for $0 \leq j \leq k$, we define Y_j as

$$Y_j = \sum_{v \in V_j} X_v,$$

i.e., Y_j is a sum of independent, 0-1 random variables. Hence we can use Theorem 2 to show that Y_j is concentrated around its mean (expectation).

We recall that for $0 \leq j \leq k$, M_j is the total number of messages that members of V_j send, i.e., for $2 \leq i \leq k$,

$$M_i = \sum_{v \in V_i} m_v = \sum_{v \in V_i, X_v = 1} d_v.$$

Lemma 4 *For any integer $i \in [2,k]$, it holds that*

$$\Pr[M_i \geq 24n \log^2 n] \leq \tfrac{1}{n^4}.$$

Proof

$$M_i = \sum_{v \in V_i} m_v = \sum_{v \in V_i, X_v = 1} d_v$$
$$\implies \sum_{v \in V_i, X_v = 1} 2^{i-1} < M_i \leq \sum_{v \in V_i, X_v = 1} 2^i \qquad \text{(since } 2^{i-1} < d_v \leq 2^i\text{)}$$
$$\implies 2^{i-1}.Y_i < M_i \leq 2^i.Y_i.$$

Case 1 **($\mathbf{E}[Y_i] = 0$):** $\mathrm{E}[Y_i] = 0$ if and only if $n_i = 0$, i.e., if and only if $\nexists v \in V$ such that $2^{i-1} < d_v \leq 2^i$. But $n_i = 0 \implies V_i = \phi$, the empty set. Therefore, $M_i = 0$.

Case 2 **($0 < \mathbf{E}[Y_i] < 1$):** Assuming $n \geq 3$, $4 \log n > 6 > 6\mathrm{E}[Y_i]$. Therefore, by Theorem 2,

$$\Pr[Y_i \geq 4 \log n] \leq 2^{-4 \log n} = n^{-4}$$
$$\implies \Pr[M_i \geq 2^i.4 \log n] \leq n^{-4} \qquad \text{(since } M_i \leq 2^i.Y_i\text{)}$$
$$\implies \Pr[M_i \geq 8n \log n] \leq n^{-4} \quad \text{(since } i \leq k < \log n + 1\text{)}$$

Case 3 **($\mathbf{E}[Y_i] \geq 1$):** We have shown before that $\mathrm{E}[Y_i] \leq \frac{3in_i}{2^i}$. But $n_i \leq n$ for all $2 \leq i \leq k$. Hence $\mathrm{E}[Y_i] \leq \frac{3ni}{2^i}$. Assuming $n \geq 3$, $4 \log n > 6$. Therefore, by Theorem 2,

$$\Pr[Y_i \geq 12n \log n. \frac{i}{2^i}] \leq \Pr[Y_i \geq 4 \log n \mathrm{E}[Y_i]]$$
$$\leq 2^{-4 \log n \mathrm{E}[Y_i]} = n^{-4\mathrm{E}[Y_i]} \leq n^{-4} \qquad \text{(since } \mathrm{E}[Y_i] \geq 1\text{)}$$
$$\implies \Pr[M_i \geq 12in \log n] \leq n^{-4} \qquad \text{(since } M_i \leq 2^i.Y_i\text{)}$$

But we have, $i \leq k < \log n + 1 < 2 \log n \implies 12in \log n < 24n \log^2 n$. Hence

$$\Pr[M_i \geq 24n \log^2 n] \leq \Pr[M_i \geq 12in \log n] \leq n^{-4}.$$

Combining the effects of all the nodes.

Lemma 5 *If M denotes the total number of messages sent by the candidates (in the first round only), then*

$$\Pr[M \geq 27n \log^3 n] < \tfrac{1}{n^3}.$$

Proof

$$M \overset{\mathsf{def}}{=} \sum_{i=0}^{k} M_i = M_0 + M_1 + \sum_{i=2}^{k} M_i$$
$$\leq n + 2n + \sum_{i=2}^{k} M_i \qquad \text{(since } M_0 \leq n \text{ and } M_1 \leq 2n\text{)}$$
$$= 3n + \sum_{i=2}^{k} M_i.$$

But for $2 \leq i \leq k$, $\Pr[M_i \geq 24n \log^2 n] \leq \frac{1}{n^4}$. Taking the union bound over $2 \leq i \leq k$,

$$\Pr[M_{i'} \geq 24n \log^2 n] \text{ for some } i' \in [2,k] \text{ is } \leq \frac{\log n}{n^4} < \frac{1}{n^3}$$
$$\implies \Pr[\sum_{i=2}^{k} M_i \geq 24n \log^3 n] < \frac{1}{n^3}$$
$$\implies \Pr[3n + \sum_{i=2}^{k} M_i \geq 3n + 24n \log^3 n] < \frac{1}{n^3}$$
$$\implies \Pr[M \geq 3n + 24n \log^3 n] < \frac{1}{n^3} \qquad \text{(since } M \leq 3n + \textstyle\sum_{i=2}^{k} M_i\text{)}$$

But $3n \leq 3n \log^3 n$ for $n \geq 2$, so, $3n + 24n \log^3 n \leq 27n \log^3 n$. Hence

$$\Pr[M \geq 27n \log^3 n] \leq \Pr[M \geq 3n + 24n \log^3 n] < \frac{1}{n^3}.$$

Lemma 6 *If M^{entire} denotes the total number of messages sent during the entire run of Algorithm 1, then*

$$\Pr[M^{entire} \geq 54n \log^3 n] < \tfrac{1}{n^3}.$$

Proof Let M' denote the number of messages sent by the "referee" nodes in the *second* round of the algorithm. We recall that M is the number of messages sent by the "candidate" nodes in the *first* round of the algorithm. Then $M' \leq M$, and $M^{\text{entire}} = M + M' \leq 2M$, and the result follows.

This completes the proof of Theorem 1.

3 A Lower Bound for Randomized Algorithms

In this section we show that $\Omega(n)$ is a lower bound on the message complexity for solving leader election with any randomized algorithm in diameter-two networks. Notice that [12] shows a lower bound of $\Omega(\sqrt{n})$ for the special case of diameter-one networks, and we know from [11] that the message complexity becomes $\Omega(m)$ for (most) diameter-three networks. Thus Theorem 3 completes the picture regarding the message complexity of leader election when considering networks according to their diameter.

