Efficient Computation of Maximal Independent Sets in Unstructured Multi-Hop Radio Networks

Thomas Moscibroda
Computer Engineering and Networks Laboratory
ETH Zurich, 8092 Zurich, Switzerland
moscitho@tik.ee.ethz.ch

Roger Wattenhofer
Computer Engineering and Networks Laboratory
ETH Zurich, 8092 Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch

***Abstract*— When being deployed, ad-hoc and sensor networks are unstructured and lack an efficient and reliable communication scheme. Hence, the organization of a MAC layer is the primary goal during and immediately after the deployment of such networks. Computing a good initial clustering facilitates this task and is therefore a vital part of the initialization process. A clustering based on a maximal independent set provides several highly desirable properties. Besides yielding a dominating set of good quality, such a clustering avoids interference between clusterheads, thus allowing efficient communication. We propose a novel algorithm that works under a model capturing the characteristics of the initialization phase of unstructured radio networks, i.e., asynchronous wake-up, scarce knowledge about the topology of the network graph, no collision detection, and the hidden terminal problem. We show that even under these hard conditions, the algorithm computes a maximal independent set in polylogarithmic time.**

I. Introduction

One of the main characteristics of ad-hoc and sensor networks is that the communication infrastructure is provided by the nodes themselves. When being deployed, the nodes of such networks initially form a chaotic *unstructured radio network*, which means that no reliable and efficient communication pattern has been established yet. Before any reasonable communication can be carried out, nodes must *structure* the network, i.e., they must set up a medium access scheme. The problem of initializing and structuring radio networks is of great importance in practice. Even in an ad-hoc network with a small number of devices such as Bluetooth, the initialization tends to be slow. In a multi-hop scenario with large number of nodes the time consumption for establishing a reasonable communication pattern increases even further. In this paper, we are going to study this vital transition from a unstructured to a structured network, the *initialization phase*.

One frequent approach to solving the problem of bringing structure into a multi-hop radio network is *clustering* [1], [2], [3], [4], [5]. Clustering allows the formation of virtual backbones enabling efficient routing and broadcasting [6], it improves the usage of scarce resources such as bandwidth and energy [7], and — most important to this paper — clustering is crucial in realizing spatial multiplexing in non-overlapping clusters (TDMA or FDMA). Hence, computing a good initial clustering is a major step towards establishing an efficient MAC layer on top of which higher-level protocols and applications can subsequently be built.

What is a good clustering? Depending on the specific network problem at hand, the answer to this question may be varying. But in light of the wireless and multi-hop nature of ad-hoc and sensor networks, a good clustering should satisfy (at least) two properties. In order to allow efficient communication between each pair of nodes, every node should have at least one clusterhead in its neighborhood. From a graph theory point of view, this first property demands a *dominating set* (and preferably a *minimum dominating set*). A dominating set in a graph $G = (V, E)$ is a subset $S \in V$ such that each node is either in S or has a neighbor in S. The use of (connected) dominating sets for clustering networks has been motivated and investigated in literature, e.g. in [8], [9], [10], [11], [12], [13].

This identification of clustering to the notion of a *dominating set*, however, does not cover the second need arising in ad-hoc and sensor networks. It has been motivated in [2], [14] that it is undesirable to have neighboring clusterheads. In particular, if no two clusterheads are within each other's mutual transmission range, the task of establishing an efficient MAC layer is greatly facilitated because clusterheads will not face interference. This second property imposed on the clustering of ad-hoc and sensor networks leads to the well-known concept of a *maximal independent set* in a graph $G = (V, E)$. An independent set (IS) S of G is a subset of V such that $\forall u, v \in S, (u, v) \notin E$. S is a *maximal independent set* (MIS) if any node v not in S has a neighbor in S.

The importance of a MIS in the context of clustering wireless networks has been widely acknowledged [14], [15]. Several algorithms for the construction of a virtual backbone (for example for allowing efficient routing) are based on computing a MIS [8], [16], [12], [17]. Due to its additional constraint, computing a MIS is a harder problem than computing a dominating set. Additionally, it is worth noting that any MIS is a $4\mathcal{O} + 1$-approximation for the minimum dominating set problem on the unit disk graph [12], where $\mathcal{O}$ denotes the size of the optimal solution. In other words, by computing a MIS, we obtain a clustering which has all advantages of

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

0-7803-8815-1/04/$20.00 ©2004 IEEE

a high-quality dominating set and moreover has the property that clusterheads do not interfere. Hence, a MIS provides an excellent *initial clustering*. Note that the computation of a MIS is also a key building block for coloring algorithms as all nodes in a MIS can be safely assigned the same color.

In view of our goal of setting up a MAC scheme in a newly deployed network, it is obvious that a clustering algorithm for the initialization phase must not rely on any previously established MAC layer. Instead, we are interested in a simple and practical algorithm which quickly computes a clustering completely from *scratch*. Note that this precludes algorithms working under any sort of *message passing model* in which messages can be sent to neighbors without fearing collision due to the hidden terminal problem.

In total absence of any MAC layer support, algorithms for clustering in a newly deployed network must be capable of working under particularly harsh conditions. Specifically, these conditions are captured by the following model assumptions [18]:

- The network is a *multi-hop* network, that is, there exist nodes that are not within their mutual transmission range, resulting in problems such as the well-known *hidden terminal problem*. Some neighbors of a sending node may receive a message, while others are experiencing interference from other senders and do not receive the message.
- Our model allows nodes to wake-up *asynchronously*. In a multi-hop environment, it is realistic to assume that some nodes wake up (e.g. become deployed, or switched on) later than others. Consequently, nodes do not have access to a global clock. It is important to observe the manifold implications of asynchronous wake-up. If all nodes started the algorithm simultaneously, we could easily assume an ALOHA style MAC-layer where each node sends with probability $\Theta(1/n)$. It is well known that this approach leads to a quick and simple communication scheme on top of which efficient clustering algorithms can be used. If nodes wake-up asynchronously, however, the same approach results in an expected linear runtime if only one single node wakes-up for a long time. In order to achieve a polylogarithmic runtime in the case of asynchronous wake-up, more sophisticated protocols are required.
- Nodes do not feature a reliable *collision detection* mechanism. In many scenarios (particularly when considering the lack of an established MAC protocol during the initialization phase!) not assuming any collision detection mechanism is realistic. Nodes may be tiny sensors with equipment restricted to the minimum due to limitations in energy consumption, weight, or cost. It has further been argued that the no collision detection assumption makes sense in the presence of noisy channels [19]. The sending node itself does not have a collision detection mechanism either, that is, a sender does not know how many (if any at all!) neighbors have received its transmission correctly. Given these additional limitations, algorithms without collision detection tend to be less efficient than algorithms with collision detection. Note that the absence of a reliable collision detection mechanism precludes using standard protocols such as Busy Tone Multiple Access (BTMA) [20].
- Nodes have only limited knowledge about the total number of nodes in the network and no knowledge about the nodes' distribution or wake-up pattern. Particularly, they have no a-priori information about the number of neighbors.

