Maximal Independent Sets in Multichannel Radio Networks

Sebastian Daum
University of Freiburg
Freiburg, Germany
sdaum@cs.uni-freiburg.de

Mohsen Ghaffari
MIT
Cambridge, MA, USA
ghaffari@csail.mit.edu

Seth Gilbert
National Univ. of Singapore
Singapore, Singapore
seth.gilbert@comp.nus.edu.sg

Fabian Kuhn
University of Freiburg
Freiburg, Germany
kuhn@cs.uni-freiburg.de

Calvin Newport
Georgetown University
Washington D.C., USA
cnewport@cs.georgetown.edu

ABSTRACT

We present new upper bounds for fundamental problems in multichannel wireless networks. These bounds address the benefits of dynamic spectrum access, i.e., to what extent multiple communication channels can be used to improve performance. In more detail, we study a multichannel generalization of the standard graph-based wireless model without collision detection, and assume the network topology satisfies polynomially bounded independence.

Our core technical result is an algorithm that constructs a maximal independent set (MIS) in $O\big(\frac{\log^2 n}{\mathcal{F}}\big) + \tilde{O}(\log n)$ rounds, in networks of size n with $\mathcal{F}$ channels, where the $\tilde{O}$-notation hides polynomial factors in $\log \log n$.

Moreover, we use this MIS algorithm as a subroutine to build a constant-degree connected dominating set in the same asymptotic time. Leveraging this structure, we are able to solve global broadcast and leader election within $O\big(D + \frac{\log^2 n}{\mathcal{F}}\big) + \tilde{O}(\log n)$ rounds, where D is the diameter of the graph, and k-message multi-message broadcast in $O\big(D + k + \frac{\log^2 n}{\mathcal{F}}\big) + \tilde{O}(\log n)$ rounds for unrestricted message size (with a slow down of only a log factor on the k term under the assumption of restricted message size).

In all five cases above, we prove: (a) our results hold with high probability (i.e., at least $1 - 1/n$); (b) our results are within $\mathrm{poly}(\log \log n)$-factors of the relevant lower bounds for multichannel networks; and (c) our results beat the relevant lower bounds for single channel networks. These new (near) optimal algorithms significantly expand the number of problems now known to be solvable faster in multichannel versus single channel wireless networks.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems—*computations on discrete structures*

General Terms

Algorithms, Theory

Keywords

wireless networks, shared spectrum, multichannel, maximal independent set, connected dominating set

1. INTRODUCTION

Modern wireless devices rarely operate on a fixed communication channel. It is more common for them to use a wide swath of spectrum that has been subdivided into multiple independent channels (e.g., [1, 6]). This reality inspires a compelling question: When and how can we leverage the availability of multiple channels to improve the performance of wireless algorithms?

One might hope that using $\mathcal{F}$ channels you can always achieve a $\mathcal{F}$-times speed-up. For distributed algorithms, however, this goal is complicated by two factors: (a) each device can only use a single channel at a time; and (b) the size and density of the network is often unknown *a priori*. (In fact, some well-known problems, such as multihop wake-up, provably derive *no* benefit from multiple channels [11].) In this paper, we overcome these challenges to significantly increase the corpus of algorithms known to solve problems faster in multichannel versus single channel wireless networks. In more detail, we prove new randomized upper bounds for the following fundamental problems in graphs satisfying polynomial bounded independence (defined below): (i) establishing a Maximal Independent Set

(MIS); (ii) establishing a constant-degree Connected Dominating Set (CDS); (iii) broadcasting one message—or a set of messages—to every device in a network; and (iv) electing a leader in a network. For each of these problems, we give solutions that are within poly-log log n factors of optimal in the multichannel setting, and that are faster than their corresponding lower bounds in single channel networks.

We argue that these results provide a powerful argument for wireless algorithm designers to more aggressively embrace the availability of multiple channels to gain performance.

Results. We assume a multichannel generalization of the standard graph-based wireless model [4, 8]. In each round, each node can choose a single channel to participate on from among $\mathcal{F} \geq 1$ available channels. We further assume that the graph representing our network topology satisfies polynomial bounded independence (the independence number of a radius r neighborhood is bounded by $f(r)$ for some polynomial f) [22, 26]. This assumption generalizes a variety of attempts to model the topology of wireless networks, including the widely used unit-disk graphs, quasi-unit-disk graphs, or, more generally, unit-ball graphs where the underlying metric has bounded doubling dimension [26].