Theorem 3 *Any algorithm that solves implicit leader election with probability at least $\frac{1}{2}+\varepsilon$ in any n-node network with diameter at most two, for any constant $\varepsilon > 0$, sends at least $\Omega(n)$ messages in expectation. This holds even if nodes have unique IDs and know the network size n.*

In the remainder of this section, we prove Theorem 3.

Proof by contradiction. Assume towards a contradiction, that there is an algorithm that elects a leader with probability $\frac{1}{2}+\varepsilon$ that sends $o(n)$ messages with probability approaching 1. In other words, we assume that the event where the algorithm sends at least $\Omega(n)$ messages (of arbitrary size) happens with probability at most $o(1)$.

Unique IDs vs. Anonymous. Before describing our lower bound construction, we briefly recall a simple reduction used in [12] that shows that assuming unique IDs does not change the success probability of the algorithm by more than $\frac{1}{n}$: Since we assume that nodes have knowledge of n, it is straightforward to see that nodes can obtain unique IDs (whp) by choosing a random integer in the range $[1,n^c]$, for some constant $c \geq 4$. Thus, we can simulate an algorithm that requires unique IDs in the anonymous case and the simulation will be correct with high probability. Suppose that there is an algorithm A that can break the message complexity bound of Theorem 3 while succeeding with probability $\geq \frac{1}{2}+\varepsilon$, for some positive constant ε, when nodes have unique IDs. Then, the above simulation yields an algorithm A' that works in the case where nodes are *anonymous* with the same message complexity bound as algorithm A and succeeds with probability at least $(\frac{1}{2}+\varepsilon-\frac{1}{n}) \geq \frac{1}{2}+\varepsilon'$, for some positive constant ε'. We conclude that proving the lower bound for the anonymous case is sufficient to imply a lower bound for the case where nodes do have unique IDs.

The Lower Bound Graph. Our lower bound is inspired by the bridge crossing argument of [11] and [16]. For simplicity, we assume that $\frac{n}{4}$ is an integer. Consider two cliques C_1 and C_2 of $\frac{n}{2}$ nodes each and let G' be the n-node graph consisting of the two (disjoint) cliques. The *port numbering* of an edge $e=(u_i,v_j) \in E(G')$ refers to the port number at u_i and the respective port number at v_j that connects e. The port numberings of the edges defines an *instance of G'*.

Given an instance of G', we will now describe how to obtain an instance of graph G that has the same node set as G'. Fix some arbitrary enumeration $u_1,\dots,u_{\frac{n}{2}}$ of the nodes in C_1 and similarly let $v_1,\dots,v_{\frac{n}{2}}$ be an enumeration of the nodes in C_2.[7] To define the edges of G, we randomly choose a maximal matching M_1 of $\frac{n}{4}$ edges in the subgraph C_1. Consider the set of edges $M_2' = \{(v_i,v_j) \mid \exists (u_i,u_j) \in M_1\}$, which we can think of as a mapping of the matching M_1 to C_2. Then, we define M_2 to be a randomly chosen maximal matching on C_2 restricted to the edges in $E(G') \setminus M_2'$. Next, we remove all edges in $M_1 \cup M_2$ from G'. So far, we have obtained a graph where each node has one *unwired port*.

The edge set of G consists of all the remaining edges of G' in addition to the set of *bridge edges*

$$M = \{(u_1,v_1),\dots,(u_{\frac{n}{2}},v_{\frac{n}{2}})\},$$

where we connect these bridge edges by using the unwired ports that we obtained by removing the edges as described above. We say that an edge is an *intra-clique edge* if it has both endpoints in either C_1 or C_2. Observe that the intra-clique edges of G are a subset of the edges of G'. Figure 1 gives an illustration of this construction.

Lemma 7 *Graph G is an n-node network of diameter two and the port numbering of each intra-clique edge in G is the same as of the corresponding edge in G'.*

Proof By construction, each node in C_1 has the same port numbering in both graphs, except for its (single) incident edge that was replaced with a bridge edge to some node in C_2. Thus we focus on showing that G has diameter two.

We will show that node $u_i \in C_1$ has a path of length two to every other node in G. Observe that any two nodes $u_i,u_j \in C_1$ each have $\frac{n}{2}-2$ incident intra-clique edges and since $\frac{n}{2}-2 > \frac{|C_1|}{2}$ they must have a common neighbor. Now, consider some node $v_j \in C_2$ and assume that $j \neq i$, as otherwise there is the bridge edge $(u_i,v_i) \in M \subseteq E(G)$, and we are done. If $(u_i,u_j) \in E(G)$, then the result follows because $(u_j,v_j) \in M$. Thus, assume $(u_i,u_j) \notin E(G)$. It follows that there is a path $u_i \to v_i \to v_j$ in G, since, by construction, the edge $(v_i,v_j) \in M_2'$ and $(v_i,v_j) \notin M_2$, thus $(v_i,v_j) \in E(G)$.

A symmetric argument shows that every node has distance ≤ 2 from a given node in C_2.

A *state* σ of the nodes in C_1 is a $\frac{n}{2}$-size vector of the local states of the $\frac{n}{2}$ nodes in C_1. Since we assume that nodes are anonymous, a state σ that is reached by the nodes in C_1, can also be reached by the nodes in C_2. More formally, when executing the algorithm on the disconnected network G', we can observe that every possible state σ (of $\frac{n}{2}$ nodes) has the same

[7] This enumeration is used only for the description of the lower bound construction and is unbeknownst to the nodes.