In this paper, we show that polylogarithmic time is enough to compute a MIS clustering even under this harsh model. We present a randomized algorithm which has practical relevance in the initialization phase of ad-hoc and sensor networks due to its being fast and simple and because it works in total absence of any existing MAC layer.

A literature review is given in Section II. Section III formally introduces the model in detail and the algorithm is presented in Section IV. The algorithm's analysis is given in Sections V and VI. Section VII concludes the paper.

II. Related Work

Before being studied in the context of clustering ad-hoc and sensor networks, the computation of a MIS has been the focus of extensive research on parallel complexity. It has been shown in [21] that the MIS problem is in $\mathcal{NC}$, meaning that a polylogarithmic running time is achievable on a PRAM containing a polynomial number of processors. A major breakthrough in the understanding of the computational complexity of MIS was the ingenious randomized algorithm by Luby [22], achieving a runtime of $\mathrm{O}(\log n)$ on a linear number of processors under the CRCW PRAM model of computation. Unfortunately, Luby's algorithm cannot be easily transformed to work under our model since it assumes synchronous wake-up, knowledge about the neighborhood, and collision-free communication. Recently, time lower bounds for the distributed construction of MIS have been given in [23]. At least $\Omega(\sqrt{\log n/\log\log n})$ and $\Omega(\log\Delta/\log\log\Delta)$ communication rounds are required to obtain a MIS, Δ being the largest degree in the network.

A model related to the one used in this paper has been studied in the context of analyzing the complexity of broadcasting in multi-hop radio network yielding a vast and rich literature, e.g. [24], [25]. The same model has also been the focus of research on two important problems called *initialization problem* and *leader election problem* in single-hop radio networks, e.g. [26], [27]. A striking difference to our model is that these algorithms consider *synchronous wake-up*, i.e., nodes have access to a global clock and it is assumed that all nodes start the distributed algorithm at the same time. In the case of ad hoc and sensor networks distributed over a large geographical area, guaranteeing that all nodes start the distributed algorithm simultaneously appears to be difficult in practice. Moreover, if (sensor) nodes are deployed dispersed in time, it may even be impossible. As mentioned in the

introduction, the additional difficulties imposed by asynchrony lead to new algorithmic designs.

A model featuring asynchronous wake-up has been studied in recent papers about the *wake-up problem* in single-hop radio networks [28], [29]. In comparison to our model, these papers define a much weaker notion of asynchrony. Particularly, it is assumed that sleeping nodes are *woken up* by a successfully transmitted message. In a single-hop network, the problem of waking up all nodes hence reduces to successfully transmitting one single message. Recently, the wake-up problem has also been studied in the multi-hop case in [30]. While this definition of asynchrony leads to a variety of interesting combinatorial problems, it it unsuited for modeling the situation of newly deployed ad hoc and sensor networks.

Our results for the construction of a maximal independent set in unstructured radio networks partly build on a previous paper for the same model [18]. In [18], a randomized algorithm which computes a constant approximation for the minimum dominating set problem in the same model is introduced. In fact, in Section V-A we are going to make use of some of the results obtained in [18] as a starting point for the analysis of our algorithm for maximal independent sets. Concluding this section, we would also like to mention that a model similar in spirit, yet even in more restricted than the one used in this paper has recently been studied in [31].

III. Model

Having already given some intuition in Section I, we now describe the model in more detail. We consider *multi-hop* radio networks *without 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. Further, in Sections IV and V, we assume that nodes have access to three independent communication channels Γ_1, Γ_2, and Γ_3. In practice, this may be realized using an FDMA scheme. Having three communication channels simplifies the analysis, but in Section VI we show it is not a fundamental necessity. Even a single communication channel suffices to compute a MIS in polylogarithmic time.

Nodes may wake up *asynchronously* at any time. We call a node *sleeping* before its wake-up, and *awake* thereafter. Only awake nodes can send or receive messages. Sleeping nodes are *not woken up* by incoming messages. Observe that this asynchronous model is more general than the usually studied synchronous wake-up model in which all nodes start their local algorithm at the same time. In fact, synchronous wake-up is just one possible scenario captured by the asynchronous model. In the other extreme case, only one of the n nodes may wake up while the others remain sleeping for an arbitrarily long time.

We consider *Unit Disk Graphs* (UDG) to model the network. In a UDG $G = (V, E)$, there is an edge $(u, v) \in E$ iff the Euclidean distance between u and v is at most 1. It is important to note however that due to asynchronous wake-up, some nodes may still be asleep, while others are already sending. Hence, at any time, there may be sleeping nodes which do not receive a message in spite of their being within the transmission range of the sender.

Nodes have only scarce knowledge about the network graph. In particulary, they have no information on the number of nodes in their neighborhood or even the density of nodes in the network. Nodes merely have an upper bound $\hat{n}$ for the total number of nodes $n = |V|$ in the graph. While n is unknown, all nodes have the same estimate $\hat{n}$. It has been shown in [29] that without any estimate of n, even in the single-hop case every algorithm requires at least time $\Omega(n/\log n)$ until one single message can be transmitted without collision. Hence, assuming n being completely unknown would ultimately preclude polylogarithmic clustering algorithms. In practice, is it usually possible to give a rough upper bound on the number of nodes in the network in advance. Further, note that nodes can be placed *completely arbitrarily*, i.e., our analysis does not rely on any kind of probabilistic (e.g. uniform) node distribution.

