Report

Maximal Independent Sets in Radio Networks

Author(s):
Moscibroda, Thomas; Wattenhofer, Roger

Publication Date:
2004

Permanent Link:
https://doi.org/10.3929/ethz-a-006776788 →

Maximal Independent Sets in Radio Networks

Thomas Moscibroda, Roger Wattenhofer
{moscitho,wattenhofer}@tik.ee.ethz.ch
Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland

Regular Presentation

Do not consider as a Brief Announcement

Student Paper

Keywords
Radio Networks, Maximal Independent Sets, Asynchronous Wake-Up
Ad Hoc Networks, Sensor Networks, Initialization

Pages
10 + 2 (References) + 4 (Appendix)

Abstract

We study the distributed complexity of computing a maximal independent set (MIS) in radio networks with unknown topology, asynchronous wake-up, no access to a global clock, and no collision detection mechanism available. Specifically, we propose a novel randomized algorithm that computes a MIS in time $O(\log^2 n)$ with high probability, significantly improving on the best previously known solutions. A lower bound of $\Omega(\log^2 n/\log\log n)$ given in [11] implies that our algorithm is close to optimal. Our result shows that the harsh radio network model imposes merely an additional $O(\log n)$ factor compared to Luby's MIS algorithm in the message passing model. This has important implications in the context of ad hoc and sensor networks whose characteristics are closely captured by the radio network model.

1 Introduction

In many ways, familiar distributed computing communication models such as the *message passing model* do not describe the harsh conditions faced in wireless ad hoc and sensor networks closely enough, rendering results obtained in these models bear little importance in practice. Ad hoc and sensor networks are multi-hop radio networks and hence, messages being transmitted may interfere with concurrent transmissions leading to collisions and packet losses. Furthermore, all nodes sharing the same wireless communication medium leads to an inherent broadcast nature of communication. A message sent by a node can be received by all nodes in its transmission range. These aspects of communication are modeled by the *radio network model* [3, 4].

Communication primitives such as broadcast, wake-up, or gossiping, have been extensively studied in the literature on radio networks in both randomized and deterministic versions (e.g., [3, 1, 5, 7, 9]). On the other hand, not much is known about the computation of *local network coordination structures* such as clusterings or colorings in the radio network model. This is surprising in view of the particular importance of such structures in the context of ad hoc and sensor networks.

In this paper, we study the problem of computing a *maximal independent set* (MIS) in the radio network model. In a graph $G = (V, E)$, a MIS is a subset $S \subseteq V$ of the nodes such that no two nodes $v, u \in S$ are neighbors and every node $w \notin S$ has at least one neighbor in S. The distributed complexity of computing a MIS in different computation models is of interest *per se*. However, the particular relevance of the MIS in the context of the multi-hop radio network model stems from the fact that the clustering induced by the MIS can be used as an *initial structure*. Ad hoc or sensor networks lack any fixed or built-in structure; it is the nodes themselves that have to provide a communication infrastructure in order to enable any reasonable communication. When being deployed, the quintessential difficulty in obtaining such an infrastructure is that a protocol designed for this task cannot rely on any previously established structure. The idea is to set up an *initial structure* based on which more sophisticated network organization procedures can subsequently be employed.[1]

Previously, the efficiency of algorithms for computing structures that can serve as an initial structure (dominating sets, MIS, coloring,...) has often been studied in message passing models and under the assumptions that all nodes wake up at the same time or that every node knows its neighbors.[2] By abstracting away these technicalities, the obtained solutions have often not been directly applicable in practical settings.

In this paper, we consider a radio network model described in [13] that is *strictly harder* than the standard radio network model and hence, our results are directly applicable to them. The model in [13] derives its motivation from the realm of *ad hoc and sensor networks* and makes the following assumptions.

- We consider *multi-hop* networks modelled as a unit disk graph $G = (V, E)$. Hence, it is possible that some neighbors of a sending node receive a message, whereas others experience a collision.
- Nodes wake up *asynchronously* without access to a global clock. In a wireless, multi-hop environment, it is realistic to assume that some nodes wake up (become deployed, or switched on, ...) later than others. In contrast to work on the so-called *wake-up problem* [10, 11, 6], nodes are *not* woken up by incoming messages. Sleeping nodes do neither send nor receive any messages.
- We consider radio networks without collision detection. That is, nodes are unable to distinguish background noise from interference noise. A sender does not know how many neighbors have received its transmission correctly.
- When waking up, nodes have no knowledge about other nodes' distribution or wake-up pattern. Particularly, a node has no a-priori information about the number of its neighbors and when waking up, it does not know whether some neighbors are al-

[1] Initializing ad hoc or sensor networks efficiently is a well-known problem in practice. Even in small-scale systems such as Bluetooth, the initialization tends to be slow.

[2] The usual justification for basing algorithms on such a rich set of assumptions is that these algorithms are running on top of an existing medium access layer (MAC) as well as an initial discovery protocol. The MAC protocol takes care of low-level aspects as interference and the discovery protocol deals with asynchronous wake-up, guaranteeing that every node knows its neighbors. Clearly, these assumptions highlight the "chicken-and-egg" problem of the initialization of ad hoc and sensor networks.

ready executing the algorithm.

All these restrictions suggest that we deal with a particularly harsh model of computation; one that closely captures the reality of many real ad hoc and sensor systems (particularly during and immediately after their deployment). What makes this model appealing to the *theory of distributed computing,* is the fact that it is not only realistic, but also concise enough to allow for stringent proofs, preventing simulations from being the last resort to evaluate an algorithm's performance.

In this paper, we show that even in our harsh model, a MIS can be computed efficiently. Specifically, we present a randomized algorithm that computes a MIS in time $O(\log^2 n)$ with high probability in unit disk graphs. This significantly improves the best previously known algorithm given in [19] that had a running time of $O(\log^9 n/\log\log n)$. Moreover, a $\Omega(\log^2 n/\log\log n)$ time lower bound established in [11] shows that our algorithm is asymptotically near optimal.

The paper is organized as follows. An overview of related work is given in Section 2. Section 3 formally introduces the model of computation. The algorithm is then presented and analyzed in Sections 4 and 5. Finally, Section 6 concludes the paper.

2 Related Work

The distributed (randomized and deterministic) complexity of computing a MIS has been of fundamental interest to the distributed computing community for various reasons[17, 8, 2, 16]. A major breakthrough in the understanding of the distributed computational complexity of MIS was the ingenious randomized algorithm by Luby [17] that has a running time of $O(\log n)$. On the other hand, [14] showed that every distributed (possibly randomized) algorithm requires at least time $\Omega(\sqrt{\log n/\log\log n})$ and $\Omega(\log\Delta/\log\log\Delta)$ in order to obtain a MIS, Δ being the largest degree in the network. Unfortunately, Luby's algorithm cannot be directly transformed to the radio network model since it assumes synchronous wake-up, knowledge about the neighborhood, and collision-free communication.

Radio networks have been extensive studied for over two decades, resulting in a vast and impressively rich literature. For a thorough survey of the single-hop case, we refer to [4]. In the multi-hop case, most work deals with the broadcast problem [3, 1, 15, 5, 9], as well as related problems such as gossiping [5] or the wake-up problem [10, 11, 6].

The harder version of the radio network used in this paper was introduced in [13] and subsequently refined in [12]. Both papers consider the problem clustering in ad hoc and sensor networks. In [12], the authors give an algorithm that computes a constant approximation for the minimum dominating set problem and runs in time $O(\log^6 n)$. A MIS is a constant approximation to the minimum dominating set problem [18] and therefore our $O(\log^2 n)$ algorithm significantly outperforms the solution presented in [12].

The problem of computing a MIS in the radio network model was first studied in [19]. In [19], the authors propose a randomized algorithm for computing a MIS in time $O(\log^9 n/\log\log n)$ with high probability. The algorithm in [19] was presented using *three independent communication* channels. That is, messages sent on different channels do neither interfere nor cause any collisions. Having independent communication channels allows for simpler algorithmic solutions and particularly, facilitates the analysis because events on different channels can be analyzed independently. The actual single-channel algorithm was then derived by using a generic procedure for simulating a multi-channel system in a single-channel system. In contrast, we design and analyze our algorithm in a *single-communication channel setting* directly. While complicating the analysis, this novel technical approach allows us to achieve the drastic improvements.

One characteristic feature of our radio network model is that nodes can wake up asynchronously at any time. There is a body of work dealing with a different form of asynchronous wake-up in radio networks. In the *wake-up problem* the time when each node joins the protocol is controlled by an adversary, and the goal of the algorithm is to perform a broadcast as quickly as possible [10, 11, 7, 6]. However, notice that in contrast to our model, sleeping nodes are *woken up* upon receiving a neighbors' message. This notion of asynchronity leads to interesting algorithmic challenges and a bunch of beautiful results. In the context of this paper, the $\Omega(\log^2 n/\log\log n)$ time-lower bound in [11] for the time required until one node can send without collision in a single-hop radio network is of particular interest because it is a lower bound for our MIS problem. In general, however, we do not believe that ad hoc and sensor nodes are woken up by incoming messages. Instead, a node simply wakes up when being deployed or switched on.

3 Model and Notation

We study the radio network model as described in [13]. The wireless nature of the network is modeled as a unit disk graph (UDG) $G = (V, E)$. In a UDG, nodes are considered to be located in the Euclidean plane and there is an edge (communication link) between two nodes u and v iff the Euclidean distance between u and v is at most 1. A node receives a message only if *exactly one* of its neighbors send. Moreover, having *no collision detection*, nodes are unable to distinguish between the situation in which two or more neighbors are sending and the situation in which no neighbor is sending. Every node $v \in V$ is assumed to have a unique identifier (not necessarily in the range $[1, \ldots, n]$). The size of a message is restricted to $O(\log n)$ bits under the assumption that the nodes' ID space is polynomial in n.