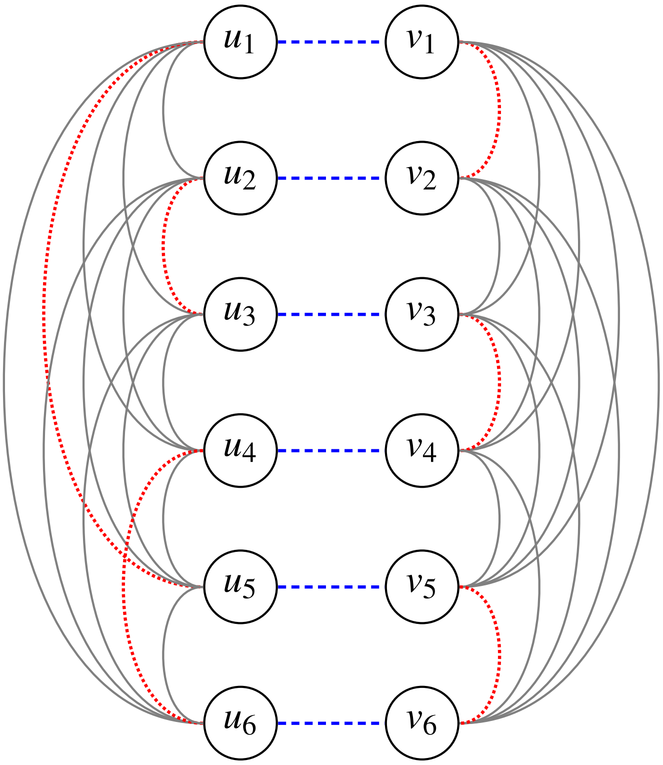

Fig. 1 The lower bound graph construction used in Theorem 3 for $n = 12$, with cliques C_1 and C_2, where $V(C_1) = \{u_1, \ldots, u_6\}$ and $V(C_2) = \{v_1, \ldots, v_6\}$. The dotted red edges are the edges in M_1 and M_2 that are removed from C_1 and C_2 when constructing G and the blue dashed inter-clique edges are given by the maximal matching M between C_1 and C_2. Each blue edge incident to some node u_i is connected by using the port number of u_i's (removed) red edge.

probability of occurring in C_1 as in C_2. Thus, a state where there is exactly one leader among the $\frac{n}{2}$ nodes of a clique in G', is reached with some specific probability q depending on the algorithm. By a slight abuse of notation, we also use G' and G to denote the event that the algorithm executes on G' respectively G. For the probability of the event One, which occurs when there is exactly 1 leader among the n nodes, we get

$$\Pr\left[\text{One} \mid G'\right] = 2q(1-q) \le \frac{1}{2}, \tag{3}$$

which holds for any value of q. Since G' is disconnected, the algorithm does not need to succeed with nonzero probability when being executed on G'. However, below we will use this observation to obtain an upper bound on the probability of obtaining (exactly) one leader in G.

Now consider the execution on the diameter-two network G (obtained by modifying the ports of G' as described above) and let $C_1 \not\leftrightarrow C_2$ be the event that no message is sent across the bridges between C_1 and C_2. Since we assume the port numbering model where nodes are unaware of their neighbors initially, it follows by Lemma 7 that

$$\Pr\left[\text{One} \mid C_1 \not\leftrightarrow C_2, G\right] = \Pr\left[\text{One} \mid G'\right]. \tag{4}$$

Let M be the event that the algorithm sends $o(n)$ messages. Recall that we assume towards a contradiction that $\Pr[M \mid G] = 1 - o(1)$.

Lemma 8 $\Pr[C_1 \leftrightarrow C_2 \mid G, M] = o(1)$.

Proof The proof is inspired by the guessing game approach of [3] and Lemma 16 in [16]. Initially, any node $u \in C_1$ has $\frac{n}{2} - 1$ ports that are all equally likely (i.e., with probability $\frac{1}{\frac{n}{2}-1}$) to be connected to the (single) bridge edge incident to u. As u sends messages to other nodes, it might learn about some of its ports connecting to non-bridge edges and hence this probability can increase over time. However, we condition on event M, i.e., the algorithm sends at most $o(n)$ messages in total and hence at least $\frac{n}{4}$ ports of each node u remain unused at any point.

It follows that the probability of some node u to send a message over a (previously unused) port that connects a bridge edge is at most $\frac{4}{n}$ at any point of the execution. Let X be the total number of ports connecting bridge edges over which messages are sent during the run of the algorithm and let X_u be the indicator random variable that is 1 iff node u sends a message across its bridge edge. Let S_u be the number of messages sent by node u. It follows by the hypergeometric distribution that

$$\mathrm{E}[X_u \mid G, M] = \frac{S_u}{\Theta(n)},$$

for each node u and hence,

$$\mathrm{E}[X \mid G, M] = \textstyle\sum_{u \in V(G)} \frac{S_u}{\Theta(n)} = \frac{1}{\Theta(n)} \sum_{u \in V(G)} S_u = o(1),$$

where we have used the fact that $\sum_{u \in V(G)} S_u = o(n)$ due to conditioning on event M. By Markov's Inequality, it follows that the event $C_1 \leftrightarrow C_2$, i.e., $X \geq 1$, occurs with probability at most $o(1)$.

We now combine the above observations to obtain

$$\begin{aligned}
&\Pr[\text{One} \mid G, M] && (5)\\
&= \Pr[\text{One} \mid C_1 \nleftrightarrow C_2, G, M] \Pr[C_1 \nleftrightarrow C_2 \mid G, M]\\
&+ \Pr[\text{One} \mid C_1 \leftrightarrow C_2, G, M] \Pr[C_1 \leftrightarrow C_2 \mid G, M]\\
&\leq \Pr[\text{One} \mid C_1 \nleftrightarrow C_2, G, M] + o(1) && \text{(by Lemma 8)}\\
&\leq \frac{1}{2} + o(1), && (6)
\end{aligned}$$

where the last inequality follows by first using (4) and noting that the upper bound (3) still holds when conditioning on the event M.

Finally, we recall that the algorithm succeeds with probability at least $\frac{1}{2} + \varepsilon$ and $\Pr[M \mid G] \geq 1 - o(1)$, which yields

$$\frac{1}{2} + \varepsilon \leq \Pr[\text{One} \mid G] \leq \Pr[\text{One} \mid G, M] + o(1) \leq \frac{1}{2} + o(1),$$

which is a contradiction, since we have assumed that $\varepsilon > 0$ is a constant.

This completes the proof of Theorem 3.

4 A Deterministic Algorithm

Our algorithm (Algorithm 2) is inspired by the solution of Afek and Gafni [1] for the n-node clique. However, there are some complications that we explain below, since we cannot rely on every pair of nodes to be connected by an edge. Note that our algorithm assumes that n (or a constant factor upper bound for $\log n$) is known to all nodes.

For any node $v \in V$, we denote the degree of v by d_v and the ID of v by ID_v. At any time-point in the algorithm, L_v denotes the highest ID that v has so far learned (among all the probe messages it has received, in the current round or in some previous round).