For the sake of simplicity, we assume — for the analysis of the algorithm — that time is divided into time-slots. However, we attach great importance to the observation that our algorithm *does not rely on synchronized time-slots* in any way. Since nodes do not have access to a global clock and synchronizing time-slots may be an expensive task, such an assumption would not always be realistic[1]. In this paper, it is solely for the purpose of analyzing the algorithm that we assume slotted channels. This simplification of the analysis is justified by the standard trick introduced in the analysis of slotted vs. unslotted ALOHA [32]. In [32], it is shown that the realistic unslotted case and the idealized slotted case differ only by a factor of 2, because a single packet can cause interference in no more than two successive time-slots. Similarly, by analyzing our algorithm in an "ideal" setting with synchronized time-slots, we obtain a result which is only by a constant factor faster as compared to the unslotted setting.

In each time-slot, a node can either send or not send. A node receives a message in a time-slot only if exactly one node in its neighborhood has sent a message in this time-slot. However, in the multi-channel case, nodes are able to correctly receive messages from the three channels simultaneously. The variables p_v and q_v denote the probabilities that node v sends a message in a given time-slot on channel Γ_1 and Γ_2, respectively.

We conclude the section with a well-known mathematical fact.

Fact 3.1: For all n, t, such that $n \geq 1$ and $|t| \leq n$,

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

IV. Algorithm

We start with an intuitive outline of the algorithm. For the sake of clarity, the algorithm is divided into three parts.

[1]It has been argued that by interfacing with a Global Positioning System (GPS), keeping local clocks synchronized has become technically possible. The accuracy provided by commercially available systems is more than sufficient for stations of a radio network to synchronize [27]. Nonetheless, our algorithm does not rely on this assumption.

Algorithm 1 MIS-Algorithm (Main-Loop)

state := uncovered; excited := false;
upon wake-up do:
1: **for** $j := 1$ to $2\delta \cdot \lceil \log^3 \hat{n} / \log\log \hat{n} \rceil$ **do**
2: wait();
3: **od**
4: counter := 0;
5: **for** $j := \lceil \log \hat{n} \rceil$ to 0 by -1 **do**
6: $p := 1/\left(2^{j+\beta}\right)$;
7: **for** $i := 1$ to $\gamma \cdot \lceil \log \hat{n} \rceil$ **do**
8: $b := \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases}$
9: **if** $b = 1$ **then**
10: **send**() on Γ_1;
11: **start candidacy**();
12: **stop executing main-loop**;
13: **fi**
14: **od**
15: **od**

Candidacy Phase():
16: **loop**
17: $b := \begin{cases} 1 & \text{with probability } q \\ 0 & \text{with probability } 1-q \end{cases}$
18: **if** $b = 1$ **then**
19: excited := true;
20: **send**$(counter)$ on Γ_2;
21: **fi**
22: **if** excited **then**
23: counter := counter + 1;
24: **fi**
25: **if** counter = $\delta \cdot \lceil \log^3 \hat{n} / \log\log \hat{n} \rceil$ **then**
26: state := MIS;
27: **send** on Γ_3 with probability $1/6$ forever;
28: **fi**
29: **end loop**

Receive Triggers:
(Only executed if the node does not send a message in the same time-slot.)
upon receiving msg on Γ_1 **do:**
  if not candidate **then**
  restart main-loop at line 1;
  fi

upon receiving msg (c') **on** Γ_2 **do:**
  $\Delta c := |c' - \text{counter}|$;
  if candidate **and** $\Delta c \leq 8 \log \hat{n}$ **then**
  counter := $-\lceil 8 \log \hat{n} \rceil$;
  fi

upon receiving msg on Γ_3 **do:**
  state := covered;
  terminate();

Each node first executes the main-loop. When sending the first message, the main-loop is stopped and the *start candidacy()* procedure is called, which runs until termination. In parallel, the algorithm's reception triggers are invoked upon receiving any messages. Note however, that in accordance to our model, a message can only be received (i.e., the reception trigger invoked) if the node *does not* send in the same time-slot.

The algorithm consists of two main phases. The purpose of the main loop is the selection of *candidates* which will subsequently compete for joining the MIS in a *candidacy-phase*. More precisely, a node becomes candidate when sending its first message on channel Γ_1 (lines 10 and 11). The main-loop is designed as to bound the number of candidates simultaneously executing the candidacy-phase, therefore enabling a quick election of MIS nodes. This selection in the candidacy-phase takes place entirely on channel Γ_2. While Γ_1 and Γ_2 correspond to communication in the main-loop and in the candidacy-phase, respectively, Γ_3 is reserved for nodes having already joined the MIS.

Note that due to asynchronous wake-up, the candidacy-phases of different nodes are not aligned with each other. On the contrary, just as they can start the main-loop at different times, nodes may join the candidacy-phase later than others. Moreover, unless a node has received a message from a neighbor, it has no knowledge whether other nodes have previously joined the main-loop or candidacy-phase. In fact, overcoming the absence of any such knowledge is one of the key challenges when designing algorithms for our model.

In more detail, the algorithm works as follows. A node starts executing the main loop upon waking up. At first, nodes wait for messages (on all channels) without sending themselves (lines 1-3). The reason is that nodes (re)starting the main-loop should not interfere with nodes currently competing in a candidacy-phase. The main part of the algorithm (starting in line 5) works in rounds, each of which contains $\gamma \cdot \lceil \log \hat{n} \rceil$ time-slots. A node becomes candidate (and starts executing the *start candidacy()* procedure) upon its first sending on channel Γ_1. Starting from a very small value, a node doubles its sending probability in each round, therefore increasing its chance to become candidate. As soon as it receives a message on Γ_1, however, it quits the current execution of the main-loop and restarts at line 1. In the analysis of the main-loop's properties in Subsection V-A, we give a bound on the number of nodes simultaneously being candidates. We will then go on to show that each time a restart occurs, some node in the 2-neighborhood will join the MIS within the required time-bounds. We call nodes in the waiting loop *inactive* and nodes in the main part of the algorithm *active*.