The primary technical result of the paper is an algorithm that constructs an MIS in $O\left(\frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds—where $\tilde{O}$ hides polynomial factors in log log n—with high probability[1]. This algorithm consists of two main pieces: a "decay filter" that reduces the number of nodes competing in each "area" to $O(\text{poly}(\log n))$, and a "herald filter" that leverages multiple channels to efficiently further reduce the nodes down to a constant number per area.

Much of the complexity resides in the Herald Filter, where we reduce the number of contenders to join the MIS from poly($\log n$) to $O(1)$. Part of the complexity comes from asynchrony: new arrivals and neighboring regions can force existing nodes to "restart," preventing progress toward the MIS. Part of the complexity comes from the fact that randomized symmetry breaking works well over large populations, but less predictably as the number of participants gets small.

To put this result in context, in the single channel model, building an MIS requires $\Theta(\log^2 n)$ time [11, 18, 20, 24]. Based on the lower bound techniques developed in [10, 11, 13, 20], we show in [9] that in bounded independence graphs (and even in unit-disk graphs) any MIS algorithm requires at least $\Omega\left(\frac{\log^2 n}{\mathcal{F}} + \log n\right)$ rounds in a network with $\mathcal{F}$ channels. Our algorithm matches this multichannel lower bound up to poly($\log \log n$) factors and beats the single channel lower bound. The lower bound also implies that even if the number of channels is arbitrarily large, solving the MIS problem still requires at least $\Omega(\log n)$ rounds, and our upper bound achieves almost the same time with just $\Theta(\log n)$ many channels.

Having developed an MIS algorithm, we use it as a subroutine to build a constant-degree CDS, with high probability, also in $O\left(\frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds. The key challenge here is to efficiently—i.e., in $o(\log^2 n)$ time—identify short paths between nearby MIS nodes, even while the MIS subroutine is ongoing. We then leverage the overlay provided by our CDS algorithm to solve global broadcast and leader election (with synchronous starts) in $O\left(D + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds, and k-message multi-message broadcast in $O\left(D + k + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds for unrestricted message size (with a slow down of only a log factor on the k term under the assumption of restricted message size). These bounds (nearly) match the relevant $\Omega\left(D + \frac{\log^2 n}{\mathcal{F}}\right)$ bound for multichannel networks [16], and beat the relevant $\Omega(D + \log^2 n)$ lower bound for single channel networks [3].

Related Work. There has been much research on algorithms for graph-based *single channel* wireless network models, beginning with Chlamtac and Kutten [8] in the centralized setting and Bar-Yehuda et al. [4] in the distributed setting. The problem of finding an MIS in a distributed fashion has been studied extensively for a standard message passing model (e.g., without collisions). On general network topologies, an MIS can be built in $O\left(\min\left\{\log n, \log \Delta \sqrt{\log n}\right\}\right)$, where Δ is the largest degree of the network graph [2, 5, 23]. For bounded independence graphs, this is improved to $O(\log^* n)$ [27]. For single-channel radio networks satisfying the unit disk graph property, it has been shown that $O(\log^2 n)$ rounds is sufficient [24]. Using a reduction from the single-hop wake-up problem, this bound was shown tight [11, 18, 20, 24].

To our knowledge, the use of a connected dominating set (CDS) as a wireless network backbone was first described in [17]. It is well-known (and already described in [17] for the case of unit disk graphs) that a CDS can be constructed by first computing a small dominating set (in the case of bounded independence graphs, an MIS provides such a small dominating set), and then connecting the nodes of the dominating set through 2 and 3 hop paths. In a bounded independence graph, connecting all MIS nodes at distance at most 3 by a short path leads to a CDS where the graph induced by the CDS has bounded degree. The MIS algorithm of [24] combined with the CDS algorithm of [7] (which assumes an MIS as a precondition) provides a constant-degree CDS in $O(\log^2 n)$ rounds in the radio network model with synchronous starts (i.e., where all nodes start during the same round).[2]

The study of algorithms for multichannel wireless networks is more recent. Initially, much of the focus in multichannel networks was providing increased fault-tolerance: even if some of the channels were faulty, the computation would proceed. This basic model of unreliable multichannel wireless communication, often called *t-disrupted*, was introduced in [14], and has since been well-studied; e.g., [12–15, 19, 25, 28, 29].