The algorithm proceeds as a sequence of $\lceil \log n \rceil$ phases. Initially every node is a "candidate" and is "active". Each node v numbers its neighbors from 1 to d_v, denoted by

$$w_{v,1}, w_{v,2}, \ldots, w_{v,d_v}$$

respectively. In phase i, if a node v is active, v sends probe-messages containing its ID to its neighbors $w_{v,2^{i-1}}, \ldots, w_{v,k}$, where $k = \min\left\{d_v, 2^i - 1\right\}$. [8] Each one of them replies back with the highest ID it has seen so far. If any of those ID's is higher than ID_v, then v stops being a candidate and becomes inactive. Node v also becomes inactive if it has finished sending probe-messages to all its neighbors. After finishing the $\lceil \log n \rceil$ phases v becomes the leader if it is still a candidate.

The idea behind the algorithm is to exploit the *neighborhood intersection property* (cf. Observation 1) of diameter-two networks. Since for any $u, v \in V$, there is an $x \in V$ that is connected to both u and v (unless u and v are directly connected via an edge) and acts as a "referee" node for candidates u and v. This means that x serves to inform u and v who among them is the winner, i.e., has the higher ID. Thus at the end of the algorithm, every node except the one with the highest ID should know that he is not a leader. We present the formal analysis of Theorem 4 in Sections 4.1 and 4.2.

Theorem 4 *There exists a deterministic leader election algorithm for n-node networks with diameter at most two that sends $O(n \log n)$ messages and terminates in $O(\log n)$ rounds.*

In the pseudocode and the subsequent analysis we use v and ID_v interchangeably to denote the node v.

4.1 Proof of Correctness

Define $v^{\max}$ to be the node with the highest ID in G.

Lemma 9 *v^{max} becomes a leader.*

Proof Since $v^{\max}$ has the highest ID in G, the if-clause of Line 15 of Algorithm 2 is never satisfied for $v^{\max}$. Therefore $v^{\max}$ never becomes a non-candidate, and hence becomes a leader at the end of the algorithm.

Lemma 10 *No other node except v^{max} becomes a leader.*

Proof Consider any $u \in V$ such that $u \neq v^{\max}$.

- **Case** 1 (**$v^{\max}$ and u are connected via an edge):** Since $v^{\max}$ has the highest ID in G, the if-clause of Line 15 of Algorithm 2 is never satisfied for $v^{\max}$. Therefore $v^{\max}$ becomes inactive only if it has already sent probe-messages to all its neighbors (or $v^{\max}$ never becomes inactive). In particular, u always receives a probe-message from $v^{\max}$ containing $ID_{v^{\max}}$. Since $ID_{v^{\max}} > ID_u$, u becomes a non-candidate at that point (if u was still a candidate until that point) and therefore does not become a leader.

[8] We note that this is the main idea borrowed from the Afek-Gafni algorithm [1] — the number of messages sent by each "active" node *increases* exponentially in every phase. That effect is, however, counterbalanced by the exponentially *decreasing* number of "active" nodes.

Algorithm 2 Deterministic Leader Election in $O(\log n)$ rounds and with $O(n \log n)$ messages: Code for a node v

```
 1: $v$ becomes a “candidate” and “active”.
 2: $L_v \leftarrow ID_v$.
 3: $N_v \leftarrow ID_v$.
 4: $v$ numbers its neighbors from 1 to $d_v$, which are called $w_{v,1}, w_{v,2}, \ldots, w_{v,d_v}$ respectively.
 5: for phase $i = 1$ to $\lceil \log n \rceil$ do        ▷ This uniformity hypothesis is needed only for termination.
 6:     if $v$ is active then
 7:         $v$ sends a “probe" message containing its ID to its neighbors $w_{v,2^{i-1}}, \ldots, w_{v,\min\{d_v,2^i-1\}}$.
 8:         if $d_v \le 2^i - 1$ then    ▷ If $v$ is finished with exploring its adjacency list, $v$ becomes inactive.
 9:             $v$ becomes inactive.
10:         end if
11:     end if
12:     Let $\mathcal{X}_v$ be the set (possibly empty) of neighbors of $v$ from whom $v$ receives messages in this
    round.
13:     Let $\mathrm{ID}(\mathcal{X}_v)$ be the set of ID’s sent to $v$ by the members of $\mathcal{X}_v$.
14:     Let $ID_u$ be the highest ID in $\mathrm{ID}(\mathcal{X}_v)$.
15:     if $ID_u > L_v$ then
16:         $L_v \leftarrow ID_u$.                        ▷ $v$ stores the highest ID seen so far in $L_v$.
17:         $v$ tells $N_v$ about $L_v = ID_u$, i.e., the highest ID it has seen so far.
18:         $N_v \leftarrow x$.                       ▷ $v$ remembers neighbor who told $v$ about $L_v$.
19:         $v$ becomes “inactive” and “non-candidate”.
20:     end if
21:     $v$ tells every member of $\mathcal{X}_v$ about $L_v$, i.e., the highest ID it has seen so far.
22: end for
23: if $v$ is still a candidate then
24:     $v$ elects itself to be the leader.
25: end if
```

- **Case** 2 (**$v^{\max}$ and u do not have an edge between them):** By Observation 1, there is some $x \in V$ such that both $v^{\max}$ and u have edges going to x. Then it is clear that x will cause u to become a non-candidate at some point of time or another. We can do a more rigorous case-by-case analysis, but we defer that mundane technicality to the appendix (Section A.3) for the sake of maintaining fluency of the presentation here in the main paper.

Lemmas 9 and 10 together imply that Algorithm 2 elects a unique leader (the node with the highest ID) and is therefore *correct*.

4.2 Message Complexity

Lemma 11 *At the end of round i, there are at most $\frac{n}{2^i}$ “active” nodes.*

Proof Consider a node v that is active at the end of round i. This implies that the if-clause of Line 15 of Algorithm 2 has not so far been satisfied for v, which in turn implies that $ID_v > ID_{w_{v,j}}$ for $1 \le j \le 2^i - 1$, therefore none of

$$w_{v,1}, w_{v,2}, \ldots, w_{v,2^i-1}$$

is active after round i. Thus for every active node at the end of round i, there are at least $2^i - 1$ inactive nodes. We call this set of inactive nodes, together with v itself, the “kingdom” of v after round i, i.e.,

$$KINGDOM_i(v) \overset{\text{def}}{=} \{v\} \cup \left\{w_{v,1}, w_{v,2}, \ldots, w_{v,2^i-1}\right\}$$

$$\text{and } |KINGDOM_i(v)| = 2^i.$$

If we can show that these kingdoms are disjoint for two different active nodes, then we are done.