Having bounded the number of candidates, the candidacy-phase works as follows. In each time-slot, a candidate sends on Γ_2 with probability q. After sending the first time, a node becomes *excited* and starts increasing a counter in every time-slot. This counter is attached to each message. Upon receiving

a message on channel Γ_2 by another candidate, the receiver compares the sender's counter c' with its own. In case its own value is within $8 \log \hat{n}$ of the sender's counter, a node resets its own counter. This prevents two neighboring nodes from joining the MIS shortly in succession. It is interesting to notice that this method of comparing counters is sufficiently powerful to avoid long cascading chains of resettings. Once a node's counter reaches $\delta \cdot \lceil \log^3 \hat{n} / \log \log \hat{n} \rceil$, the node joins the MIS and immediately starts sending on channel Γ_3 with constant probability. Since no two nodes' counter reaches the threshold within $8 \log \hat{n}$ time-slots, there is sufficient time for the first MIS node to inform its neighbors, thus ensuring that no two neighbors join the MIS.

The algorithm's parameters q and β are defined as

$$q := \log \log \hat{n} / \log^2 \hat{n} \qquad \beta := 6.$$

Intuitively, the choice of q is motivated by two contradicting aims. On the one hand, q must be large enough such that some node will join the MIS within the desired runtime. On the other hand, a small q ensures that no two neighboring nodes join the MIS. In Subsection V-B, we prove that the choice of q results in *exactly one* node in each "neighborhood" joining the MIS. The parameter β is defined as to maximize the probability of a successful computation [18]. The parameters δ and γ can be used to tune the trade-off between running-time (small δ and γ) and probability of success (large δ and γ).

V. Analysis

This section contains the main theoretical contribution of this paper. We show that with high probability the algorithm computes a MIS in time $\mathrm{O}\big(\log^3 \hat{n} / \log \log n\big)$. Note that for the analysis, it is sufficient to assume $\hat{n} = n$, because solving MIS for $n' < n$ cannot be more difficult than for n. If it were, the imaginary adversary controlling the wake-up schedule of all nodes could simply decide to let $n - n'$ sleep infinitely long, which is indistinguishable from having n' nodes. Subsections V-A and V-B analyze the events on channels Γ_1 and Γ_2, respectively. The algorithm's runtime is derived in Subsection V-C. For the sake of clarity, we will sometimes omit the ceiling signs as imposed by the algorithm. Further, we assume n to be large enough, such that $\log^3 \hat{n} / \log \log n \geq 8 \log n$. A more rigorous analysis leads to the same results.

A. Main-Loop

In this subsection, the term *sum of sending probabilities* refers to channel Γ_1. We cover the plane with circles C_i of radius $r = 1/2$ in a hexagonal lattice, as shown in Figure 1. Let D_i be the circle centered at the center of C_i having radius $R = 3/2$. It can be seen in Figure 1. that D_i is (fully or partially) covering 19 smaller circles C_j. Note that every node in a circle C_i can hear all other nodes in C_i. Nodes outside D_i are not able to cause a collision in C_i.

Our analysis of the main loop builds on two lemmas obtained in [18]. In that paper, the authors propose an algorithm which computes an asymptotically optimal dominating set. In the following, we introduce both lemmas up to a level of detail necessary to understand our results. For details and the proofs, we refer to [18]. We first need the following definition:

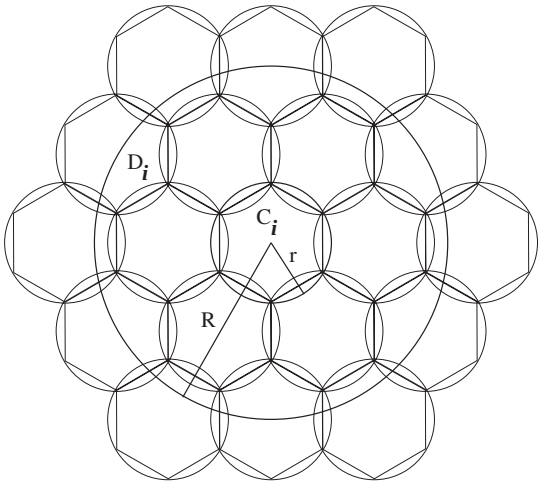

Fig. 1. Circles C_i and D_i

Definition 5.1: Consider a circle C_i. Let t be a time-slot in which a message is sent by a node $v \in C_i$ on channel Γ_1 and received (without collision) by all other nodes in C_i. We say that circle C_i *clears* itself in time-slot t.

The first key lemma given in [18] is a probabilistic bound on the sum of sending probabilities in a circle C_i. The idea is that once the sum of sending probabilities in C_i surpasses the (constant) threshold $1/2^\beta$, C_i will clear itself within the next $\gamma \cdot \lceil \log \hat{n} \rceil$ rounds with high probability. In particular, the probability is high enough as to ensure that the same property holds for all circles throughout the algorithm. This intuition is formalized in the following lemma, the proof of which is based on induction over all time-slots in which the sum of sending probabilities in an arbitrary C_i surpasses $1/2^\beta$.

Lemma 5.1: The sum of sending probabilities of nodes in a circle C_i is bounded by $\sum_{k \in C_i} p_k \leq 3/2^\beta$ with probability at least $1 - \mathrm{o}\big(\frac{1}{n^2}\big)$. The bound holds for all C_i in G with probability at least $1 - \mathrm{O}\big(\frac{1}{n}\big)$.

With Lemma 5.1, we can now bound the number of candidates in each circle C_i before a clearance. This is done in two steps. First, we compute the number of *collisions* in a circle before a clearance occurs. Secondly, Lemma 5.3 establishes an upper bound on the number of *new candidates per collision*. Consequently, combining both results leads to a bound on the number of candidates before a clearance and concludes our analysis of the main-loop.

We say that a *collision* in C_i occurs if more than one node in D_i is sending on Γ_1 in a particular time-slot.

Lemma 5.2: Let F be the number of collisions in a circle C_i between two subsequent clearances (or before the first clearance). For some constant $\tau \leq 8$, it holds $F < \tau \log n$ with probability at least $1 - \mathrm{o}\big(\frac{1}{n^2}\big)$.

It remains to establish a bound on the number of new candidates per collision.