We previously tackled the problem of leader election in single-hop networks (i.e., the diameter is 1) [10], where we solved the problem in $O\left(\frac{\log^2 n}{\mathcal{F}} + \log n\right)$ rounds. These techniques did not directly translate to multihop networks. We also have studied the problem of broadcast in multihop networks [16]. In this case, we assumed that nodes had access to collision detection, showing how to leverage this information to solve broadcast in $O\left((D + \log n)\left(\log \mathcal{F} + \frac{\log n}{\mathcal{F}}\right)\right)$. For $\mathcal{F} = \log n$, this yields results similar to this paper, i.e., $O(D) + \tilde{O}(\log n)$. The results are hard to compare, however,

---

[1]We use the phrase *high probability* to indicate a probability at least $1 - \frac{1}{n^c}$, for some arbitrary constant $c \geq 1$.

---

[2]The MIS result of [24] does not require the synchronous start property, but the CDS piece from [7] does.

as [16] assumes collision detection (which we do not), but we assume bounded independence (which [16] does not).

Finally, we have studied the problem of wake-up and approximating a minimum dominating set (MDS) in a multi-hop network with a topology that satisfies a clique decomposition assumption [11]. For the MDS problem, we achieved a constant-factor approximation of an MDS, in expectation, requiring $O\big(\frac{\log^2 n}{\mathcal{F}}\big) + \tilde{O}(\log n)$ rounds. We found the technique could not easily be extended to achieve the strict independence of an MIS (with high probability) or tolerate the more general bounded independence assumption (instead of a clique decomposition assumption).

2. PRELIMINARIES

Radio Network Model. We consider a multichannel variant of the standard graph-based radio network model [4]. The network is modeled as an n-node graph $G = (V, E)$. Each node knows n or a polynomial upper bound on n. There are $\mathcal{F}$ communication channels. Time is divided into synchronized slots, i.e., rounds. For the purpose of analysis, we imagine a global round numbering, but nodes do not have access to this global time. In each round, each node can choose one of the $\mathcal{F}$ channels to operate on; it can either transmit or listen on the channel. A node u that listens on a channel C receives a message from a neighbor v if and only if node v is transmitting on C and no other neighbor of u is transmitting on C. If two or more neighbors of u transmit on C, or if no neighbor of u transmits on C, then u receives silence. That is, we assume there is no collision detection available. A node that transmits does not receive anything. Notably, a node that operates on channel C in a given round learns nothing about events on channels other than C in that round.

Notation. For a subset of nodes $S \subseteq G$, we use $N_G^d(S)$ to denote the set $\{u \mid \exists v \in S, \; dist_G(u, v) \leq d\}$, where $dist_G(u, v)$ is the shortest distance between u and v in graph G. When $|S| = 1$, e.g., $S = \{v\}$, we use $N_G^d(v)$ to mean $N_G^d(\{v\})$. We use $N_G(v)$ to denote the neighbors of v, i.e., $N_G(v) = N_G^1(v) \backslash \{v\}$. When clear from the context, we omit the subscript G. In later sections, we describe algorithms in which nodes can be in various states, e.g.: $\mathbb{A}$, $\mathbb{H}'$, $\mathbb{H}$, $\mathbb{L}'$, $\mathbb{L}$, $\mathbb{M}$, $\mathbb{E}$. Where appropriate, we slightly abuse notation and use the state names to denote the set of nodes in a given state, e.g, $\mathbb{A}$ to denote the set $\{v \in V : v$ is in state $\mathbb{A}\}$. We sometimes study $N^d(u) \cap \mathbb{A}$ and write $N_\mathbb{A}^d(u)$. When referring to a local variable X of a node u, we write $X(u)$. If the round number is not clear from the context, we denote $X(u)$ in round r as $X(u, r)$.

Bounded Independence. We assume that the network graph G is a bounded independence graph as introduced and described in [22, 26]. Formally, any independent set $S \subseteq N_G^d(v)$ for any node v has size at most $\alpha(d)$, where $\alpha(d)$ is a polynomial function in d and (in particular) independent of n. Hence, any subgraph induced by a subset of a neighborhood $N_G^d(v)$ for $d = O(1)$ has only constant size independent sets.