Proof by contradiction. Suppose not. Suppose there are two active nodes u and v such that

$$u \neq v \text{ and } KINGDOM_i(u) \cap KINGDOM_i(v) \neq \phi$$

(after some round i, $1 \leq i \leq \log n$). Let x be such that $x \in KINGDOM_i(u) \cap KINGDOM_i(v)$. Since an active node obviously cannot belong to the kingdom of another active node, this x equals neither u nor v, and therefore

$$x \in \left\{w_{v,1}, w_{v,2}, \ldots, w_{v,2^i-1}\right\} \cap \left\{w_{u,1}, w_{u,2}, \ldots, w_{u,2^i-1}\right\},$$

that is, both u and v have sent their respective probe-messages to x. Then it is straightforward to see that x would not allow u and v to be active at the same time. We can do a more rigorous case-by-case analysis, but we defer that mundane technicality to the appendix (Section A.3) for the sake of maintaining fluency of the presentation here in the main paper.

Lemma 12 *In round i, at most* $3n$ *messages are transmitted.*

Proof In round i, each active node sends exactly 2^{i-1} probe messages, and each probe-message generates at most two responses (corresponding to Lines 17 and 21 of Algorithm 2). Thus, in round i, each active node contributes to, directly or indirectly, at most 3.2^{i-1} messages. The result immediately follows from Lemma 11.

Since the algorithm runs for $\log n$ rounds, the total number of messages transmitted is at most $3n\log n$, and Theorem 4 immediately follows.

5 A Deterministic Lower Bound

In this section we show that $\Omega(n\log n)$ is a lower bound on the message complexity for solving leader election with any deterministic algorithm in diameter-two networks. Notice that [1] shows a lower bound of $\Omega(n\log n)$ for the special case of complete graphs (i.e., diameter-one networks), and we know from [11] that the message complexity becomes $\Omega(m)$ for (most) diameter-three networks. Thus, Theorem 5 completes the picture regarding the message complexity of leader election when considering networks according to their diameter.

More Formally: For every deterministic algorithm that solves implicit leader election in all diameter-two networks, there exists some graph of diameter two where the said algorithm sends at least $\Omega(n\log n)$ messages. As usual, n is the number of nodes in the network.

In other words, we show that

Theorem 5 *There is no deterministic algorithm that solves leader election in every diameter-two network of n nodes with* $o(n\log n)$ *messages.*

Remark 4 In [1], the $\Omega(n\log n)$ message lower bound was showed for complete networks under the non-simultaneous wakeup model in synchronous networks. The same message bound can be shown to hold in the simultaneous wake-up model as well under the restriction that the number of rounds is bounded by a function of n [10]. We will show a lower bound of $\Omega(n\log n)$ message complexity by reducing the problem of "leader election in complete graphs" to that of "leader election in graphs of diameter two". This reduction itself would take two rounds and $O(n)$ messages. Then, since the former is known to have $\Omega(n\log n)$ message complexity under the aforementioned constraints, the latter would have the same lower bound too (cf. Section 1.1).

In the remainder of this section, we prove Theorem 5.

Proof by reduction. Suppose $\mathscr{A}$ is a leader election algorithm that works for any graph of diameter two. Let $G = (V,E)$ be our input instance for the problem of "leader election in complete graphs", i.e., G is the complete graph on n nodes, say.

The Reduction. G sparsifies itself into a diameter-two graph (G', say, where $G' = (V,E')$, where $E' \subsetneq E$) on which $\mathscr{A}$ works thereafter. This sparsification takes $O(n)$ messages and a constant number of rounds (two, to be exact) and is done as follows.

- **Round** 1**:** Each node v chooses one of its neighbours w (any arbitrary one), say, and asks its ID. If $ID_w > ID_v$, then both v and w agree to "drop" that edge, i..e., it won't use that for communication in the subsequent simulation of $\mathscr{A}$. Otherwise v and w keep that edge.
  For $v \in V$, if v has $\lceil \frac{n}{2} \rceil$ or more edges removed, then v makes itself a "candidate".
- **Round** 2**:** The candidates from the previous round send their ID's to all the nodes in the network using edges of G. By Lemma 13, there can be at most two such nodes. Thus the total number of messages sent is still $O(n)$. Then each node (including the candidates themselves) receives the ID's of up to two candidates and chooses the highest of them to be the ID of the leader.

If no such node exists which has had $\lceil \frac{n}{2} \rceil$ or more edges removed, then G' has diameter two (please refer to Lemma 14), and we run $\mathscr{A}$ on G'. $\mathscr{A}$ returns a leader on G' which makes itself the leader of G too, and informs all its neighbors. This takes $O(n)$ messages.

5.1 Proof of Correctness

Observation 4 E has at most $(n-1)$-edges more than E'.

Proof Each node except the node with the highest ID drops at most one edge. The node with the highest ID drops no edge.

Lemma 13 *For $n \geq 3$, there can be at most two nodes in G' that has had $\lceil \frac{n}{2} \rceil$ or more edges removed.*

Proof A simple counting argument tells us that Lemma 13 follows from Observation 4. We need to be careful with the ceiling operators and the floor operators here though. We can do a more rigorous case-by-case analysis (for the two cases when n is *even* and when n is *odd*, respectively), but we defer that mundane technicality to the appendix (Section A.4) for the sake of maintaining fluency of the presentation here in the main paper.

Lemma 14 *If no node exists in G' which has had $\lceil \frac{n}{2} \rceil$ or more edges removed, then G' has diameter two.*

Proof Clearly G' is not of diameter one since the node with the smallest ID in V always drops at least one edge.

Now let us consider any pair of nodes u and v, say. We show that the distance between u and v cannot be greater than 2. This follows from this simple "pigeon hole principle" argument: If no node has had $\frac{n}{2}$ or more edges removed, then all nodes are of degree at least $\frac{n}{2}$; so for each pair of nodes there should be at least one common neighbour.

Of course, we need to be careful with the ceiling operators and the floor operators here. We can do a more rigorous case-by-case analysis (for the two cases when n is *even* and when n is *odd*, respectively), but we defer that mundane technicality to the appendix (Section A.5) for the sake of maintaining fluency of the presentation here in the main paper.