Lemma 5.3: Let D be the number of nodes in C_i send-

ing in a time-slot. Given the occurrence of a collision, the expected number of sending nodes (i.e., new candidates) is $\mathrm{E}[D|D \geq 2] \in \mathrm{O}(1)$. Furthermore, $D \in \mathrm{O}(\log n/\log\log n)$ with probability $1 - \mathrm{o}\left(\frac{1}{n^2}\right)$.

Proof: Since the nodes send independently of each other, we can bound the conditional expectation as

$$E[D|D \geq 2] \quad \leq \quad 2 + E[D] \;=\; 2 + \sum_{k \in D_i} p_k$$
$$\underset{\text{Lemma 5.1}}{\leq} \quad 2 + 19 \cdot \frac{3}{2^\beta} \;\in\; \mathrm{O}(1)\,.$$

The high probability result can be derived using the upper tail Chernoff bound. Let $\mu = E[D|D \geq 2]$ and $\delta = \tau \log n/\log\log n$ for some constant τ. For P_+ defined as $\mathrm{P}[X > (1+\delta)\mu]$, it holds that

$$P_+ \quad < \quad \left(\frac{e^{-\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}$$

Taking the logarithm of P_+, this term simplifies to

$$\begin{aligned}
\log P_+ \quad &< \quad \mu\left(-\delta \cdot \log e - (1+\delta)\log(1+\delta)\right) \\
&\leq \quad -\frac{\mu\tau\log n}{\log\log n}\log\left(1 + \frac{\tau\log n}{\log\log n}\right) \\
&\leq \quad -\frac{\mu\tau\log n}{\log\log n}\left(\log(\tau\log n) - \log\log\log n\right) \\
&\leq \quad -\mu\tau\log n \cdot \left(1 - \frac{\log\log\log n}{\log\log n}\right) \\
&\leq \quad -2\log n
\end{aligned}$$

for large enough $\tau > \mu/2$. The lemma now follows from $P_+ < 2^{-2\log n} \leq n^{-2}$. □

Combining Lemmas 5.2 and 5.3 we have thus established an $\mathrm{O}\left(\log^2 n/\log\log n\right)$ upper bound on the number of candidates emerging in a circle C_i before its clearance.

B. Candidacy-Phase

Upon sending on Γ_1 in the main-loop of the algorithm, a node becomes candidate and competes for joining the MIS. In this subsection, we are going to show that each candidate will either join the MIS or will be covered by a MIS node within time $2\delta \cdot \log^3 n/\log\log n$. Based on the analysis of the previous subsection, we can bound the number of nodes simultaneously executing the *candidacy phase()* procedure. In particular, we know by Lemmas 5.2 and 5.3 that with high probability, there are at most $\tau\log^2 n/\log\log n$ candidates emerging in C_i *before a clearance*, for a constant τ. Further, all but the sending node restart the main loop after a clearance, and the sending node itself stops executing the main loop altogether. Due to the waiting loop at the beginning of the algorithm, no node in C_i is going to compete for becoming candidate during the next $2\delta \cdot \log^3 n/\log\log n$ time-slots. In other words, nodes *after a clearance* do not interfere with the current candidacy-phase due to their being inactive. The same holds for all $C_i \in D_i$ and hence, the number of candidates within the transmission range of a node v may not exceed $19\tau\log^2 n/\log\log n$. This crucial observation allows us to *separate* candidacy phases in a circle D_i and analyze them individually because a node's candidacy phase does not take longer than $2\delta \cdot \log^3 n/\log\log n$ time-slots, as shown in the sequel.

Lemma 5.4: Let t_m be the time-slot in which node v_m joins the MIS. The counter of all neighboring nodes v_c, $(v_m, v_c) \in E$, at time t_m is at most $c \leq 2\delta\log^3 n/\log\log n - 8\log n$ with high probability.

Proof: Let v_c be a neighboring node having counter $c > \delta\log^3 n/\log\log n - 8\log n$ by the time t_m. Assume for contradiction that v_c exists. By the definition of the algorithm, v_m must have sent in time-slot $t_m - \delta\log^3 n/\log\log n$ and v_c must have sent within the subsequent $8\log n$ time-slots. Afterwards, v_c has not received a message from v_m. If it had, it would have reset its counter. The probability $P_{recv}(t)$ that v_c receives a message from v_m in an arbitrary time-slot t is

$$P_{recv}(t) \geq \frac{\log\log n}{\log^2 n}\left(1 - \frac{\log\log n}{\log^2 n}\right)^{d(t)}$$

where $d(t)$ denotes the number of candidates within the transmission range of v_c at time t. We know that $d(t)$ is in the range between 1 and $19\tau\log^2 n/\log\log n$. $P_{recv}(t)$ is a monotonously decreasing function in $d(t)$ and therefore,

$$\begin{aligned}
P_{recv}(t) \quad &\geq \quad \frac{\log\log n}{\log^2 n}\left(1 - \frac{\log\log n}{\log^2 n}\right)^{\frac{19\tau\log^2 n}{\log\log n}} \\
&\in \quad \Omega(\log\log n/\log^2 n).
\end{aligned}$$

The probability that this event does not occur in any of the $\delta \cdot \lceil\log^3 \hat{n}/\log\log\hat{n}\rceil$ time-slots following t_m can be shown to be $n^{-\nu\delta}$, for some constant ν by applying Fact 3.1. By choosing δ accordingly, this probability can be made arbitrarily small. □

Let E_i denote the circle with radius $5/2$ centered at the center of C_i. Further, let t_c be the time-slot in which a node v_c becomes candidate in D_i. The following lemma shows that with high probability, at least one node in E_i (possibly more) joins the MIS within time $t_c + 2\delta\log^3 n/\log\log n$.

Lemma 5.5: For every candidate v_c, either v_c joins the MIS or a neighboring candidate v_c', $(v_c, v_c') \in E$, joins the MIS within time $t_c + 2\delta\log^3 n/\log\log n$ with high probability.