Probability Notation. Consider an event A, a constant c, and a variable k. If $\mathbf{P}(A) \geq 1 - e^{-ck}$, then we say that A happens with very high probability with regard to k (w.v.h.p.(k)). If $\mathbf{P}(A) \geq 1 - k^{-c}$, then we say A happens with high probability with regard to k (w.h.p.(k)), and if A happens w.h.p.(n), then we simply say A happens with

high probability (w.h.p.). Finally, w.c.p. abbreviates *'with constant probability'*.

Number of Channels. We assume $\omega(1)$ channels are available; otherwise there are existing algorithms that solve the problem in the same asymptotic time frame. If $\omega(\log n)$ channels are available, we restrict the usage to $\Theta(\log n)$, as there is no benefit from using more—in [9] we show that computing an MIS requires $\Omega\big(\frac{\log^2 n}{\mathcal{F}} + \log n\big)$ rounds. Solely for ease of exposition, we assume a minimum number of $\Omega(\log \log n)$ channels for all descriptions and proofs in this paper; this is not a requirement for the algorithm to work. We explain in [9] how to adapt our algorithms to work in a setting with $o(\log \log n)$ channels.

3. PROBLEM STATEMENT

We study randomized algorithms algorithms for the following problems, with high probability:

Maximal Independent Set. We say that an algorithm solves MIS in time T, if the following three properties hold: *(P1)* Each node v that wakes up in round r declares itself as either *dominating* or *dominated* by round $r' \in [r, r + T]$ and this decision is permanent. *(P2)* For each round r and node v, if v is *dominated* in round r, then v has at least one *dominating* neighbor in that round. *(P3)* For each round r and node v, if v is *dominating* in round r, then v does not have any neighboring *dominating* node in that round.

Connected Dominating Set. We say that an algorithm solves (constant-degree) CDS in time T, if the following four properties hold: *(P1)* Each node v that wakes up in round r declares itself as either *dominating* or *dominated* by round $r' \in [r, r + T]$ and this decision is permanent. *(P2)* For each round r and node v, if v is *dominated* in round r, then v has at least one *dominating* neighbor in round r. *(P3)* For each round r and node v, if v is *dominating* in round r, then v has at most $O(1)$ *dominating* neighbors in that round. *(P4)* For each round r and each *connected component* C in the graph induced by nodes awake in round $r - T$, the *dominating* nodes in C form a connected subgraph within C.

Other Problems. We also consider **global broadcast**, where a node starts with a message, and **multi-message broadcast**, where k nodes start with a message; in both cases the algorithm succeeds when every node in the network has received the message(s). Finally, we consider leader election, which terminates when exactly one node has declared itself the leader (and no future nodes declare themselves the leader).

4. OVERVIEW OF MIS ALGORITHM

In this section, we provide an overview of the MIS algorithm.

Algorithm Outline. Our algorithm consists of two main building blocks: the *decay filter* and the *herald filter*. The decay filter is used to reduce the maximum degree of the communication graph to a poly-logarithmic value. The herald filter assumes that the maximum degree is bounded to a poly-logarithmic value and establishes an MIS in this setting.

The flow of the algorithm is as follows. Each node, on activation, starts in the decay filter. As time passes, some of the nodes move from the decay filter to the herald filter. Nodes exit the herald filter when either they have joined the

MIS and have status *dominating*, or when they have an MIS neighbor and are thus *dominated*. In order to analyze the time complexity of our algorithm, we bound the time each node spends in each of the filters.

We note that nodes do not move backward in this flow. The dominating and dominated statuses are permanent; a node that is in the herald filter does not go back to the decay filter. However, a node u that is in the decay filter might skip the herald filter and directly become *dominated* if u receives a message from a *dominating* neighbor node v. Also, a node that has made progress in one of the filters can be forced to restart at the beginning of its current filter.

A node halts as soon as it discovers that it is dominated. On the other hand, a dominating node v cannot halt: it continues transmitting its status to its neighbors every so often, ensuring that each neighbor w that awakes at a later time becomes *dominated*.

Block Guarantees. We now present the guarantees of each block. We later discuss how the blocks are implemented and prove the specified guarantees. The first property holds for all components of the algorithm, and acts in parallel with the other block guarantees. It plays an important role in combining the blocks.