Nodes wake up *asynchronously* at any time; formally, every node $v \in V$ has a wake-up time t_v. Before waking up, a node does neither send nor receive any messages and in particular, nodes are *not woken up* by incoming messages. At time t_v, node v does not know a) how many neighbors it has and b) whether (and if yes, when) some of its neighbors have woken up and started executing the algorithm. We assume that both time and location of a node's wake-up is determined by an adversary, i.e., we consider worst case distributions in time and space.

As motivated in the introduction, asynchronous wake-up and lack of knowledge are two important features of the unstructured radio network model. The only a-priori knowledge given to the nodes is an upper bound for the total number of nodes $|V| = n$ in the network. It has been shown in [11] that without any such estimate of n, every algorithm requires at least time $\Omega(n/\log n)$ until one single message can be transmitted without collision. In practice, the number of nodes in a network may not be known exactly, but it can roughly be estimated in advance.

For the sake of the analysis, we assume time to be divided into time-slots that are synchronized among all nodes. However, our algorithm does not rely on synchronization in any way. By the standard argument introduced in [21] for slotted vs. unslotted ALOHA, the realistic unslotted case and the idealized slotted case differ only by a constant factor. A node v's running time T_v is defined as the number of time-slots between its wake-up and its (irrevocable) decision of whether or not to become a MIS node. The algorithm's running time R is $R = \max_{v \in V} T_v$.

We denote by N_v the set of neighbors of node v and define $N_v^+ = N_v \cup \{v\}$. In each time-slot, a node can either send or not send. A node v receives a message in a time-slot t only if exactly one node in N_v has sent a message in t. We write $p_v(t)$ to denote node v's sending probability in time-slot t. If it is clear from the context which time-slot is considered, we will also use the short form p_v. Finally, we say that node v sends *successfully* in time-slot t if all its neighbors receive v's message. During the execution of the algorithm, we call node v *covered* if there is a node $w \in N_v^+$ that has decided to become a MIS node. Note that a covered node may not *know* of its being covered.

We conclude this section with two facts. The first was proven in [11] and the second can be found in standard mathematical textbooks.

Fact 1. *Given a set of probabilities* $p_1 \ldots p_n$ *with* $\forall i : p_i \in [0, \frac{1}{2}]$, *the following inequalities hold:* $(1/4)^{\sum_{k=1}^n p_k} \leq \prod_{k=1}^n (1 - p_k) \leq (1/e)^{\sum_{k=1}^n p_k}$.

Fact 2. *For all* n, t, *with* $n \geq 1$ *and* $|t| \leq n$, $e^t \left(1 - t^2/n\right) \leq (1 + t/n)^n \leq e^t$.

In this paper, the term "with high probability" is reserved for an event which occurs with probability $1 - n^{-c}$ for a constant c which can be made arbitrarily large by setting the corresponding algorithm constants to large enough values.

4 Algorithm

In this section, we present Algorithm 1. Every time-slot corresponds to one iteration of the main loop. The *Receive Triggers* are executed immediately after the receipt of a message, regardless of the current state of the algorithm. In accordance to the model given in the previous section, however, a node receives a message only if it does not send a message itself in the same time-slot. Throughout the paper, Greek letters represent constants.

At any time during the execution of Algorithm 1, a node can be in one of five states. Upon waking up, a node is in the *waiting state* $\mathcal{W}$ in it only listens. If a node does not become covered by a MIS node in this state already, it will become *active*. Active nodes are in state $\mathcal{A}$. An active node v tries to join the MIS by increasing its probability p_v of becoming a *candidate*. Eventually, some active nodes will become candidates by entering state $\mathcal{C}$, whereas others will restart the algorithm, returning to the initial waiting state $\mathcal{W}$. Finally, MIS nodes are elected from among neighboring candidates. Nodes that have decided to be a MIS node end up in state $\mathcal{M}$, nodes that are covered become *slaves* and enter state $\mathcal{S}$. Throughout the paper, we will use the expression $\mathcal{W}$ to denote both the *state* in which the algorithm is currently in, as well as the subset of nodes $v \in V$ that are currently in the state $\mathcal{W}$. The same holds for all other states/sets. Next, we describe each of the five states in more detail. In the waiting state $\mathcal{W}$, a node listens for messages and increases the counter $step$ in each time-slot. The purpose of state $\mathcal{W}$ is that newly awakening or restarting nodes should not interfere with nodes that are actively competing for becoming a MIS node.

Once the $step$ counter of a node $v \in \mathcal{W}$ reaches the threshold $4\mu\delta\log^2 n$ (Line 3), it proceeds to the *active state* $\mathcal{A}$. Every active node has a sending probability p_v which is the probability that it sends a message m_A and becomes a candidate in a given time-slot (Lines 9-11). Starting from a small initial probability p_v, a node $v \in \mathcal{A}$ doubles p_v every $\lambda \log n$ time-slots, thereby exponentially increasing its chance to become a candidate (Lines 6 and 7). If however, an active node $v \in \mathcal{A}$ receives a message m_A from another active node, it returns to the start of the algorithm, i.e., it sets its state to $\mathcal{W}$ and resets $step$ to 0 (Receive Trigger 1). Such nodes may again try to become a candidate subsequently. State $\mathcal{A}$ is designed to bound the number of candidates simultaneously being in state $\mathcal{C}$ in a certain area of the graph. This enables a quick election of MIS nodes among the limited number of candidates. In other words, state $\mathcal{A}$ allows for a first rough selection on the way towards picking MIS nodes.

Having bounded the number of *candidates*, it remains to select MIS nodes from the candidates. The idea is that neighboring candidates compete with each other such that no two neighboring nodes join state $\mathcal{M}$. They do so by means of a $count$ variable. Intuitively, the current value of the $count$ describes a node's progress towards joining the MIS. In each time-slot, a candidate sends a message $m_C(count)$ contain-

Algorithm 1 MIS-Algorithm (Code of node v)

```
$step := count := 0$;  $state := \mathcal{W}$;  $p_v := \frac{2^{-\alpha-1}}{n}$;
upon wake-up do:
 1: loop
 2:   case $state$ do
 3:   $\mathcal{W}$ : if $step \geq 4\mu\delta\log^2 n$ then
 4:           $state := \mathcal{A}$;  $step := 0$;
 5:       end if
 6:   $\mathcal{A}$ : if $step \geq \lambda \log n$ then
 7:           $p_v := 2p_v$;  $step := 0$;
 8:       else
 9:           $s := \begin{cases} 1 & \text{with probability } p_v \\ 0 & \text{with probability } 1 - p_v \end{cases}$
10:           if $s = 1$ then
11:             send($m_A$);  $state := \mathcal{C}$;
12:           end if
13:       end if
14:   $\mathcal{C}$  : $count := count + 1$;
15:         $s := \begin{cases} 1 & \text{with prob. } q_C = \frac{\tau}{2^\alpha \log n} \\ 0 & \text{with prob. } 1 - q_C \end{cases}$
16:         if $count \geq \delta\log^2 n$ then
17:           $state := \mathcal{M}$       {$*$ $v$ joins MIS $*$}
18:         else if $s = 1$ then
19:           $count := \max\{count, \delta\log n + 1\}$;
20:           send $m_C(count)$;
21:         end if
22:   $\mathcal{M}$ : send $m_M$ with probability $q_M = 2^{-\alpha}$
23:   end case
24:   $step := step + 1$;
25: end loop
```

Receive Triggers (only when not sending):

```
1: upon receiving $m_A$ do:
     if $state = \mathcal{W}$ or $state = \mathcal{A}$ then
       $state := \mathcal{W}$;  $step := 0$
     end if
2: upon receiving $m_C(count')$ do:
     $\Delta c := |count' - count|$;
     if $state = \mathcal{C}$ and $\Delta c \leq \delta \log n$ then
       $count := 0$;
     end if
3: upon receiving $m_M$ do:
     $state := \mathcal{S}$;  stop();    {$*$ $v$ becomes slave $*$}
```

ing its current $count$ value with probability q. Candidates increase their $count$ value in every time-slot. When receiving a message $m_C(count')$ from another candidate, the receiver compares the sender's $count'$ to its own. If the two values are within $\delta \log n$ of each other, the receiver resets its own $count$ (Receive Trigger 2). The idea is that a candidate v resets its $count$ if the progress of v and w are too close to one another. As we will show in Section 5, this method of comparing counters prevents two neighboring nodes from joining the MIS shortly in succession and consequently ensures the correctness of the resulting MIS. On the other hand, it also allows the nodes to make fast progress in all parts of the network graph. Specifically, it avoids long cascading chains of resets that slow down the algorithm's running time. Once a candidate's $count$ reaches the threshold $\delta \log^2 n$, it *becomes a MIS node* and enters its final state $\mathcal{M}$ (Line 17). MIS nodes continue to send messages m_M with a probability q_M in order to inform their neighbors that they are covered. Regardless of its current state, if a node receives a messages m_M during the algorithm (Receive Trigger 3), it decides to *become a slave*.