Proof: The main idea is that once a candidate v_c sends without collision at time t_s, it will either join the MIS at time $t_s + \delta\log^3 n/\log\log n$ or a neighboring candidate will join the MIS before. Let $c(v_c)$ be the value of v_c's counter at time t_s. As the message is sent without collision, all neighboring candidates v_c' having counter values in the interval $[v_c - \lceil 8\log n\rceil, \ldots, v_c + \lceil 8\log n\rceil]$ will set $c(v_c') := -\lceil 8\log n\rceil$ due to the received message on Γ_2. That is, only nodes with much larger counter values than $c(v)$ will keep on counting without resetting. Consequently, after time-slot t_s, it holds that

$$|c(v_c) - c(v_c')| > \lceil 8\log n\rceil \tag{1}$$

for all neighboring candidates v_c'. By sending the message at time t_s, v_c has become excited and hence, $c(v_c)$ is increased in each subsequent time-slot. By equation (1), no

neighboring candidate will be able to cause a reset of v_c's counter after t_s. In absence of any such neighbor with close enough counter, there is no way to prevent $c(v_c)$ from reaching $\delta \cdot \lceil \log^3 \hat{n} / \log\log \hat{n} \rceil$, which enables v_c to join the MIS.

It remains to be shown that with high probability, one candidate in D_i sends without collision in the interval $[t_c, \ldots, t_c + \delta \log^3 n / \log\log n]$, such that the above observation can conclude the proof. Let t be an arbitrary time-slot. Again, $d(t)$ denotes the number of candidates within the transmission range of the first candidate in D_i. The probability $P_{suc}(t)$ that one node sends without collision is given by

$$P_{suc}(t) = \frac{d(t)\log\log n}{\log^2 n}\left(1 - \frac{\log\log n}{\log^2 n}\right)^{d(t)-1}.$$

$P_{suc}(t)$ being a concave function in $d(t)$, we can focus our attention on the two border values $d(t) \geq 1$ and $d(t) \leq 19\tau \log^2 n / \log\log n$, for all $t_c \leq t \leq t_c + \delta \log^3 n / \log\log n$. For $d(t) = 1$, $P_{suc}(t)$ simplifies to

$$P_{suc}(t) = \frac{\log\log n}{\log^2 n}$$

while for $d(t) = 19\tau \log^2 n / \log\log n$, we have

$$\begin{aligned} P_{suc}(t) &= 19\tau \cdot \left(1 - \frac{\log\log n}{\log^2 n}\right)^{\frac{19\tau\log^2 n}{\log\log n}-1} \\ &\underset{\text{Fact 3.1}}{\geq} 19\tau e^{-19\tau}\left(1 - \frac{\log\log n}{\log^2 n}\right) \\ &\underset{n\geq 4}{\geq} 15\tau e^{-19\tau} \in \Omega(1). \end{aligned}$$

Putting things together, the probability of a successful time-slot is lower bounded by

$$P_{suc}(t) \geq \min\{\frac{\log\log n}{\log^2 n}, 15\tau e^{-19\tau}\}$$

throughout the considered time interval. The probability P_n that no candidate sends without collision in the interval $[t_c, \ldots, t_c + \delta \log^3 n / \log\log n]$ is therefore

$$\begin{aligned} P_n &\leq \left(1 - \min\left\{\frac{\log\log n}{\log^2 n}, 15\tau e^{-19\tau}\right\}\right)^{\frac{\delta\log^3 n}{\log\log n}} \\ &\leq \max\left\{n^{-\delta}, n^{-\frac{15\tau e^{-19\tau}\delta\log^2 n}{\log\log n}}\right\}. \end{aligned}$$

For large enough δ, this probability becomes arbitrarily small. Thus, with high probability, at least one node will send without collision within the first $\delta \log^3 n / \log\log n$ time-slots of the candidacy-phase. Since the same argument can be repeated for every node, the lemma follows from the observation stated at the beginning of the proof. □

We are now ready to state the main correctness theorem.

Theorem 5.6: With high probability, no two neighboring candidates join the MIS, i.e., the resulting independent set is correct.

Proof: Let v_m be a MIS node. Assume for contradiction that v'_m, $(v_m, v'_m) \in E$, is the first node violating the MIS condition. By Lemma 5.4, v'_m joins the MIS at least $8\log n$ time-slots after v_m. During these time-slots, v_m sends with constant probability $1/6$ on channel Γ_3. It is well-known that in a unit-disk graph, v'_m can have at most 6 independent neighbors (i.e., MIS nodes). The probability that v'_m has received no message by v_m can thus easily be shown to be $\overline{P_{recv}} \in \mathrm{O}(n^{-2})$. Observe that the same argument holds for nodes which area already covered by (up to 6) MIS nodes by the time of their wake-up. □

C. Running Time

Finally, we derive the algorithm's running time. In view of Lemma 5.5 and Theorem 5.6, every node will either join the MIS or become covered within time $\delta \log^3 n / \log\log n$ upon becoming candidate. We require the following simple observation which immediately follows from the algorithm's definition.

Lemma 5.7: Consider a circle C_i and let t_i be the time-slot in which the first node $v_c \in C_i$ executes line 5 of the main-loop. With high probability, there is a node in D_i which becomes candidate before time $t_i + \gamma \log^2 n$.

Proof: By the definition of the algorithm, v_c sends with $p_{v_c} = 2^{-\beta}$ on Γ_1 after $\log n$ rounds (unless v_c receives a message from a neighbor in which case the claim holds). The probability P_{no} that v_c does not send in any of this round's $\gamma \log n$ time-slots can be made arbitrarily small by choosing γ large enough, i.e.

$$P_{no} \leq \left(1 - \frac{1}{2^\beta}\right)^{\gamma\log n} \underset{\text{Fact 3.1}}{\leq} n^{-\gamma/2^\beta}.$$

□

We are now ready to prove the claimed running time of the algorithm.

Theorem 5.8: Every node $v \in G$ either joins the MIS or becomes covered by a neighboring node joining the MIS within time $\mathrm{O}(\log^3 n / \log\log n)$ upon waking up.

Proof: By Lemma 5.5, we know that if a node $w \in D_i$ becomes candidate at time t_w, it will be covered (possibly by joining the MIS itself) before $t_w + 2\delta \log^3 n / \log\log n$. This implies that there is a node $v_m \in E_i$ joining the MIS before $t_i + 2\delta \log^3 n / \log\log n$, where t_i is defined as the first time-slot a candidate emerges in D_i.