(G1) For each node u, if u is awake in round r and it has a *dominating* neighbor v in that round, then w.h.p. node u becomes *dominated* by round $r' = r + O(\log n)$.

Implementation is straightforward: each node u that does not have its final status listens to one of a constant number of channels, w.c.p., every $O(1)$ rounds. Each node that is *dominating* periodically transmits on those channels, w.c.p., every $O(1)$ rounds. If u receives a message from a *dominating* neighbor, then u becomes *dominated*. Since each node can have at most $\alpha(1)$ *MIS* neighbors, applying Chernoff bound gives us guarantee (G1). We show later that both filters satisfy this guarantee.

The guarantees that we get from the decay filter are as follows:

(G2) W.h.p., for each node v and each round r, at most $O(\log n)$ nodes in $N_G^1(v)$ exit the decay filter in round r to enter the herald filter. Each node v that enters the herald filter has spent $\Omega(\log n)$ rounds in the decay filter, long enough to become dominated if v already had a dominating neighbor after waking up.

(G3) W.h.p., for each node v that is in the decay filter in round r, by round $r' = r + O\left(\frac{\log^2 n}{\mathcal{F}} + \log n\right)$, either v is *dominated*, in which case it has a dominating neighbor, or at least one node in $N_G^1(v)$ exits the decay filter and enters the herald filter between rounds r and r'.

The guarantees that we get from herald filter are as follows:

(G4) W.h.p., for each node v that is in the herald filter in round r, by round $r' = r + \tilde{O}(\log n)$, v is *dominating* or *dominated*. In the latter case, v has a dominating neighbor.

(G5) W.h.p., in any round r, the set of dominating nodes is an independent set.

Note that (G2) and (G4) together provide that, w.h.p., the maximum degree in the graph induced by undecided nodes in the herald filter is bounded by some $\Delta_H = O(\text{polylog } n)$.

We will describe the algorithm and prove guarantees (G1)–(G5) in the following two sections. Before doing so, we state our main theorem (for a detailed proof we refer to [9]).

THEOREM 4.1. *W.h.p., an algorithm satisfying (G1)–(G5) solves the MIS problem in time* $O\left(\frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$.

PROOF SKETCH. Property (G5) implies that the set of dominating nodes is independent. It remains only to show that within $O(\log^2(n)/\mathcal{F}) + \tilde{O}(\log n)$ time, every node is either dominating or dominated. We claim that for each node v that is awake in round r, there is a round $r' = r + O(\log^2(n)/\mathcal{F}) + \tilde{O}(\log n)$ such that, by the end of round r', either v is dominating or dominated, or at least one 'new' node $w \in N_G^2(v)$ that was not dominating in round r has become *dominating*. Since the MIS can contain at most $O(1)$ dominating nodes in $N_G^2(v)$, the result follows from this claim. The claim follows from the fact that, by (G3), within $O(\log^2(n)/\mathcal{F}) + \tilde{O}(\log n)$ rounds, either v is dominated or it has a neighbor u that has entered the herald filter; by (G4), within $\tilde{O}(\log n)$ rounds, either u itself or a neighbor of u is dominating. $\square$

5. DECAY FILTER

The decay filter is a slightly modified version of the active state of the active wake-up algorithm in [11]. In essence, the decay filter is a backoff style protocol in which nodes broadcast with exponentially increasing probabilities; whenever a node receives a message from another node, it is knocked out and restarts the filter. To reduce contention, the broadcasts are distributed uniformly over the $\mathcal{F}$ channels.

Algorithm description. For pseudo-code and detailed proofs we refer to [9]. The decay filter uses $\Theta(\mathcal{F})$ channels, divided into two sets: (i) the *decay channels* $\mathcal{D}_1, \ldots, \mathcal{D}_F$, where $F = \Theta(\mathcal{F})$; and (ii) the *report channels* $\mathcal{R}_1, \ldots, \mathcal{R}_{3\alpha(1)}$.

A node v in the decay filter proceeds as follows. First, v spends $\Theta(\log n)$ rounds listening to one of the report channels, chosen at random in each round. If it hears from an MIS node, it halts and becomes dominated.

Otherwise, node v proceeds through $\log n$ phases. In these phases, the nodes use a subset of $F = \Theta(\mathcal{F})$ out of the total $\mathcal{F}$ channels, the "decay channels." Each phase consists of $\Theta\left(\frac{\log n}{F}\right) = \Theta\left(\frac{\log n}{\mathcal{F}}\right)$ rounds, except for the last phase, which consists of $\Theta(\log n)$ rounds.