The constants μ and α are defined as $\mu = 19$ and $\alpha = 6.4$, respectively. The other constant parameters can be chosen to fine-tune the trade-off between running time and the probability of a correct execution. In particular, the larger λ and δ, the more probable it becomes that Algorithm 1 performs correctly. In order to obtain the high probability results in Section 5, we set $\lambda = 3 \cdot 2^{\alpha+2} 4^{\frac{9/4+3\mu}{2^\alpha}}$, $\delta = \frac{8 \cdot 2^\alpha}{\tau} \cdot 4^{\frac{6\mu}{2^\alpha}}$, and $\tau = 9000^{-1}$. Finally, note that due to asynchronous wake-up, different nodes may be in completely different states at the same time. To make things worse, a node has no a-priori knowledge in which states its (potential) neighbors are. Overcoming the absence of any such knowledge is one of the key challenges of this paper.

5 Analysis

In this section, we prove that Algorithm 1 computes a correct MIS in time $O(\log^2 n)$ with high probability. Following the approach in [19], we make use of an imaginary covering of the plane by disks D_i of radius $1/2$ as shown in Figure 1 in Appendix A. By placing these disks on a hexagonal lattice, the entire Euclidean plane is covered. By E_i^r, we denote the disk with radius r centered at the center of D_i. Observe that all nodes within a disk D_i can hear each other. On the other hand, a node outside $E_i^{1.5}$ cannot cause a collision at a node $v \in D_i$. We will use the following simple geometric facts which can be proven by standard area arguments.

Fact 3. *Disks $E_i^{1.5}$, and $E_i^{2.5}$ can be fully covered by μ and 2μ smaller disks D_i for $\mu = 19$, respectively. Also, the number of independent nodes in $E_i^{1.5}$, and $E_i^{2.5}$ is at most μ and 2μ, respectively.*

A main difficulty of the analysis is that nodes can unintentionally interfere with neighbors in different states. For instance, an active node v may cause a collision at a candidate node w in $\mathcal{C}$. This results in w not receiving a message m_C or m_M from a neighboring node, potentially causing a violation of the MIS condition. Or, a collision caused by a MIS message m_M may cause that a node in $\mathcal{A}$ does not receive a message m_A, which could lead to too many candidates. In view of the dependencies between all states, proving the impossibility (or the low probability) of the above examples is not trivial.[3]

Throughout the proof, we denote by $\mathcal{A}_i$ and $\mathcal{A}_i^r$ the set of active nodes in D_i and E_i^r, respectively. $\mathcal{W}_i$, $\mathcal{C}_i$, and the other sets are defined analogously. We begin with a definition and a simple observation that follows directly from the definition of Algorithm 1.

Definition 5.1. *Let t be a time-slot in which a message m_A is sent by a node $v \in \mathcal{A}_i$ and received without collision by all nodes $w \in D_i \setminus \{v\}$. We call time-slot t a* clearance *of D_i.*

Lemma 5.2. *Consider a disk D_i. After a clearance, no node $v \in D_i$ is in state $\mathcal{A}$ for the next $4\mu\delta \log n$ time-slots. Consequently, two clearances in the same disk D_i must be at least $4\mu\delta \log n$ time-slots apart.*

A critical ingredient of the analysis is to bootstrap the argument. In this section, we will show that with high probability the algorithm maintains three properties (probabilistic invariants) throughout its execution. The first property states that the sum of sending probabilities by active nodes does not exceed a certain constant. This helps to bound the "noise" caused by such nodes when analyzing other aspects of the algorithm.

[3] Note that none of these difficulties did arise in the analysis of previous work on the computation of static structures in radio networks [13, 12, 19], because these papers assumed the availability of three independent communication channels.

Property 1 (P1). *For all disks D_i and at any time-slot t throughout the execution of the algorithm, $\sum_{v \in \mathcal{A}_i} p_v(t) \leq 2^{-\alpha}$.*

The second and third properties state that the number of simultaneous candidates is bounded and that $\mathcal{M}$ forms a correct independent set, respectively

Property 2 (P2). *For all disks D_i and at any time-slot t throughout the execution of Algorithm 1, it holds that $|\mathcal{C}_i| \leq \kappa \log n$, for a constant $\kappa = \tau^{-1}$.*

Property 3 (P3). *Throughout the execution of the algorithm, the set $\mathcal{M}$ forms a correct independent set.*

The first technical lemma shows that MIS nodes are capable of quickly informing their neighbors that they are covered. This is necessary to ensure the independence of the resulting set $\mathcal{M}$. For now, we can formalize this intuitive notion only under the assumption that all three Properties hold.

Lemma 5.3. *Assume Properties 1, 2, and 3 hold. With probability $1-n^{-3}$, every node $v \in V$ joins $\mathcal{S}$ and terminates the algorithm by the time $t_v + \delta \log n$, where t_v is the first time-slot in which v becomes covered by a MIS node $w \in \mathcal{M} \cap N_v^+$.*

Proof. See Appendix B for the proof. □

One difficulty when proving that the three Properties hold will be that our results depend on arguments of the following kind: *Before* a particular Property can be violated, there must exist some time-interval in which certain critical conditions hold. We can prove that in any interval exhibiting these conditions, the Property will *not* be violated with high probability. However, such a result is useless if there can be infinitely many time-intervals with these critical conditions. To bound the number of critical time-intervals, the next lemma lower bounds the *progress* made by the algorithm in case all three properties hold.

Lemma 5.4. *Assume Properties 1, 2, and 3 hold. Let t_c be a time-slot in which an uncovered node $v \in D_i$ is a candidate, i.e., $v \in \mathcal{C}$. In the interval $[t_c - \delta \log n, \ldots, t_c + 2\delta \log^2 n]$, a new MIS node emerges in $E_i^{2.5}$ with probability $1 - O(n^{-3})$.*

Proof. We first show that unless all candidates in D_i become covered by a MIS node (in which case the lemma clearly holds), there is at least one candidate that can send successfully in the interval $\mathcal{I}_1 = [t_c + 1, \ldots, t_c + \delta \log^2 n]$. Clearly, if a node $w \in N_v^+ \cap \mathcal{C}$ sends and no other node in $E_i^{2.5}$ sends, then w sends successfully. Hence, the probability P_1 that there is a candidate $w \in N_v^+ \cap \mathcal{C}$ sending successfully in a time-slot $t \in \mathcal{I}_1$ is at least

$$\begin{aligned}
P_1 \quad &\geq \quad \sum_{w \in N_v^+ \cap \mathcal{C}} \left(p_w \cdot \prod_{u \in E_i^{2.5} \setminus \{w\}} (1 - p_u) \right) \\
&\geq \quad \sum_{w \in N_v^+ \cap \mathcal{C}} p_w \cdot \prod_{u \in E_i^{2.5}} (1 - p_u) \\
&\underset{\text{Fact 1}}{\geq} \quad \sum_{w \in N_v^+ \cap \mathcal{C}} p_w \cdot (1/4)^{\sum_{u \in E_i^{2.5}} p_u} .
\end{aligned}$$

Similar to the proof of Lemma 5.3, we can bound the sum in the exponent as

$$\begin{aligned}
\sum_{u \in E_i^{2.5}} p_u = & \sum_{u \in \mathcal{A}_i^{2.5}} p_u + \sum_{u \in \mathcal{C}_i^{2.5}} p_u + \sum_{u \in \mathcal{M}_i^{2.5}} p_u \\
\leq & \sum_{D_j \in E_i^{2.5}} \left(\sum_{u \in \mathcal{A}_j} p_u + \sum_{u \in \mathcal{C}_j} p_u + \sum_{u \in \mathcal{M}_j} p_u \right) \\
\leq & \sum_{D_j \in E_i^{2.5}} \left(\frac{1}{2^\alpha} + \frac{\kappa \tau \log n}{2^\alpha \log n} + \frac{1}{2^\alpha} \right) \\
\leq & \; \frac{2\mu(2 + \kappa\tau)}{2^\alpha} .
\end{aligned}$$

Unless all candidates receive a message m_M and join the set $\mathcal{S}$ in $\mathcal{I}_1$, there is at least one candidate node in $N_v^+ \cap \mathcal{C}$ and therefore $\sum_{w \in N_v^+ \cap \mathcal{C}} p_w \geq q_C$. Plugging these results into the above expression for P_1 gives $P_1 \geq \frac{\tau}{(2^\alpha \log n)} \cdot (1/4)^{\frac{2\mu(2+\kappa\tau)}{2^\alpha}}$ and the probability P_{no} that none of the $\delta \log^2 n$ time-slots in $\mathcal{I}_1$ is successful is at most

$$\begin{aligned}
P_{no} \quad &\leq \quad \left(1 - \frac{\tau}{(2^\alpha \log n)} \cdot \left(\frac{1}{4} \right)^{\frac{2\mu(2+\kappa\tau)}{2^\alpha}} \right)^{\delta \log^2 n} \\
&\underset{\text{Fact 2}}{\leq} \quad e^{-\delta \log n \cdot \frac{\tau}{2^\alpha} (1/4)^{\frac{2\mu(2+\kappa\tau)}{2^\alpha}}} \in O(n^{-3}).
\end{aligned}$$

Therefore, with probability $1 - O(n^{-3})$, there exists a candidate w which manages to send successfully at a time-slot $t_w^* \in \mathcal{I}_1$. Since time-slot t_w^* was successful, all neighboring candidates of w received w's message $m_C(count_w)$. According to Receive Trigger 2, all nodes u in w's neighborhood that are candidates at time t^*_w and whose $count_u$ at time t^*_w (abbreviated by $c_u(t^*_w)$) is in the range $c_u(t^*_w) \in |c_w(t^*_w) - \delta \log n, \dots, c_w(t^*_w) + \delta \log n|$ will reset their *count* to 0. Moreover, by Line 19 of Algorithm 1, w has a *count* value of at least $\delta \log n + 1$. Therefore, for all existing candidates $u \in \mathcal{C} \cap N_w$ as well as for all potential new candidates (which set $count = 0$ when becoming candidate), it holds that after t^*_w,

$$|c_u(t^*_w + 1) - c_w(t^*_w + 1)| \ > \ \delta \log n.$$

By the definition of Algorithm 1 (Receive Trigger 2), this means that no node $u \in \mathcal{C}$ adjacent to w is able to prevent w from incessantly increasing its *count* until it eventually reaches the threshold $\delta \log^2 n$ that enables to join the MIS. The only possible way for w to be stopped from joining $\mathcal{M}$ is if it receives a message m_M from a node $x \in N_w$ that joined $\mathcal{M}$ *before* w. If x had already been a MIS node before time $t_c - \delta \log n$, it could have informed w in the interval $\mathcal{I}_2 = [t_c - \delta \log n, \dots, t_c - 1]$ with probability $1 - O(n^{-3})$ by Lemma 5.3.