6 Conclusion

We settle the message complexity of leader election throughout the diameter spectrum, by presenting almost tight bounds (tight upto polylog(n) factors) for diameter-two graphs which were left open by previous results [12, 11]. Several open problems arise from our work.

1. Is it possible to show a high probability bound of $O(n)$ messages for randomized, implicit leader election that runs in $O(1)$ rounds? This will match the lower bounds, by closing the polylog(n) factor. It might be possible to improve the analysis of our randomized, implicit algorithm to show $O(n \log n)$ messages.
2. Coming to deterministic algorithms, another very interesting question is whether *explicit* leader election can be performed in $\tilde{O}(n)$ messages in diameter-two graphs *deterministically*.
3. The question of explicit leader election naturally begs the question whether *broadcast*, another fundamental problem in distributed computing, can be solved *deterministically* in *diameter-two* graphs with $\tilde{O}(n)$ messages and $O(\text{polylog}(n))$ rounds if n is known. [9]
4. Removing the assumption of the knowledge of n (or showing that it is not possible) for deterministic, implicit leader election algorithms with $\tilde{O}(n)$ message complexity and running in $\tilde{O}(1)$ rounds is open as well.

References

1. Afek, Y., Gafni, E.: Time and message bounds for election in synchronous and asynchronous complete networks. SIAM Journal of Computing **20**(2), 376–394 (1991). DOI 10.1137/0220023. URL http://dx.doi.org/10.1137/0220023
2. Gilbert, S., Robinson, P., Sourav, S.: Leader election in well-connected graphs. In: Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC '18, pp. 227–236. ACM, New York, NY, USA (2018). DOI 10.1145/3212734.3212754. URL http://doi.acm.org/10.1145/3212734.3212754
3. Gilbert, S., Robinson, P., Sourav, S.: Slow links, fast links, and the cost of gossip. In: 2018 IEEE 38th International Conference on Distributed Computing Systems, ICDCS '18, pp. 786–796 (2018). DOI 10.1109/ICDCS.2018.00081
4. Humblet, P.A.: Electing a leader in a clique in $O(n \log n)$ messages (1984). Intern. Memo., Laboratory for Information and Decision Systems, MIT, Cambridge, Massachusetts
5. Khan, M., Kuhn, F., Malkhi, D., Pandurangan, G., Talwar, K.: Efficient distributed approximation algorithms via probabilistic tree embeddings. Distributed Computing **25**(3), 189–205 (2012). DOI 10.1007/s00446-012-0157-9. URL https://doi.org/10.1007/s00446-012-0157-9
6. Korach, E., Kutten, S., Moran, S.: A modular technique for the design of efficient distributed leader finding algorithms. ACM Transactions on Programming Languages and Systems (TOPLAS) **12**(1), 84–101 (1990). DOI 10.1145/77606.77610. URL http://doi.acm.org/10.1145/77606.77610
7. Korach, E., Moran, S., Zaks, S.: Tight lower and upper bounds for some distributed algorithms for a complete network of processors. In: Proceedings of the Third Annual ACM Symposium on Principles of Distributed Computing, PODC '84, pp. 199–207. ACM, New York, NY, USA (1984). DOI 10.1145/800222.806747. URL http://doi.acm.org/10.1145/800222.806747
8. Korach, E., Moran, S., Zaks, S.: The optimality of distributive constructions of minimum weight and degree restricted spanning trees in a complete network of processors. SIAM Journal on Computing **16**(2), 231–236

[9] In contrast, we note that $\Omega(m)$ is a lower bound for deterministic on graphs of *diameter at least three*, even if n is known [11].

(1987). DOI 10.1137/0216019. URL https://doi.org/10.1137/0216019

9. Korach, E., Moran, S., Zaks, S.: Optimal lower bounds for some distributed algorithms for a complete network of processors. Theoretical Computer Science **64**(1), 125 – 132 (1989). DOI http://dx.doi.org/10.1016/0304-3975(89)90103-5. URL http://www.sciencedirect.com/science/article/pii/0304397589901035
10. Kutten, S.: Private communication (2017)
11. Kutten, S., Pandurangan, G., Peleg, D., Robinson, P., Trehan, A.: On the complexity of universal leader election. Journal of the ACM **62**(1), 7:1–7:27 (2015). DOI 10.1145/2699440. URL http://doi.acm.org/10.1145/2699440. Invited paper from *ACM PODC* 2013
12. Kutten, S., Pandurangan, G., Peleg, D., Robinson, P., Trehan, A.: Sublinear bounds for randomized leader election. Theoretical Computer Science **561, Part B**, 134 – 143 (2015). DOI https://doi.org/10.1016/j.tcs.2014.02.009. URL http://www.sciencedirect.com/science/article/pii/S0304397514001029. Special Issue on Distributed Computing and Networking
13. Le Lann, G.: Distributed Systems — Towards a formal approach. In: IFIP Congress, pp. 155–160 (1977)
14. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1996)
15. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomized Algorithms and Probabilistic Analysis, 2$^{\text{nd}}$ edn. Cambridge University Press, Cambridge CB2 8BS, United Kingdom (2017). URL http://www.cambridge.org/9781107154889
16. Pai, S., Pandurangan, G., V. Pemmaraju, S., Riaz, T., Robinson, P.: Symmetry breaking in the congest model: Time- and message-efficient algorithms for ruling sets. In: 31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria, pp. 38:1–38:16 (2017). DOI 10.4230/LIPIcs.DISC.2017.38. URL https://doi.org/10.4230/LIPIcs.DISC.2017.38
17. Peleg, D.: Time-optimal leader election in general networks. Journal of Parallel and Distributed Computing **8**(1), 96–99 (1990). DOI 10.1016/0743-7315(90)90074-Y. URL http://dx.doi.org/10.1016/0743-7315(90)90074-Y
18. Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. Society for Industrial and Applied Mathematics (2000). DOI 10.1137/1.9780898719772. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898719772
19. Santoro, N.: Design and Analysis of Distributed Algorithms (Wiley Series on Parallel and Distributed Computing). Wiley-Interscience, New York, NY, USA (2006)
20. Tel, G.: Introduction to Distributed Algorithms, 2$^{\text{nd}}$ edn. Cambridge University Press, New York, NY, USA (2001)