In each round of each phase, each node listens to one of the report channels with probability $1/2$. If node u is not listening to a report channel and it is in phase j, then u chooses uniformly at random one of the $F = \Theta(\mathcal{F})$ channels $\mathcal{D}_1, \ldots, \mathcal{D}_F$. Then, with probability $\frac{2^j}{4n}$, u transmits on this selected channel and otherwise u listens to this selected channel. Thus, transmission probabilities are exponentially increasing over the phases, going from $\frac{1}{2n}$ to $\frac{1}{4}$.

If a node u transmits in a round, then u immediately exits the decay filter. Moreover, if a node u receives a message on some channel $\mathcal{D}_m$, then u gets knocked out and it restarts the decay filter. If u passes through all the phases without ever transmitting, then u moves to the herald filter.

Analysis. The main difference between the decay filter here and the version in [11] is that the graph model of the present paper is more general. Based on the following weighted version of Turán's theorem it is possible to generalize the analysis from bounded-degree clique partition assumption to general bounded independence graphs.

LEMMA 5.1. *Let $G = (V, E)$ be a graph with independence number $\alpha(G)$ and assume that every node $u \in V$ has a positive edge weight $w_u > 0$. Define $W := \sum_{v \in V} w_v$ and for each $u \in V$, $W_u := \sum_{v \in N_G^+(u)} w_v$. Further, let $V_{heavy} \subseteq V$ be the set of nodes v for which $W_v \geq \frac{W}{2\alpha(G)}$. Then,*

$$\sum_{v \in V} \frac{w_v}{W_v} \leq \alpha(G), \qquad \sum_{v \in V} w_v \cdot W_v \geq \frac{W^2}{\alpha(G)}$$

$$and \qquad \sum_{v \in V_{heavy}} w_v > \frac{W}{2\alpha(G)}.$$

LEMMA 5.2. *W.h.p., for all rounds $r \geq 1$ and for all nodes $u \in V$, we have $P_u(r) = O(\mathrm{F}) = O(\log n)$.*

PROOF SKETCH. For a round r, let $p_u(r)$ be the transmission probability of node u in round r. Moreover, let $P_u(r) := \sum_{v \in N^1(u)} p_v(r)$. For the sake of contradiction, suppose that $P_u(r)$ exceeds threshold $3c\mathrm{F}$ for the first time in round r. It is easy to see that in all rounds $r' \geq r - T$, $P_u(r') \in [c\mathrm{F}, 3c\mathrm{F}]$, where $T = \Theta(\log n/\mathrm{F})$ is the length of one phase. This is because, in each phase the transmission probability of each node can at most double, and the newly awakened nodes can contribute at most $c\mathrm{F}$. In each such round r', w.v.h.p.(F), for $\Theta(\mathrm{F})$ channels, each of these channels are chosen by a set of nodes from $N^1(u)$ that contribute a probability mass that is on the order of $\Theta(c)$. Similarly, we see that on at least half of these channels, there is no parge probability mass within distance 2. Thus, for each such channel, w.c.p., a single node $v \in N^1(u)$ transmits and no node in $N^2(u)$ interferes. Lemma 5.1 says that the transmitting node itself has a constant chance to be part of a set $S \subseteq N^1(u)$ of nodes, which together contribute a constant fraction to the probability mass on that channel, in which case a constant fraction of the probability mass in $N^1(u)$ on that channel is knocked out. This happens w.c.p. on each of the aforementioned channels, and w.v.h.p.(F) there are $\Theta(\mathrm{F})$ of them. Thus, assuming that the constants in $T = \Theta(\log n/\mathrm{F})$ are sufficiently large, we get that, w.h.p., the probability mass is reduced by at least $1/3$ over the course of T rounds, contradicting our initial assumption. □

It is then easy to show that property (G2) holds:

LEMMA 5.3. *(G2): With high probability, for each node v and each round r, at most $O(\log n)$ nodes in $N_G^1(v)$ come out of the decay filter in round r to enter the herald filter. Each node that enters herald filter has spent $\Omega(\log n)$ rounds in decay filter.*

Property (G3) follows from the structure of the protocol:

LEMMA 5.4. *(G3): W.h.p., for each node u that is in decay filter in round r, by round $r' = r + O\left(\frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$, either u is* dominated, *in which case it has a dominating neighbor, or at least one node in $N^1(u)$ gets out of decay filter and enters herald filter.*

6. HERALD FILTER ALGORITHM

In this section, we present the herald filter. Detailed pseudo-code can be found in [9]. Recall that, to simplify explanations and ease understanding, we assume $\Omega(\log \log n)$ channels to be available.