In other words, either w sends successfully before $t_c + \delta \log^2 n$, increases its *count* to $\delta \log^2 n$, and therefore joins the MIS before time $t_c + 2\delta \log^2 n$; or there is a node x that joined $\mathcal{M}$ after $t_c - 2\delta \log n$. This claim holds with probability $(1 - O(n^{-3}))^2 \in 1 - O(n^{-3})$. Finally, the proof is concluded using the fact that $x \in N^2_v$ and $w \in N^+_v$, thus both nodes are in $E^{2.5}_i$. □

Having proven Lemma 5.4 allows us derive that once a node becomes a candidate, it either joins the MIS or becomes covered shortly thereafter, if all three Properties hold.

Lemma 5.5. *Assume Properties 1, 2, and 3 hold. Let t be an arbitrary time-slot. Every node v that is a candidate at time t, i.e., $v \in \mathcal{C}$, will either have joined $\mathcal{M}$ or be covered by time $t + 4\mu\delta \log^2 n$ with probability $1 - O(n^{-2})$.*

Proof. See Appendix C for the proof. □

We now return to the notion of a *clearance* which will be crucial in proving the validity of Properties 1 and 2. In particular, we use the two previous Lemmas to bound the number of clearances that can occur in a disk D_i during the execution of the algorithm.

Lemma 5.6. *Assume Properties 1, 2, and 3 hold. For all disks D_i, there are no more than μ clearances in D_i with probability $1 - O(n^{-2})$.*

Proof. See Appendix D for the proof. □

All previous lemmas were derived under the condition that Properties 1, 2, and 3 hold. In the following sequence of Theorems 5.7, 5.8, and 5.10, we will show that with high probability none of them is the *first* to be violated.

Theorem 5.7. *Assume Property 1 is the first property to be violated and let t_1 be the first time-slot in which the violation occurs. The probability that there exists such a time-slot t_1 during the execution of Algorithm 1 is at most $P_{fail} \in O(n^{-1})$.*

Proof. If t_1 is the first time-slot in which the violation occurs in a disk D_i, it holds $\sum_{v \in \mathcal{A}_i} p_v(t_1 - 1) \leq 2^{-\alpha}$ and $\sum_{v \in \mathcal{A}_i} p_v(t_1) \ > \ 2^{-\alpha}$. Consider the interval $\mathcal{I} = [t_1 - \lambda \log n, \dots, t_1 - 1]$. By the definition of Algorithm 1 (Lines 6 and 7), every active node $v \in \mathcal{A}_i$ doubles its sending probability p_v exactly once during this interval $\mathcal{I}$. Additionally, new nodes that were previously in state $\mathcal{W}_i$ may join the set $\mathcal{A}_i$ during $\mathcal{I}$, but these nodes' combined sending probability is at most $n \cdot \frac{2^{-\alpha-1}}{n} = 2^{-\alpha-1}$, according to the definition of a node's initial sending probability. That is, the sum of sending probabilities at time $t_1 - \lambda \log n$ is at least

$$\sum_{v \in \mathcal{A}_i} p_v(t_1 - \lambda \log n) \geq \frac{1}{2}(2^{-\alpha} - 2^{-\alpha-1}) = 2^{-\alpha-2}.$$

Consequently *if* Property 1 is violated, there must be an interval $\mathcal{I}$ preceding the violation during which the sum of the sending probabilities is in the range

$$2^{-\alpha-2} \ \leq \ \sum_{v \in \mathcal{A}_i} p_v(t) \ \leq \ 2^{-\alpha} \quad \forall t \in \mathcal{I}. \qquad (1)$$

In all neighboring disks $D_j \in E^{1.5}_i$, the sum of sending probabilities is

$$0 \ \leq \ \sum_{v \in \mathcal{A}_j} p_v(t) \ \leq \ 2^{-\alpha} \quad \forall t \in \mathcal{I} \qquad (2)$$

because t_1 is the first time-slot in which Property 1 is violated.

We continue the proof by showing that with high probability, a clearance occurs in the interval $\mathcal{I}$. For that purpose, let P_{no} be the probability that in a given time-slot $t \in \mathcal{I}$ no node in $E^{1.5}_i \setminus D_i$ sends. By P_{one} we denote the probability that exactly one node in D_i sends in t. The probability P_{clear} of a clearance at time t is $P_{clear} = P_{one} \cdot P_{no}$. Using Fact 1, the probabilities P_{one} and P_{no} can be bounded as follows:

$$\begin{aligned} P_{one} &= \sum_{v \in \mathcal{A}_i} \left(p_v \prod_{w \in D_i \setminus \{v\}} (1 - p_w) \right) \\ &\geq \sum_{v \in \mathcal{A}_i} p_v \prod_{w \in D_i} (1 - p_w) \\ &\underset{\text{Fact 1}}{\geq} \sum_{v \in \mathcal{A}_i} p_v (1/4)^{\sum_{w \in D_i} p_w} \\ &\geq \sum_{v \in \mathcal{A}_i} p_v (1/4)^{\sum_{w \in \mathcal{A}_i} p_w + \frac{1+\tau\kappa}{2^\alpha}}, \end{aligned}$$

where the last inequality is valid because of $\sum_{w \in \mathcal{C}_i} p_w \leq \tau\kappa/2^\alpha$ and $\sum_{w \in \mathcal{M}_i} p_w \leq 1/2^\alpha$ under the assumption that Properties 2 and 3 hold.

$$\begin{aligned} P_{no} &\geq \prod_{v \in E_i^{1.5}} (1 - p_v) \geq \prod_{D_j \in E_i^{1.5}} \prod_{v \in D_j} (1 - p_v) \\ &\geq \prod_{D_j \in E_i^{1.5}} (1/4)^{\sum_{v \in D_j} p_v} \\ &\geq \left[(1/4)^{\sum_{v \in \mathcal{A}_j} p_v + \frac{1+\tau\kappa}{2^\alpha}} \right]^\mu \geq (1/4)^{\frac{\mu(2+\tau\kappa)}{2^\alpha}} \end{aligned}$$

where the last step follows from (2). The probability of $t \in \mathcal{I}$ being a clearance is therefore at least

$$P_{clear} \geq \sum_{v \in \mathcal{A}_i} p_v (1/4)^{\sum_{w \in \mathcal{A}_i} p_w + \frac{1+\tau\kappa}{2^\alpha}} \cdot (1/4)^{\frac{\mu(2+\tau\kappa)}{2^\alpha}}.$$

For $x \in [2^{-\alpha-2}, \ldots, 2^{-\alpha}]$, $x(1/4)^{x + \frac{1+\tau\kappa}{2^\alpha}}$ is minimized for $x = 2^{-\alpha-2}$ and when applying (1), we get

$$\begin{aligned} P_{clear} &\geq 2^{-\alpha-2} (1/4)^{2^{-\alpha-2} + \frac{1+\tau\kappa}{2^\alpha}} \cdot (1/4)^{\mu \cdot \frac{2+\tau\kappa}{2^\alpha}} \\ &= 2^{-\alpha-2} (1/4)^{\frac{\frac{5}{4} + 2\mu + (1+\mu)\tau\kappa}{2^\alpha}}. \end{aligned}$$

The probability P_x that *none* of the $\lambda \log n$ time-slots $t \in \mathcal{I}$ is a clearance is therefore at most $P_x \geq (1 - P_{clear})^{\lambda \log n} \in O(n^{-3})$ by the definitions of λ and τ. Notice that the reason for defining $\alpha = 6.4$ is that this value maximizes P_{clear}.

Unfortunately, the argument that in every critical interval $\mathcal{I}$ occurs a clearance with probability $1 - n^{-3}$ is not sufficient. Potentially, the number of intervals $\mathcal{I}$ could be infinitely large, rendering the $1 - O(n^{-3})$ high probability result useless. However, the probability that in the first μ intervals $\mathcal{I}$ in D_i, there is at least one *without a clearance* is at most $\mu n^{-3} \in O(n^{-3})$. By Lemma 5.6, there are no more than μ clearances in D_i with probability $1 - O(n^{-2})$. Thus, there is no time-slot t_1 in D_i during the execution of Algorithm 1 with probability at least $1 - O(n^{-2})$. Because the same argument can be applied for all D_i and because there are at most n such disks covering all $v \in V$, the claim holds for all disks with probability $1 - O(n^{-1})$. □

We continue by showing that Property 2 holds under the assumption that the two other Properties hold.