A Appendix

A.1 Proof for function minima in Lemma 1

Proof We define the Lagrangian as

$$\mathscr{L} \overset{\text{def}}{=} f(x_1,x_2,\ldots,x_n) - \lambda(\sum_{i=1}^{n} x_i - C) \tag{7}$$

where λ is the Lagrange multiplier. We can find the *critical points* of the Lagrangian by solving the set of equations

$$\frac{\partial f}{\partial x_i} = \lambda \frac{\partial \sum_{i=1}^{n} x_i}{\partial x_i} \text{ for } i = 1,2,\ldots,n \tag{8}$$

and

$$\sum_{i=1}^{n} x_i = C \tag{9}$$

Simplifying Equation 8, we get

$$-\frac{\log x_i}{x_i^2} = \lambda \text{ for } i = 1,2,\ldots,n \tag{10}$$

One possible (feasible) solution of Equations 10 and 9 is,

$$x_i = \frac{C}{n} \text{ for all } 1 \leq i \leq n \tag{11}$$

and

$$\lambda^* = -\frac{\log(\frac{C}{n})}{(\frac{C}{n})^2} \tag{12}$$

Let X^* be a vector of dimension n defined by $X^* \overset{\text{def}}{=} (\frac{C}{n}, \frac{C}{n}, \ldots, \frac{C}{n})$. Then we have already shown that X^* and λ^* are a *critical point* for the Lagrange function $\mathscr{L}$. We claim that X^* is also a local minima for $f(x)$ under the constraint of Equation 9.

We show that by constructing the Bordered Hessian matrix H^B of the Lagrange function. Let $L_{ij}^* \overset{\text{def}}{=} \frac{\partial}{\partial x_j}(\frac{\partial \mathscr{L}}{\partial x_i})\Big|_{X^*}$, where $\mathscr{L}$ is the Lagrange function as defined in Equation 7. Then

$$H^B = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & L_{11}^* & L_{12}^* & \cdots & L_{1n}^* \\ 1 & L_{21}^* & L_{22}^* & \cdots & L_{2n}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & L_{n1}^* & L_{n2}^* & \cdots & L_{nn}^* \end{bmatrix}$$

We note that $L_{ii}^* = \frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^*$ for all $1 \leq i \leq n$, and $L_{ij}^* = 0$ for all (i,j) such that $i \neq j$. Hence

$$H^B = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & \frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^* & 0 & \cdots & 0 \\ 1 & 0 & \frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^* & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & \frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^* \end{bmatrix}$$

We show that H^B is *positive definite* (which is a sufficient condition for X^* to be a local minima) by checking the signs of the leading principal minors. For any $1 \leq i \leq n$, $|H_i^B|$ is the determinant of a square matrix of dimension $i+1$, and is given by

$$|H_i^B| = \begin{vmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & \frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^* & 0 & \cdots & 0 \\ 1 & 0 & \frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^* & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & \frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^* \end{vmatrix}$$

$$= -i(\tfrac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^*)^{i-1}.$$

$$\text{But } \frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} > 0 \qquad (\text{since } C \geq n\sqrt{2}, 2\log(\tfrac{C}{n})-1 > 0)$$

and $\lambda^* = -\frac{\log(\frac{C}{n})}{(\frac{C}{n})^2} < 0$.

Hence

$$\frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^* > 0$$

$$\implies -i(\frac{2\log(\frac{C}{n})-1}{(\frac{C}{n})^3} - \lambda^*)^{i-1} < 0$$

i.e., $|H_i^B| < 0$ for all $1 \leq i \leq n$

$\implies H^B$ is positive definite.

A.2 Detailed proof of Lemma 10

Proof Consider any $u \in V$ such that $u \neq v^{\max}$.

- **Case** 1 (**$v^{\max}$ and u are connected via an edge):** Since $v^{\max}$ has the highest ID in G, the if-clause of Line 15 of Algorithm 2 is never satisfied for $v^{\max}$. Therefore $v^{\max}$ becomes inactive only if it has already sent probe-messages to all its neighbors (or $v^{\max}$ never becomes inactive). In particular, u always receives a probe-message from $v^{\max}$ containing $ID_{v^{\max}}$. Since $ID_{v^{\max}} > ID_u$, u becomes a non-candidate at that point (if u was still a candidate until that point) and therefore does not become a leader.
- **Case** 2 (**$v^{\max}$ and u do not have an edge between them):** By Observation 1, there is some $x \in V$ such that both $v^{\max}$ and u have edges going to x. And we have already established that $v^{\max}$ will always send a probe-message to x at some point of time or another.
    - **Case** 2**(a)** (**u does not send a probe-message to x):** This implies that u became inactive before it could send a probe-message to x. But then u could have become inactive only if the if-clause of Line 15 of Algorithm 2 got satisfied at some point. Then u became a non-candidate too at the same time and therefore would not become a leader.
    - **Case** 2**(b)** (**u sends a probe-message to x before $v^{\max}$ does):** Suppose u sends a probe-message to x at round i and $v^{\max}$ sends a probe-message to x at round i', where $1 \leq i < i' \leq \log n$. If x had seen an ID higher than ID_u up until round i, then x immediately informs u and u becomes a non-candidate.

      So suppose not. Then, after round i, x sets its local variables L_x and N_x to ID_u and u respectively. Let $j > i$ be the smallest integer such that x receives a probe-message from a neighbor u' at round j, where $ID_{u'} > ID_u$. Note that $v^{\max}$ will always send a probe-message to x, therefore such a u' exists. Then, after round j, x sets its local variables L_x and N_x to $ID_{u'}$ and u' respectively, and informs u of this change (please see Line 17 of Algorithm 2). u becomes a non-candidate at that point of time.

- **Case** 2(**c**) (*u* **and** $v^{\max}$ **each sends a probe-message to** *x* **at the same time):** Since $ID_{v^{\max}}$ is the highest ID in the network, L_x is assigned the value $ID_{v^{\max}}$ at this point, and x tells u about $L_x = ID_{v^{\max}} > ID_u$, causing u to become a non-candidate.
- **Case** 2(**d**) (*u* **sends a probe-message to** *x* **after** $v^{\max}$ **does):** Suppose $v^{\max}$ sends a probe-message to x at round i and u sends a probe-message to x at round i', where $1 \le i < i' \le \log n$. Then x sets its local variables L_x and N_x to $ID_{v^{\max}}$ and $v^{\max}$, respectively, after round i. So when u comes probing at round $i' > i$, x tells u about $L_x = ID_{v^{\max}} > ID_u$, causing u to become a non-candidate.