The herald filter assumes that in the subgraph induced on the nodes in the filter the degree of each node is always bounded by $\Delta_H = O(\text{polylog } n)$. Given this assumption, the objective of the filter is to find an MIS.

6.1 Algorithm Outline

During the algorithm, each node is in one of 7 states: the *active state* $\mathbb{A}$, the *handshake states* $\mathbb{H}'$ and $\mathbb{L}'$, the *red-blue game* states $\mathbb{H}$ and $\mathbb{L}$, the MIS state $\mathbb{M}$ or the exclusion state $\mathbb{E}$. State $\mathbb{A}$ (*active*) indicates the initial state; state $\mathbb{M}$ indicates that the node is in the MIS (permanently); and $\mathbb{E}$ (*eliminated*) indicates nodes that know of a neighboring MIS node. States $\mathbb{L}'$ (*leader candidate*) and $\mathbb{L}$ (*leader*) are temporary states through which a node v passes to get to state $\mathbb{M}$, while states $\mathbb{H}'$ (*herald candidate*) and $\mathbb{H}$ (*herald*) are accompanying temporary states through which a node u passes to help a neighboring node v to pass through states $\mathbb{L}'$ and $\mathbb{L}$ to get to state $\mathbb{M}$.

In general, a node v can go to state $\mathbb{M}$ (i.e., join the MIS) in two ways: (1) either v does not receive any message for a long time and it joins the MIS assuming it is alone, or (2) v joins the MIS with the help of one of its neighbors u. In the latter case, in order to get to state $\mathbb{M}$, node v goes through states $\mathbb{L}'$ and $\mathbb{L}$, while u goes through states $\mathbb{H}'$ and $\mathbb{H}$ simultaneously. During these states, u helps node v to make sure that no other neighbor of v is trying to join MIS.

Until the state of a node v in herald filter is determined (i.e., until it moves to $\mathbb{M}$ or $\mathbb{E}$), it maintains a counter $lonely(v)$ that measures for how long v has not heard from any neighbors; in addition, it maintains a parameter $\gamma(v)$, called the *activity level*, which is always in $\left[\frac{1}{4\Delta_H}, \frac{1}{2}\right]$ and governs the behavior of v in state $\mathbb{A}$. By definition, we assume that for nodes v in states $\mathbb{M}$ and $\mathbb{E}$ and for nodes v that are not presently in the herald filter, we have $\gamma(v) = 0$.

We divide the filter into 4 parts, depending on whether:

(i) the node is in the active state $\mathbb{A}$ (Section 6.2),
(ii) the handshake states $\mathbb{H}'$ and $\mathbb{L}'$ (Section 6.3),
(iii) the red-blue game states $\mathbb{H}$ and $\mathbb{L}$ (Section 6.4), or
(iv) the MIS state $\mathbb{M}$ (Section 6.5).

6.2 Active State

Consider a node w that is in state $\mathbb{A}$ in round r. In the active state, we use $O(\log \log n)$ channels, divided into three sets:
(i) the *active channels* $\{\mathcal{A}_1, \ldots, \mathcal{A}_{n_\mathcal{A}}\}$, (ii) the *lonely channels* $\{\mathcal{S}_1, \ldots, \mathcal{S}_{n_\mathcal{S}}\}$, and (iii) the *report channels* (see Section 5), where $n_\mathcal{A}, n_\mathcal{S} = O(\log \Delta_H) = O(\log \log n)$. In round r, node w does one of the following three things, with probability $\gamma(w)$ for (a), probability $0.9 - \gamma(w)$ for (b), and probability 0.1 for (c):

(a) Node w picks an active channel using an exponential distribution, choosing channel $\mathcal{A}_j$ with probability 2^{-j}. Then, with a fixed constant probability π_ℓ (chosen in the analysis), w listens to that channel, and with probability $1 - \pi_\ell$, w transmits its id on that channel.
(b) Node w listens to one of the $3\alpha(1)$ report channels chosen uniformly at random.
(c) Node w runs a protocol that we call the *loneliness support block*, and explain later in this subsection.