Theorem 5.8. *Assume Property 2 is the first property to be violated and let t_2 be the first time-slot in which the violation occurs. The probability that there exists such a time-slot t_2 during the execution of Algorithm 1 is at most $P_{fail} \in O(n^{-1})$.*

Before actually proving Theorem 5.8, we will introduce some notation and establish a key lemma. Assume that T_c is an interval between either a) two subsequent clearances in a disk D_i, or b) between a clearance and the end of the algorithm, or c) between a clearance and time-slot t_2 (i.e., the first violation of Property 2), depending on which comes first. Further, let t_c be the clearance that has initiated T_c. We show that the probability of Property 2 being violated in this interval (i.e., $t_2 \in T_c$) is $1 - O(n^{-3})$. By Lemma 5.2, there is no new candidate emerging in D_i in the interval $[t_c, \ldots, t_c + 4\mu\delta \log^2 n]$. We therefore need to analyze only the interval $[t_c + 4\mu\delta \log^2 n, \ldots, t_q]$, where t_q is the time-slot of a) the subsequent clearance, b) the end of the algorithm, or c) time-slot t_2.

Let a *failure* be a time-slot in which at least one new candidate in D_i emerges, but no clearance occurs. The next lemma bounds the number of failures.

Lemma 5.9. *There are no more than $\frac{1}{6e}\kappa \log n$ failures in D_i in the interval $[t_c + 4\mu\delta \log^2 n, \ldots, t_q]$ with probability $1 - n^{-3}$.*

Proof. See Appendix E for the proof. □

Proof of Theorem 5.8. We begin by showing that in expectation, there are only a constant number of new candidates emerging in D_i per failure time-slot. We denote this number by $C(t)$ the number of active nodes that send at time t (new candidates) and write $\mathcal{E}_f(t)$ for the event of a failure. The conditional expected value of $C(t)$ given a failure is

$E[C(t)|\mathcal{E}_f(t)] \leq E[C(t)] + 2 \leq \sum_{v \in \mathcal{A}_i} p_v + 2$. By the assumption that Property 1 holds, this is at most $E[C(t)|\mathcal{E}_f(t)] \leq 2^{-\alpha} + 2$.

In the following, we bound the number of candidates $C(T_C)$ emerging in the case in which during the interval $T_C = [t_c + 4\mu\delta \log^2 n, \ldots, t_q]$, there are no more than $\frac{1}{6e}\kappa \log n$ failures. Observe that bounding $C(T_C)$ suffices to prove the theorem because by Lemma 5.5, all candidates existing at time t_c are covered by the time $t_c + 4\mu\delta \log^2 n$ with probability $1 - n^{-2}$. Consequently, we only need to bound the number of *new* emerging candidates when analyzing the interval $[t_c + 4\mu\delta \log^2 n, \ldots, t_2]$.

The following random experiment allows us to derive a high probability bound on $C(T_C)$. We consider random variables X_{ij} for $i = 1 \ldots n$ and $j = 1 \ldots |C|$, where $|C|$ is defined as the number of failures in T_C, formally $|C| = |\{t \in T_C | \mathcal{E}_f(t)\}|$. Further, we define $X := \sum_{j=1}^{|C|} \sum_{i \in \mathcal{A}_i(t_j)} X_{ij}$ as the sum of all X_{ij}. The semantic meaning of X_{ij} is that $X_{ij} = 1$, if node i sends (and becomes a candidate) in the j^{th} failure of T_C, and $X_{ij} = 0$ otherwise. Therefore, X stands for the number of new candidates emerging in D_i during T_C. Considering the X_{ij} as being independently distributed Bernoulli trials is not precise because of dependencies between different X_{ij}. Specifically, $X_{ij} = 1 \Rightarrow X_{ij'} = 0$, for all $j' > j$ because an active node that sends becomes a candidate. Note that these dependencies cause X to be strictly smaller or equal as compared to the case in which all X_{ij} that are depending on previous events were chosen randomly and independently with an arbitrary probability distribution. Thus, when assuming all X_{ij} to be independent Bernoulli trials, X is an upper bound for $C(T_C)$.

We know from the above argument, that in expectation, at most $2^{-\alpha} + 2$ active nodes send per failure. With our assumption that there are no more than $\frac{1}{6e}\kappa \log n$ failures, we get

$$E[X] = \sum_{j=1}^{|C|} E\left[\sum_{i \in \mathcal{A}_i(t_j)} X_{ij}\right] = \sum_{j=1}^{|C|} E[C(t_j)|\mathcal{E}_f(t_j)]$$
$$= (2^{-\alpha} + 2)\frac{1}{6e}\kappa \log n < \frac{1}{2e+1}\kappa \log n.$$

As mentioned before, assuming X_{ij} to be randomly and independently distributed Bernoulli variables yields an upper bound on $C(T_C)$, $E[C(T_C)] \leq E[X]$. Hence, we can use the Chernoff bound [20] with $E[X] = \frac{\kappa \log n}{2e+1}$. In particular, the probability P_x that X is larger than $(2e+1)\frac{\kappa \log n}{2e+1} = \kappa \log n$ is at most

$$P[C(T_C) > \kappa \log] \leq \left(\frac{e^{2e}}{(2e+1)^{2e+1}}\right)^{\frac{\kappa \log n}{2e+1}} \in O(n^{-2}).$$

That is, *if* there are at most $\frac{1}{6e}\kappa \log n$ failures in T_C, then at most $\kappa \log n$ candidates emerge in T_C with probability $1 - O(n^{-2})$. As shown, this bound suffices to prove that Property 2 is not violated in T_C with probability $1 - O(n^{-2})$. On the other hand, we know by Lemma 5.9 there probability of there being *more than* $\frac{1}{6e}\kappa \log n$ failures in T_C is $1 - O(n^{-2})$. That is, the probability that in an arbitrary interval T_C after a clearance, Property 2 is indeed violated *before* the next clearance is at most $1 - 2O(n^{-2}) \in 1 - O(n^{-2})$.

Now, consider the first μ intervals T_C in *every disk* D_i. The probability that there is at least one interval in which there are more than $\kappa \log n$ new candidates is at most $n\mu \cdot O(n^{-2}) \in O(n^{-1})$. That is, with probability $1 - O(n^{-1})$, Property 2 is *not violated* after the first μ intervals T_C in every disk D_i. By Lemma 5.6, there are no more than μ clearances (and inter-clearance intervals T_C) per disk with probability $1 - O(n^{-2})$ if all three Properties hold. Thus, Property 2 is *not* the first Property to be violated with probability $1 - O(n^{-1})$. □

Finally, we prove the correctness of Property 3.

Theorem 5.10. *Assume Property 3 is the first property to be violated and let t_3 be the first time-slot in which the violation occurs. The probability that there exists a time-slot t_3 during the execution of Algorithm 1 is at most $P_{fail} \in O(n^{-2})$.*

Proof. We show that if a node joins $\mathcal{M}$, the *count* value of all neighboring candidates are at least $\delta \log n$ away from the threshold that enables to join $\mathcal{M}$. Applying Lemma 5.3 then concludes the proof. Let v_v be the node that violates Property 3 at time t_3 and let v_m be the neighbor of v_v that has previously joined $\mathcal{M}$, say at time $t_m \leq t_v$. We claim that at time t_m, the *count* value of all neighbors of v_m (including v_v) is at most $\delta \log^2 n - \delta \log n$. By the definition of the algorithm, v_m must have started increasing its *count* by the time $t_m - \delta \log^2 n + \delta \log n$, because it could "skip" at most the first $\delta \log n$ values. Similarly, every potential node v_x that ends up having a *count* value larger than $\delta \log^2 n - \delta \log n$ at time

t_m must have started increasing its $count$ by the time $t_m - \delta \log^2 n + 2\delta \log n$. By the definition of the critical range $\delta \log n$ in Receive Trigger 2, such a node v_x has not received a message m_C from v_m in the interval $[t_m - \delta \log^2 n + 2\delta \log n, \ldots, t_m]$ because if it had, it would have reset its $count$.

The probability P_t that $v_x \in D_i$ receives a message m_C from v_m in an arbitrary time-slot t in the interval $[t_m - \delta \log^2 n + 2\delta \log n, \ldots, t_m]$ is at least

$$\begin{aligned} P_t &\geq p_m \cdot \prod_{v \in E_i^{1.5}} (1 - p_v) \\ &\geq q_C \prod_{D_j \in E_i^{1.5}} (1/4)^{\sum_{v \in \mathcal{C}_j \cup \mathcal{A}_j} p_v + \sum_{v \in \mathcal{M}_j} p_v} \\ &\underset{\text{P 1, P 2}}{\geq} \frac{\tau}{2^\alpha \log n} \left[(1/4)^{\frac{(1+\kappa\tau)}{2^\alpha} + \sum_{v \in \mathcal{M}_j} p_v} \right]^\mu . \end{aligned}$$

Because v_v is the first node violating Property 3, it holds that before t_m, the set $\mathcal{M}$ forms a correct independent set. Therefore, due to Fact 3, $\sum_{v \in \mathcal{M}_j} p_v \leq \frac{1}{2^\alpha}$, and hence $P_t \geq 2^{-\alpha} \tau (\log n)^{-1} (1/4)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}}$. The probability P_n that a node v_x does *not* receive any message m_C from v_m during the $\delta \log^2 n - 2\delta \log n$ remaining time-slots before t_m joins $\mathcal{M}$ is for $n \geq 16$,

$$\begin{aligned} P_n &\leq \left(1 - \frac{\tau}{2^\alpha \log n} \left(\frac{1}{4} \right)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}} \right)^{\delta \log^2 n - 2\delta \log n} \\ &\leq \left(1 - \frac{\frac{\tau \log n}{2^\alpha}}{\log^2 n} \left(\frac{1}{4} \right)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}} \right)^{\frac{1}{2} \delta \log^2 n} \\ &\leq e^{-\frac{1}{2} \delta \log n \cdot \frac{\tau}{2^\alpha} \left(\frac{1}{4} \right)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}}} \in O(n^{-4}). \end{aligned}$$

Because there are n^2 pairs of nodes $(v_m, v_x) \in V \times V$, the probability that the "$count$-difference" claim holds for all nodes v_m and v_x is at least $1 - O(n^{-2})$.