A.3 Detailed proof of Lemma 11

Proof Consider a node v that is active at the end of round i. This implies that the if-clause of Line 15 of Algorithm 2 has not so far been satisfied for v, which in turn implies that $ID_v > ID_{w_{v,j}}$ for $1 \le j \le 2^i - 1$, therefore none of

$$w_{v,1}, w_{v,2}, \ldots, w_{v,2^i-1}$$

is active after round i. Thus for every active node at the end of round i, there are at least $2^i - 1$ inactive nodes. We call this set of inactive nodes, together with v itself, the "kingdom" of v after round i, i.e.,

$$KINGDOM_i(v) \overset{\text{def}}{=} \{v\} \cup \left\{w_{v,1}, w_{v,2}, \ldots, w_{v,2^i-1}\right\}$$

$$\text{and } |KINGDOM_i(v)| = 2^i.$$

If we can show that these kingdoms are disjoint for two different active nodes, then we are done.

Proof by contradiction. Suppose not. Suppose there are two active nodes u and v such that

$$u \neq v \text{ and } KINGDOM_i(u) \cap KINGDOM_i(v) \neq \phi$$

(after some round i, $1 \le i \le \log n$). Let x be such that $x \in KINGDOM_i(u) \cap KINGDOM_i(v)$. Since an active node obviously cannot belong to the kingdom of another active node, this x equals neither u nor v, and therefore

$$x \in \left\{w_{v,1}, w_{v,2}, \ldots, w_{v,2^i-1}\right\} \cap \left\{w_{u,1}, w_{u,2}, \ldots, w_{u,2^i-1}\right\},$$

that is, both u and v have sent their respective probe-messages to x. Without loss of generality, let $ID_v > ID_u$.

- **Case** 1 (*u* **sends a probe-message to** *x* **before** *v* **does):** Suppose u sends a probe-message to x at round j and v sends a probe-message to x at round j', where $1 \le j < j' \le i$. If x had seen an ID higher than ID_u up until round j, then x immediately informs u and u becomes inactive. Contradiction.
  So suppose not. Then, after round j, x sets its local variables L_x and N_x to ID_u and u respectively. Let $k > j$ be the smallest integer such that x receives a probe-message from a neighbor u' at round k, where $ID_{u'} > ID_u$. Note that v sends a probe-message to x at round j', where $j < j' \le i$, and $ID_v > ID_u$. Therefore such a u' exists. Then, after round k, x sets its local variables L_x and N_x to $ID_{u'}$ and u' respectively, and informs u of this change (please see Line 17 of Algorithm 2). u becomes inactive at that point of time, i.e., after round k, where $k \le j' \le i$. Contradiction.
- **Case** 2 (*u* **and** *v* **each sends a probe-message to** *x* **at the same time):** Suppose that u and v each sends a probe-message to x at the same round j, where $1 \le j \le i$. Since $ID_v > ID_u$, x has at least one neighbor u' such that $ID_{u'} > ID_u$. Therefore x would not set L_x to ID_u (or N_x to u), and x would inform u about that after round j, causing u to then become inactive. Contradiction.
- **Case** 3 (*u* **sends a probe-message to** *x* **after** *v* **does):** Suppose v sends a probe-message to x at round j and u sends a probe-message to x at round j', where $1 \le j < j' \le i$. Then x sets its local variables L_x and N_x to ID_v and v, respectively, after round j. So when u comes probing at round $j' > j$, x tells u about $L_x \ge ID_v > ID_u$, causing u to become inactive. Contradiction.

A.4 Detailed proof of Lemma 13

Proof We consider the two cases separately — when n is even and when n is odd — in order to make the presentation simpler.

- **Case** 1: $n = 2k$ **for some integer** $k \ge 2$.

  Proof by contradiction. Suppose that there are three or more nodes that have had $\lceil \frac{n}{2} \rceil = k$ or more edges removed each (either by themselves or by their neighbors). Let u, v, and w be three such nodes. Since an edge is removed only if one of the incident nodes has a higher ID than the other, all of (u,v), (v,w), and (w,u) cannot have been removed. Thus by a simple counting argument (by the "principle of inclusion-exclusion"), the total number of edges removed is at least $(3k-3)+1 = 3k-2 > 2k-1 = n-1$, which contradicts Observation 4.

- **Case** 2: $n = 2k+1$ **for some integer** $k \ge 1$.

  Proof by contradiction Suppose that there are three or more nodes that have had $\lceil \frac{n}{2} \rceil = k+1$ or more edges removed each (either by themselves or by their neighbors). Let u, v, and w be three such nodes. Since an edge is removed only if one of the incident nodes has a higher ID than the other, all of (u,v), (v,w), and (w,u) cannot have been removed. Thus by a simple counting argument (by the "principle of inclusion-exclusion"), the total number of edges removed is at least $(3(k+1)-3)+1 = 3k+1 > 2k = n-1$, which contradicts Observation 4.

A.5 Detailed proof of Lemma 14

Proof Clearly G' is not of diameter one since the node with the smallest ID in V always drops at least one edge. Now let us consider any pair of nodes u and v, say. We show that the distance between u and v cannot be greater than 2.

If $(u,v) \in E'$, then we are already done. So suppose $(u,v) \notin E'$. Next we show that for any $u, v \in V$, either u and v are directly connected in G' or $\exists w \in V$ such that $(u,w) \in E'$ and $(w,v) \in E'$.

We consider the two cases — when n is even and when n is odd — separately, in order to make the presentation simpler.

- **Case** 1: $n = 2k$ **for some integer** $k \ge 2$.
  Since no node exists in G' which has had $\lceil \frac{n}{2} \rceil = k$ or more edges removed, every node in G' has degree at least $(n-1)-(k-1) = k$. Thus for any $u, v \in V$, if $(u,v) \notin E'$, then there are at least $k + k - (n-2) = 2$ nodes in $V \setminus \{u,v\}$ that are common neighbors to both u and v.

- **Case** 2: $n = 2k+1$ **for some integer** $k \ge 1$.
  Since no node exists in G' which has had $\lceil \frac{n}{2} \rceil = k+1$ or more edges removed, that implies that every node in G' has degree at least $(n-1)-k = k$. Thus for any $u, v \in V$, if $(u,v) \notin E'$, then there is at least $k + k - (n-2) = 1$ node in $V \setminus \{u,v\}$, which is a common neighbor to both u and v.