Consider an arbitrary node $v \in C_i$. By Lemma 5.7, we know that $2\delta \log^3 n / \log\log n + \gamma \log^2 n \in \mathrm{O}(\log^3 n / \log\log n)$ time-slots after its wake-up, v will either become candidate or there will be another candidate in D_i, from which v has received a message. In the first case, v will be covered within the next $2\delta \log^3 n / \log\log n$ time-slots by Lemma 5.5. In the latter case, at least one node in E_i joins the MIS within the same period. If this node covers v, we are done. If not, we know that the same conditions as above hold in the remaining, uncovered part of E_i, because the waiting period before the main-loop guarantees that a node cannot take part in the same candidacy-phase twice. The above argument can thus be repeated. Each time v either joins the MIS or becomes covered or one node in E_i joins the MIS in time $\mathrm{O}(\log^3 n / \log\log n)$.

By Theorem 5.6, no two neighboring nodes join the MIS. Hence, the number of different nodes joining the MIS in E_i is bounded by a constant because no more than a constant number of nodes with transmission range 1 can be packed in a circle E_i of radius $5/2$ such that no two nodes are within each other's mutual transmission range. In other words, at most a constant number of repetitions are required and it follows that node v is covered by a node in the MIS (possibly itself) within time $\mathrm{O}\big(\log^3 n/\log\log n\big)$ upon its wake-up. The same argument holds for every node $v \in G$ which concludes the proof. □

Our analysis is concluded by combining Theorems 5.6 and 5.8 in the following Corollary.

Corollary 5.9: With high probability, the algorithm computes a correct MIS such that each node is covered within time $\mathrm{O}\big(\log^3 n/\log\log n\big)$ upon waking up.

VI. Single-Channel

Realizing independent communication channels by means of an FDMA scheme may not always be desirable or possible. In this section, we show that a MIS clustering can be efficiently computed in the most basic *single-channel* setting, too. We do so by adapting a technique developed in [31]. Intuitively, the idea is to simulate each time-slot in the multi-channel model by a number of time-slots in the single-channel model. In particular, we show that the algorithm's time complexity remains polylogarithmic.

For the mapping to the single-channel case, assume that all senders sending on any channel in the multi-channel case send on a single common channel Γ. It is clear that this can lead to additional collisions. When simulating multiple channels with a single channel, we must guarantee that each successful transmission in the multi-channel case corresponds to a successful transmission in the single-channel case. The critical cases are those in which a node receives a message in the multi-channel case, but does not receive it in the single-channel case, due to a collision caused by the mapping. For instance, the situation when one node sends on Γ_1 and a collision occurs on Γ_2 and Γ_3 is a critical case, because the message on Γ_1 is not received on Γ when simulating the three communication channels with a single one. If, however, a collision occurs on all channels, it is not a critical case since no message is successfully transmitted in the multi-channel case. Our simulation algorithm must ensure that a message can be successfully transmitted in all critical cases.

In accordance to [31], let s and t be time-slots in the single-channel and multi-channel model, respectively. We simulate each time-slot t with $3\alpha\log^3 n/\log\log n$ time-slots s for a large enough constant α. The idea is that each node sending on Γ_1, Γ_2, or Γ_3 in t randomly sends on the single common channel Γ with probability $\log\log n/\log^2 n$ during the "middle" $\alpha\log^3 n/\log\log n$ time-slots (i.e., $t \in [\alpha\log^3 n/\log\log n \ldots 2\alpha\log^3 n/\log\log n - 1]$) corresponding to t. During the first and last $\alpha\log^3 n/\log\log n$ time-slots, such a node will not send. A node not sending on any channel in t remains quiet during the entire interval. This approach follows the intuition that $\alpha\log^3 n/\log\log n$ time-slots suffice to "spread" the (at most) $\mathrm{O}\big(\log^2 n/\log\log n\big)$ senders on each channel in time, enabling each one of them to send without collision at least once.

Formally, the simulation algorithm is defined as follows. We write $send(t) = 1$ if a sender sends in time-slot t and $send(t) = 0$, otherwise. Further, let $\lambda := \alpha\log^3 n/\log\log n$ and $p := \log\log n/\log^2 n$. Each node simulates time-slot t by 3λ single-channel time-slots $s_1 \ldots s_{3\lambda}$ in the following way:

$$
\begin{aligned}
send(t) = 0 \Rightarrow\ & \forall s_i \in [s_1 \ldots s_{3\lambda}] : \\
& send(s_i) := 0 \\
send(t) = 1 \Rightarrow\ & \forall s_i \in [s_1 \ldots s_\lambda\ ,\ s_{2\lambda+1} \ldots s_{3\lambda}] : \\
& send(s_i) := 0 \\
send(t) = 1 \Rightarrow\ & \forall s_i \in [s_{\lambda+1} \ldots s_{2\lambda}] : \\
& send(s_i) := \begin{cases} 1, \text{ with prob. } p \\ 0, \text{ with prob. } 1-p \end{cases}
\end{aligned}
$$

Let $suc(t) = 1$ denote that a message has been successfully transmitted in time-slot t, and $suc(t) = 0$ otherwise. We can state the following lemma, the proof of which is similar to the one given in [31] and is omitted due to space limitations.

Lemma 6.1: Time-slot t can be simulated by $\mathrm{O}\big(\log^3 n/\log\log n\big)$ time-slots s_i, $i \in [1 \ldots 3\alpha\log^3 n/\log\log n]$, for a large enough constant α such that $suc(t) = 1 \Leftrightarrow \exists i : suc(s_i) = 1$ with probability $1 - \mathrm{O}\big(1/n^{2\alpha}\big)$.

Note that due to asynchronous wake-up, we cannot and do not assume that intervals of length 3λ are aligned among nodes. Thanks to the "buffer-periods" at the beginning and end of each interval, we do not rely on such an assumption. Having Lemma 6.1, it is now straightforward to correctly simulate the entire algorithm with a single channel within polylogarithmic running time: All nodes simulate each of their time-slots with the algorithm given above, leading to the following Theorem.