We now have all ingredients to prove the theorem. Assume for contradiction that v_v is the first node to violate the MIS condition (Property 3) at time t_3 and let v_m be v_v's neighbor that has correctly joined $\mathcal{M}$ at time $t_m \leq t_3$. By definition of v_v, Property 3 holds until t_3. Therefore, we can apply the result obtained above. In particular, with probability $1 - O(n^{-2})$, there are at least $\delta \log n$ time-slots between v_m and any potential node v_v causing the violation. Because all properties hold before t_3, it follows from Lemma 5.3 that v_v receives a message m_C by $v_m \in \mathcal{M}$ with probability $1 - O(n^{-3})$. Hence, the probability that there exists a time-slot t_3 is bounded by $O(n^{-2})$. □

When the first node wakes up, all three Properties are valid. Theorems 5.7, 5.8, and 5.10 show that none of them is the first to be violated, thus establishing the algorithm's correctness. For the running-time, we show that *every node* decides in time $O(\log^2 n)$ whether to become a MIS node or a slave. The proofs of both Theorems are given in Appendices G and H.

Theorem 5.11. *Properties 1, 2, and 3 hold with probability* $1 - O(n^{-1})$*. Particularly, the set* $\mathcal{M}$ *as computed by Algorithm 1 is a correct maximal independent set with probability* $1 - O(n^{-1})$.

Theorem 5.12. *With probability* $1 - O(n^{-1})$*, every node* $v \in V$ *decides irrevocably whether it joins set* $\mathcal{M}$ *or* $\mathcal{S}$ *within time* $O(\log^2 n)$ *after its wake-up.*

Theorems 5.11 and 5.12 show that with high probability, Algorithm 1 computes a correct maximal independent set in time $O(\log^2 n)$.

Remark: The large constants in Algorithm 1 are an artefact of our worst-case analysis. Simulation results show that much smaller constants suffice to yield correctness. In fact, the constants are in an order of magnitude that renders Algorithm 1 practical even for time-critical applications.

6 Conclusions

The $\Omega(\log^2 n / \log \log n)$ lower bound of [11] even holds for the restricted problem of having one node send alone in a single-hop environment. It is intriguing that $O(\log^2 n)$ suffices to compute a sophisticated network structure like a MIS. We believe that our algorithm is asymptotically optimal, but closing the remaining gap of $O(\log \log n)$ in either way remains a challenging task. It will also be interesting to investigate the respective power of deterministic and randomized MIS algorithms and study other initial network structures.

Whereas in the message passing model, Luby's algorithm [17] computes a MIS in time $O(\log n)$, our algorithm requires time $O(\log^2 n)$ in the radio network model. This sheds an interesting new light on the relation between these two models of distributed computing, particularly because both algorithms are close from optimal [14, 11].

References

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

[2] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *Proceedings of the 30^{th} Symposium on Foundations of Computer Science (FOCS)*, pages 364–369, 1989.

[3] R. Bar-Yehuda, O. Goldreich, and A. Itai. On the time-complexity of broadcast in radio networks: an exponential gap between determinism randomization. In *Proceedings of the 6^{th} annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 98–108. ACM Press, 1987.

[4] B. S. Chlebus. *Randomized Communication in Radio Networks*, chapter in Handbook of Randomized Computing, pages 401–456. Kluwer Academic Publishers, 2001.

[5] B. S. Chlebus, L. Gasieniec, A. Östlin, and J. M. Robson. Deterministic Radio Broadcasting. In *Proceedings of the 27^{th} International Colloquium on Automata, Languages and Programming (ICALP)*, pages 717–728. Springer-Verlag, 2000.

[6] B. S. Chlebus and D. R. Kowalski. A Better Wake-Up in Radio Networks. In *Proceedings of 23^{rd} ACM Symposium on Principles of Distributed Computing (PODC)*, pages 266–274. ACM Press, 2004.

[7] M. Chrobak, L. Gasieniec, and D. Kowalski. The Wake-Up Problem in Multi-Hop Radio Networks. In *Proceedings of the 15^{h} ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 992–1000. Society for Industrial and Applied Mathematics, 2004.

[8] R. Cole and U. Vishkin. Deterministic Coin Tossing with Applications to Optimal Parallel List Ranking. *Inf. Control*, 70(1):32–53, 1986.

[9] A. Czumaj and W. Rytter. Broadcasting Algorithms in Radio Networks with Unknown Topology. In *Proceedings of the 44^{th} Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, page 492. IEEE Computer Society, 2003.

[10] L. Gasieniec, A. Pelc, and D. Peleg. The Wakeup Problem in Synchronous Broadcast Systems (Extended Abstract). In *Proceedings of the 19^{th} ACM Symposium on Principles of Distributed Computing (PODC)*, pages 113–121. ACM Press, 2000.

[11] T. Jurdzinski and G. Stachowiak. Probabilistic Algorithms for the Wakeup Problem in Single-Hop Radio Networks. In *Proceedings of 13^{th} Annual International Symposium on Algorithms and Computation (ISAAC)*, volume 2518 of *Lecture Notes in Computer Science*, pages 535–549, 2002.

[12] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing Newly Deployed Ad Hoc and Sensor Networks. In *Proceedings of 10^{th} Annual International Conference on Mobile Computing and Networking (MOBICOM)*, 2004.

[13] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Radio Network Clustering from Scratch. In *Proceedings of 12^{th} Annual European Symposium on Algorithms (ESA)*, 2004.

[14] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What Cannot Be Computed Locally! In *Proceedings of 23^{rd} Annual Symposium on Principles of Distributed Computing (PODC)*, pages 300–309, 2004.

[15] E. Kushilevitz and Y. Mansour. An $\Omega(D \log(N/D))$ Lower Bound for Broadcast in Radio Networks. *SIAM Journal on Computing*, 27:702–712, 1998.

[16] N. Linial. Locality in Distributed Graph Algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[17] M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM Journal on Computing*, 15:1036–1053, 1986.

[18] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple Heuris-

tics for Unit Disk Graphs. *Networks*, 25:59–68, 1995.

[19] T. Moscibroda and R. Wattenhofer. Efficient Computation of Maximal Independent Sets in Unstructured Multi-Hop Radio Networks. In *Proceedings of 1st International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 51–59.

[20] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[21] L. G. Roberts. Aloha Packet System with and without Slots and Capture. *ACM SIGCOMM, Computer Communication Review*, 5(2):28–42, 1975.

Appendix

A Geometric Facts

Throughout the proof, we make use of the following imaginary covering of the Euclidean plane with disks of radius $1/2$ denoted by D_i.

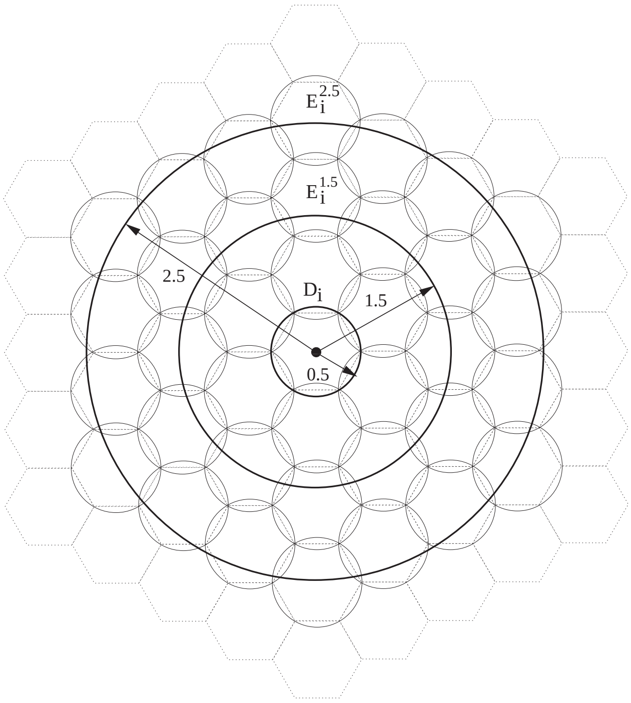

Figure 1: Circles D_i, $E_i^{1.5}$, and $E_i^{2.5}$

B Proof of Lemma 5.3

Proof. Consider a node $v \in D_i$ and let t_v be the time-slot defined in the lemma. The probability P_1 that in an arbitrary time-slot $t \in [t_v+1, \ldots, t_v+\delta \log n]$ MIS node $w \in N_v^+ \cap \mathcal{M}$ sends and no other node in N_v^+ sends (i.e., that v receives w's message m_M) is at least

$$P_1 \;\geq\; q_M \prod_{u \in N_v^+} (1-p_u) \underset{\text{Fact 1}}{\geq} \frac{1}{2^\alpha}\left(\frac{1}{4}\right)^{\sum_{u\in N_v^+} p_u}.$$

To bound $\sum_{u \in N_v^+} p_u$, we make use of the assumption that the three Properties hold and that nodes in $\mathcal{W} \cup \mathcal{S}$ do not send. It follows that

$$\begin{aligned}
\sum_{u\in N_v^+} p_u &= \sum_{u\in\mathcal{A}\cap N_v^+} p_u + \sum_{u\in\mathcal{C}\cap N_v^+} p_u + \sum_{u\in\mathcal{M}\cap N_v^+} p_u \\
&= \sum_{D_j \in E_i^{1.5}} \left(\sum_{u\in\mathcal{A}_j} p_u + \sum_{u\in\mathcal{C}_j} q_C + \sum_{u\in\mathcal{M}_j} q_M \right) \\
&\leq \sum_{D_j \in E_i^{1.5}} \left(\frac{1}{2^\alpha} + \kappa \log n \cdot \frac{\tau}{2^\alpha \log n} + \frac{1}{2^\alpha} \right) \\
&\leq \mu \left(\frac{1}{2^\alpha} + \frac{\tau\kappa \log n}{2^\alpha \log n} + \frac{1}{2^\alpha} \right) \;=\; \frac{\mu(2+\kappa\tau)}{2^\alpha},
\end{aligned}$$

where the first inequality is derived by replacing $q_C = \tau/(2^\alpha \log n)$ and $q_M = 2^{-\alpha}$ as defined in Algorithm 1. The second inequality follows from Fact 3. Plugging this in the inequality for P_1 yields

$$P_1 \;\geq\; 2^{-\alpha}\,(1/4)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}}.$$

The probability P_{no} that none of the $\delta \log n$ in the interval $[t_v+1, \ldots, t_v+\delta \log n]$ is successful is therefore upper bounded by

$$\begin{aligned}
P_{no} &\leq \left(1 - \frac{1}{2^\alpha}\left(\frac{1}{4}\right)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}} \right)^{\delta \log n} \\
&\leq e^{-\delta \log n \cdot \frac{1}{2^\alpha}(1/4)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}}}
\end{aligned}$$

which is, by the definitions of α, κ, τ, μ, and δ, $P_{no} \in O(n^{-4})$. Finally, the argument is concluded by the observation that every node needs to be informed about its being covered at most once. That is, the claim holds for all nodes $v \in V$ with probability $1 - O(n^{-3})$. □

C Proof of Lemma 5.5

Proof. Because all three Properties are assumed to be true, we can prove the claim by repeatedly applying Lemma 5.4. Consider a node $v \in D_i$ that is a candidate by at time t. We know by Lemma 5.4 that there is at least one MIS node in $E_i^{2.5}$ by $t + 2\delta \log^2 n$ with probability $1 - O(n^{-3})$. If v is covered by this new MIS node, the lemma holds. If not, v is still a candidate and hence, again by Lemma 5.4, there is a MIS node emerging in the interval $[t + 2\delta \log^2 n - \delta \log n, \ldots, t + 4\delta \log^2 n]$ with high probability. Thus applying Lemma 5.4 repeatedly yields that a MIS node emerges in the interval $[t+2i\delta\log^2 n-\delta\log n,\dots,t+2(i+1)\delta\log^2 n]$ with probability $1-O(n^{-3})$ for every $i\geq 0$, as long as v is uncovered. Since every emerging MIS node can cover only two "adjacent" intervals, it holds with probability $(1-O(n^{-3}))^i$ that there are at least $\lceil i/2\rceil$ new MIS nodes emerging in $E_i^{2.5}$ if v is still uncovered by time $t+2i\delta\log^2 n$. Due to Property 3, we assume the set $\mathcal{M}$ to be a correct independent set which means that there can be at most 2μ MIS nodes in $E_i^{2.5}$. Hence, it follows that with probability $(1-O(n^{-3}))^{2\mu}\in 1-O(n^{-3})$, v is covered by the time $t+4\mu\delta\log^2 n$. Since there are at most n candidates v, the Lemma holds for all nodes with probability $(1-O(n^{-3}))^n\in 1-O(n^{-2})$. □

D Proof of Lemma 5.6

Proof. By Lemma 5.2, there can be at most one clearance every $4\mu\delta\log^2 n$ time-slots in a disk D_i. Let t_c be a clearance of D_i. By definition, exactly one active node $v\in\mathcal{A}_i$ sends successfully in time-slot t_c. By definition of Algorithm 1, this node becomes a candidate and by Lemma 5.4, there is a node $w\in E_i^{2.5}$ that joins the MIS in the interval $[t_c-\delta\log n,\dots,t_c+2\delta\log^2 n]$ with probability $1-O(n^{-3})$. Hence, after μ clearances, at least μ MIS nodes have emerged in $E_i^{2.5}$ with probability $(1-O(n^{-3}))^{\mu}\in 1-O(n^{-3})$. By Fact 3 and under the assumption of Property 3, these μ MIS nodes entirely cover $E_i^{2.5}$ and all nodes located therein. Moreover, it follows from Lemma 5.3 that every such node receives a message m_M $\delta\log n$ time-slots after its becoming covered with probability $1-n^{-3}$. That is, no node will be in the active state $\mathcal{A}$ once all of $E_i^{2.5}$ is covered. With probability $1-O(n^{-3})$, this is the case after μ clearance. □

E Proof of Lemma 5.9

Proof. We prove the claim by showing that before $\frac{1}{6e}\kappa\log n$ failures can occur, there is with high probability at least one clearance. The argument is completed by the fact that, by definition, t_q must take place before or at the time of such a clearance.

We define the following events. $\mathcal{E}_c(t)$ denotes the event of a clearance in D_i at time-slot t and $\mathcal{E}_0(t)$ is the event of no node in $\mathcal{A}_i$ sending in time-slot t. Observe that $\mathcal{E}_c(t)$ can only be true if $\mathcal{E}_0(t)$ is false. In the sequel, we want to find a bound on the probability $P[\mathcal{E}_c(t)|\overline{\mathcal{E}_0}(t)]$. Clearly, if in a time-slot exactly one node in D_i sends and no other node in $E_i^{1.5}$ sends, then a clearance occurs. Hence, $P[\mathcal{E}_c(t)|\overline{\mathcal{E}_0}(t)]\geq P[\mathcal{E}_1(t)|\overline{\mathcal{E}_0}(t)]\cdot P[\mathcal{E}_e(t)]$ where $\mathcal{E}_1(t)$ is the event of *at most one* node sending in $\mathcal{A}_i$, and $\mathcal{E}_e(t)$ is the event of no node sending in $E_i^{1.5}\setminus\mathcal{A}_i$. It will be convenient to state the above expression in terms of $\mathcal{E}_+(t)$ which is the event that 2 or more nodes in $\mathcal{A}_i$ send. Thus,

$$\begin{aligned}P[\mathcal{E}_c(t)|\overline{\mathcal{E}_0}(t)] &\geq P[\mathcal{E}_e(t)]\cdot(1-P[\overline{\mathcal{E}_1}(t)|\overline{\mathcal{E}_0}(t)])\\ &= P[\mathcal{E}_e(t)]\cdot(1-P[\mathcal{E}_+(t)|\overline{\mathcal{E}_0}(t)])\\ &= P[\mathcal{E}_e(t)]\cdot\left(1-\frac{P[\mathcal{E}_+(t)]}{P[\overline{\mathcal{E}_0}(t)]}\right),\end{aligned}$$

because of $P[\overline{\mathcal{E}_0}(t)|\mathcal{E}_+(t)]=1$. By the definition of t_q (which is t_2 or earlier), we can assume that in the interval $[t_c+4\mu\delta\log^2 n,\dots,t_q]$, all three properties hold. Thus, we are allowed reuse some results that we have established based on the assumption that all three properties hold. First, we need a result on $P[\mathcal{E}_e(t)]$ from the proof of Theorem 1:

$$P[\mathcal{E}_e(t)]\;\geq\prod_{v\in E_i^{1.5}}(1-p_v)\;\geq(1/4)^{\mu\cdot\frac{2+\tau\kappa}{2^{\alpha}}}\,.$$

For succinctness, $\mathcal{X}_A=\sum_{v\in\mathcal{A}_i}p_v$ in the sequel. We obtain the following lower bound for $P[\overline{\mathcal{E}_0}(t)]$,

$$\begin{aligned}P[\overline{\mathcal{E}_0}(t)] &= 1-\prod_{v\in\mathcal{A}_i}(1-p_v(t))\\ &\geq 1-(1/e)^{\sum_{v\in\mathcal{A}_i}p_v}\;\geq\;1-(1/e)^{\mathcal{X}_A}\,.\end{aligned}$$

Finally, we consider $P[\mathcal{E}_+(t)]$,

$$\begin{aligned}P[\mathcal{E}_+(t)] &\leq P[\overline{\mathcal{E}_0}(t)]-\sum_{v\in\mathcal{A}_i}\left(p_v\prod_{w\in\mathcal{A}_i\setminus\{v\}}(1-p_w)\right)\\ &\leq 1-\prod_{v\in\mathcal{A}_i}(1-p_v)-\sum_{v\in\mathcal{A}_i}p_v\cdot 4^{-\sum_{v\in\mathcal{A}_i}p_v}\\ &\leq 1-4^{-\sum_{v\in\mathcal{A}_i}p_v}-\sum_{v\in\mathcal{A}_i}p_v\cdot 4^{-\sum_{v\in\mathcal{A}_i}p_v}\\ &\leq 1-(1+\mathcal{X}_A)\,4^{-\mathcal{X}_A}.\end{aligned}$$

Plugging everything together, the probability $P[\mathcal{E}_c(t)|\overline{\mathcal{E}_0}(t)]$ that there is a clearance if a new candidate emerges in D_i is at least $P[\mathcal{E}_c(t)|\overline{\mathcal{E}_0}(t)]\geq\mathcal{Q}$

where $\mathcal{Q}$

$$\mathcal{Q} \;=\; 4^{-\mu\frac{2+\tau\kappa}{2^\alpha}} \cdot \left(1 - \frac{1-(1+\mathcal{X}_A)\,4^{-\mathcal{X}_A}}{1-(1/e)^{\mathcal{X}_A}}\right),$$

By Property 1, we know that the expression $\mathcal{X}_A = \sum_{v\in\mathcal{A}_i} p_v$ is in the range $[0,\ldots,2^{-\alpha}]$. Under this condition, the above function is minimized for $\mathcal{X} = 2^{-\alpha}$ and therefore $P[\mathcal{E}_c(t)|\overline{\mathcal{E}_0}(t)] \geq 0.23 \cdot 4^{-\mu\frac{2+\tau\kappa}{2^\alpha}}$. Finally, the probability P_f that there are more than $\frac{1}{6e}\kappa\log n$ failures in D_i in the interval $[t_c + 4\mu\delta\log^2 n,\ldots,t_q]$ if $t_2 \in T_c$ is asymptotically in

$$\begin{aligned} P_f &\leq (1 - P[\mathcal{E}_c(t)|\overline{\mathcal{E}_0}(t)])^{\frac{1}{6e}\kappa\log n} \\ &\leq \left(1 - 0.23\cdot 4^{-\mu\frac{2+\tau\kappa}{2^\alpha}}\right)^{\frac{1}{6e}\kappa\log n} \leq n^{-3} \end{aligned}$$

by the definition of κ.