Theorem 6.2: The MIS algorithm in the single-channel model has time-complexity $\mathrm{O}(polylog(n))$. With high probability, all critical steps are executed like in the multi-channel algorithm.

Proof: Time-complexity follows from Theorem 5.8 and Lemma 6.1. For correctness, we compute the probability P that all critical steps are correctly simulated. Since the algorithm's execution takes at most $C \cdot n\log^3 n/\log\log n$ steps for a constant C in the multi-channel case, P is

$$
P \ \geq\ \left(1 - \frac{1}{n^{2\alpha}}\right)^{\frac{Cn\log^3 n}{\log\log n}} \ \in\ 1 - \mathrm{O}\left(\frac{\log^3 n}{n^\alpha}\right).
$$

□

VII. Conclusions

How can we structure the chaos existing during the deployment of an ad-hoc or sensor network? In this paper, we have tried to provide an answer by analyzing the initialization phase of unstructured multi-hop radio networks. Immediately after deployment, organizing an efficient medium access scheme is

probably the most urgent task at hand, and computing a good initial clustering is one of the key ingredients to solving it.

We have proposed a novel algorithm which computes a maximal independent set in polylogarithmic time even under a model featuring many of the realities of unstructured networks. Besides being a dominating set of excellent quality, a MIS has the additional property that no two clusterheads interfere. This is particularly desirable in the initialization phase of ad-hoc and sensor networks, facilitating the construction of an efficient MAC layer.

We believe that due to its being fast and simple, our algorithm has practical relevance in a variety of scenarios, particularly in newly deployed ad-hoc and sensor networks. Analyzing important issues such as energy-efficiency in the *unstructured radio network* model is an interesting and promising field for future research.

ACKNOWLEDGEMENTS

We would like to thank Rohan Fernandes and Miguel Mosteiro for various helpful comments and in particular, for pointing out an improvement of Lemma 5.5. We would also like to thank Fabian Kuhn for valuable discussions.

REFERENCES

[1] D. J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Transactions on Communications*, vol. COM-29, no. 11, pp. 1694–1701, 1981.

[2] S. Basagni, "Distributed Clustering for Ad Hoc Networks," in *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, pp. 310–315.

[3] M. Chatterjee, S. K. Das, and D. Turgut, "An On-Demand Weighted Clustering Algorithm (WCA) for Ad-Hoc Networks," in *Proceedings of IEEE GLOBECOM 2000*. ACM Press, 2000, pp. 1697–1701.

[4] M. Gerla and J. Tsai, "Multicluster, mobile, multimedia radio network," *ACM/Baltzer Journal of Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

[5] C. R. Lin and M. Gerla, "Adaptive Clustering in Mobile Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 1265–1275, 1997.

[6] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 12.

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *Proceedings of the 33^{rd} Annual Hawaii International Conference on System Sciences*, 2000, pp. 3005–3014.

[8] K. Alzoubi, P.-J. Wan, and O. Frieder, "Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks," in *Proceedings of the 3^{rd} ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, EPFL Lausanne, Switzerland, 2002, pp. 157–164.

[9] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Discrete Mobile Centers," in *Proceedings of the 17^{th} annual symposium on Computational geometry (SCG)*. ACM Press, 2001, pp. 188–196.

[10] L. Jia, R. Rajaraman, and R. Suel, "An Efficient Distributed Algorithm for Constructing Small Dominating Sets," in *Proceedings of the 20^{th} ACM Symposium on Principles of Distributed Computing (PODC)*, 2001, pp. 33–42.

[11] F. Kuhn and R. Wattenhofer, "Constant-Time Distributed Dominating Set Approximation," in *In Proceedings of 22^{nd} ACM Int. Symposium on the Principles of Distributed Computing (PODC)*, 2003, pp. 25–32.

[12] P. Wan, K. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proceedings of Infocom 2002*, 2002.

[13] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," in *Proc. of the 3^{rd} Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM)*, 1999, pp. 7–14.

[14] S. Basagni, "A Distributed Algorithm for finding a Maximal Weighted Independent Set in Wireless Networks," in *Proceedings of the 11^{th} IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*, 1999, pp. 517–522.

[15] K. Alzoubi, P.-J. Wan, and O. Frieder, "Maximal Independent Set, Weakly-Connected Dominating Set, and Induced Spanners in Wireless Ad Hoc Networks," *International Journal of Foundations of Computer Science*, vol. 14, no. 2, pp. 287–303, 2003.

[16] M.Cardei, X.Cheng, X.Cheng, and D. Z. Du, "Connected Domination in Ad Hoc Wireless Networks," in *Proceedings of the 6^{th} International Conference on Computer Science and Informatics*, 2002.

[17] Y. Wang and X.-Y. Li, "Geometric Spanners for Wireless Ad Hoc Networks," in *Proceedings of the 22^{nd} International Conference on Distributed Computing Systems (ICDCS)*, 2002.

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

[19] R. Bar-Yehuda, O. Goldreich, and A. Itai, "Efficient Emulation of Single-Hop Radio Networks with Collision Detection on Multi-Hop Radio Networks with no Collision Detection," *Distributed Computing*, vol. 5, pp. 67–71, 1991.

[20] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple Access and the Busy Tone Solution," vol. COM-23, no. 12, pp. 1417–1433, 1975.

[21] R. M. Karp and A. Widgerson, "A fast Parallel Algorithm for the Maximal Independent Set Problem," in *Proceedings of the 16^{th} Annual ACM Symposium on Theory of Computing (STOC)*, 1984, pp. 266–272.

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

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

[24] 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)*. ACM Press, 1987, pp. 98–108.

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

[26] T. Hayashi, K. Nakano, and S. Olariu, "Randomized Initialization Protocols for Packet Radio Networks," in *Proceedings of the 13^{th} International Parallel Processing Symposium (IPPS)*, 1999, pp. 544–548.

[27] K. Nakano and S. Olariu, "Energy-Efficient Initialization Protocols for Single-Hop Radio Networks with no Collision Detection," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, 2000.

[28] 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)*. ACM Press, 2000, pp. 113–121.

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

[30] 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)*. ACM Press, 2004, pp. 266–274.

[31] 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.

[32] L. G. Roberts, "Aloha Packet System with and without Slots and Capture," *ACM SIGCOMM, Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.