That is, with probability $1 - O(n^{-3})$, if as many as $\frac{1}{6e}\kappa\log n$ failures had occurred before t_q, there would have been another clearance *before* t_q which contradicts the definition of t_q. □

F Proof of Theorem 5.10

Proof. Let v_v be the node that violates Property 3 at time t_3 and let v_m be the neighbor of v_v that has previously joined $\mathcal{M}$, say at time $t_m \leq t_v$. We claim that at time t_m, the *count* value of all neighbors of v_m (including v_v) is at most $\delta\log^2 n - \delta\log n$. By the definition of the algorithm, v_m must have started increasing its *count* by the time $t_m - \delta\log^2 n + \delta\log n$, because it could "skip" at most the first $\delta\log n$ values. Similarly, every potential node v_x that ends up having a *count* value larger than $\delta\log^2 n - \delta\log n$ at time t_m must have started increasing its *count* by the time $t_m - \delta\log^2 n + 2\delta\log n$. By the definition of the critical range $\delta\log n$ in Receive Trigger 2, such a node v_x has not received a message m_C from v_m in the interval $[t_m - \delta\log^2 n + 2\delta\log n,\ldots,t_m]$ because if it had, it would have reset its *count*.

The probability P_t that $v_x \in D_i$ receives a message m_C from v_m in an arbitrary time-slot t in the interval $[t_m - \delta\log^2 n + 2\delta\log n,\ldots,t_m]$ is at least

$$\begin{aligned} P_t \;&\geq\; p_m \cdot \prod_{v\in E_i^{1.5}} (1-p_v) \\ &\geq\; q_C \prod_{D_j\in E_i^{1.5}} \left(\frac{1}{4}\right)^{\sum_{v\in\mathcal{C}_j\cup\mathcal{A}_j} p_v + \sum_{v\in\mathcal{M}_j} p_v} \\ &\underset{\text{P 1, P 2}}{\geq}\; \frac{\tau}{2^\alpha\log n}\left[\left(\frac{1}{4}\right)^{\frac{(1+\kappa\tau)}{2^\alpha}+\sum_{v\in\mathcal{M}_j} p_v}\right]^\mu . \end{aligned}$$

Because we assume v_v to be the first node violating Property 3, it holds that before t_m, the set $\mathcal{M}$ forms a correct independent set. Therefore, due to Fact 3, $\sum_{v\in\mathcal{M}_j} p_v \leq \frac{1}{2^\alpha}$, and hence

$$P_t \;\geq\; \frac{\tau}{2^\alpha\log n}\left(\frac{1}{4}\right)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}} .$$

The probability P_n that a node v_x does *not* receive any message m_C from v_m during the $\delta\log^2 n - 2\delta\log n$ remaining time-slots before t_m joins $\mathcal{M}$ is for $n \geq 16$,

$$\begin{aligned} P_n \;&\leq\; \left(1 - \frac{\tau}{2^\alpha\log n}\left(\frac{1}{4}\right)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}}\right)^{\delta\log^2 n - 2\delta\log n} \\ &\leq\; \left(1 - \frac{\frac{\tau\log n}{2^\alpha}}{\log^2 n}\left(\frac{1}{4}\right)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}}\right)^{\frac{1}{2}\delta\log^2 n} \\ &\leq\; e^{-\frac{1}{2}\delta\log n\cdot\frac{\tau}{2^\alpha}\left(\frac{1}{4}\right)^{\frac{\mu(2+\kappa\tau)}{2^\alpha}}} \;\in\; O(n^{-4}). \end{aligned}$$

Because there are at most n^2 pairs of nodes $(v_m, v_x) \in V \times V$, the probability that the "*count*-difference" claim holds for all nodes v_m and v_x is at least $1 - O(n^{-2})$.

We now have all ingredients to prove the theorem. Assume for contradiction that v_v is the first node to violate the MIS condition (Property 3) at time t_3 and let v_m be the neighbor of v_v that has correctly joined $\mathcal{M}$ at time $t_m \leq t_3$. Because v_v is the first node to violate Property 3, this property holds until time-slot t_3 (when v_v joins the MIS). Therefore, we can apply the result obtained above. In particular, we know that with probability $1 - O(n^{-2})$, there are at least $\delta\log n$ time-slots between v_m and any potential node v_v causing the violation. Because all properties hold before t_3, it follows from Lemma 5.3 that v_v receives a message m_C by $v_m \in \mathcal{M}$ with probability $1 - O(n^{-3})$.

Hence, the probability that there exists a time-slot t_3 as stated in the theorem is bounded by $O(n^{-2})$ which concludes the proof. □

G Proof of Theorem 5.11

Proof. By Theorem 5.7, Property 1 is not the first to be violated with probability $1 - O(n^{-1})$. Similarly, by Theorems 5.8 and 5.10, Properties 2 and 3 are not the first to be violated with probabilities $1 - O(n^{-1})$ and $1 - O(n^{-2})$, respectively. Hence, with probability $(1 - O(n^{-1}))^2 \cdot (1 - O(n^{-2})) \in 1 - O(n^{-1})$, none of the three properties is violated during the execution of the algorithm. If all three Properties hold, Property 3 implies that with probability $1 - O(n^{-1})$, the resulting set $\mathcal{M}$ forms a correct independent set. The maximality of the independent set stems from the fact that a a node joins set $\mathcal{S}$ only upon receiving a message m_M from a neighboring MIS node. That is, every non-MIS node has at least one MIS node in its neighborhood. □

H Proof of Theorem 5.12

Proof. Consider an arbitrary node $v \in D_i$ and let $T_\mathcal{W}$, $T_\mathcal{A}$, and $T_\mathcal{C}$ be the total time node v spends in the corresponding state during its execution of Algorithm 1. Assume that all three Properties hold. Once node v becomes a candidate at time t_v, it will decide to become a MIS node or a slave within time $t_v + 4\mu\delta \log^2 n$ by Lemma 5.5 with probability $1 - O(n^{-2})$. Hence, $T_\mathcal{C} \leq 4\mu\delta \log^2 n$. It therefore remains to bound the time that v spends in states $\mathcal{W}$ and $\mathcal{A}$.

If v does not receive a message m_A from a neighboring node for $4\mu\delta \log^2 n$ time-slots after its wake-up (or after being reset to state $\mathcal{W}$ in Receive Trigger 1), it becomes active and joins $\mathcal{A}$. Unless it receives a message m_A thereafter, its sending probability reaches the value $p_v(t) = 2^{-\alpha-2}$ at most $(\log n - 1) \cdot \lambda \log n$ time-slots after becoming active. This is because the sending probability is initially $2^{-\alpha-1}/n$ and is doubled once per $\lambda \log n$ time-slots. So, either there is a node $w \in N_v^+$ whose message m_A v has received and who subsequently becomes a candidate, or the sending probability of $v \in \mathcal{A}$ exceeds the value $p_v(t) = 2^{-\alpha-2}$. If the latter is the case, conditions (1) and 2 are fulfilled. It follows by the same argument as in the proof of Theorem 5.7 that there is a clearance in the subsequent $\delta \log n$ time-slots with probability $1 - O(n^{-3})$.

Putting things together, it holds that $4\mu\delta \log^2 n + \lambda \log^2 n$ time-slots after wake-up or a reset because of Receive Trigger 1, there exists a node $w \in N_v^+$ that becomes a candidate, say at time t_c. If $w = v$, we are done because v joins $\mathcal{C}$ and $T_\mathcal{C} \leq 4\mu\delta \log^2 n$. If $w \neq v$, a new MIS node appears in the interval $[t_c - \delta \log n, \ldots, t_c + \delta \log^2 n]$ in $E_i^{3.5}$ with probability $1 - O(n^{-2})$ by Lemma 5.4. Because there can be at most 4μ MIS nodes in $E_i^{3.5}$, the total time spent by v in states $\mathcal{W}$ and $\mathcal{A}$ is bounded by

$$T_\mathcal{W} + T_\mathcal{A} \;\leq\; 4\mu \cdot (4\mu\delta + \lambda) \log^2 n \;\in\; O(\log^2 n)$$

Together with the above bound on $T_\mathcal{C}$, this shows that *if* all three Properties hold, every node v decides within time $O(\log^2 n)$ upon its wake-up with probability $(1 - O(n^{-2}))^n \in 1 - O(n^{-1})$. By Theorem 5.11, all three properties hold with probability $1 - O(n^{-1})$ which concludes the proof. □