In (a), if w transmits on a channel $\mathcal{A}_i$ in state $\mathbb{A}$, then w goes to state $\mathbb{L}'$, attempting to become a leader. On the other hand, if w listens and receives a message from a node v, then

w goes to state $\mathbb{H}'$ (while v moves to state $\mathbb{L}'$). Node w will try to help v to become a leader and join the MIS. In (b), if w hears an id with status $\mathbb{M}$ on a report channel, then w is dominated by an MIS node and enters state $\mathbb{E}$ (*eliminated*).

Loneliness Support Block. Each node w maintains a counter *lonely*, to keep track of how long it has been in the herald filter without receiving any messages. Whenever w receives a message from a neighbor (anywhere in the herald filter), it resets the *lonely* counter. If *lonely* exceeds a threshold $\tau_{lonely} = \Theta(\log n \log \log n)$, then node w 'assumes' that it is isolated (i.e., that it does not have any neighbor in the herald filter). In this case, w joins the MIS and moves to state $\mathbb{M}$. Node w may in fact *not* be isolated, since a neighbor can show up later. However, we show in Theorem 7.10 that this is in fact safe.

Every time w executes the loneliness support block, it picks a channel $\mathcal{S}_j$ uniformly at random from the lonely channels. Then w transmits on channel $\mathcal{S}_j$ with probability 2^{-j}; otherwise, it listens to channel $\mathcal{S}_j$. If w receives a message, it resets its *lonely* counter to zero.

Activity Level Adjustment. Now we explain the adjustment of $\gamma(w)$. When w enters the herald filter, $\gamma(w)$ is $\frac{1}{4\Delta_H}$. The value of $\gamma(w)$ is gradually increased by a small constant factor every round, until it reaches the maximal possible value of $1/2$ after $O(\log \log n)$ rounds. The intuitive idea behind this activity level is as follows. Because of nodes waking up asynchronously and the fact that nodes exit the decay filter and enter the herald filter in an asynchronous manner, we need to deal with an undesirable fact: the transmission of the new nodes that enter the herald filter might affect the MIS election process which is going on among the nodes that entered the herald filter a while before that. With the gradual change in the activity level $\gamma(w)$, we can control this undesired effect and keep it below a tolerable level.

Thus, on first entering the herald filter, a node listens most of the time, but eventually, after some $O(\log \log n)$ steps, it spends a constant fraction of its time using the active channels to try to become a leader or a herald.

6.3 The Handshake

Consider a node h that just moved from state $\mathbb{A}$ to state $\mathbb{H}'$ when it received a message from a node ℓ, that has also just entered state $\mathbb{L}'$. Then, h and ℓ perform a 6-round *handshake* on a designated handshake channel $\mathcal{H}$. If this handshake succeeds, then node h moves to state $\mathbb{H}$ and ℓ moves to state $\mathbb{L}$. Otherwise, both return to state $\mathbb{A}$.

The handshake proceeds as follows: In rounds 1 and 2, h transmits the id of ℓ on $\mathcal{H}$, and ℓ listens. If ℓ receives both of these messages successfully, then in rounds 3 and 4, ℓ transmits its id on $\mathcal{H}$, and h listens. In addition, ℓ transmits a *meeting channel*, i.e., a randomly chosen report channel, which is used in the red-blue game (see Section 6.4). Finally, assuming that these messages are received successfully by h, then in rounds 5 and 6, h again transmits the id of ℓ on $\mathcal{H}$ and ℓ listens. If in any of these rounds, either of these nodes does not receive the message that it was supposed to receive, then it considers the handshake failed and returns to state $\mathbb{A}$.

Each of the 3 transmissions in the handshake is repeated twice in order to synchronize properly with the red-blue game and the nodes in the MIS. Nodes in these later states broadcast in every other round. By requiring two consecutive successful rounds of the handshake, we can be sure that there is no concurrent red-blue game or neighboring MIS node.

Note that it is possible for ℓ to consider the handshake failed due to not receiving a message in round 5 or 6, while h assumes that the handshake was performed successfully. This situation is detected in the first 6 rounds of the red-blue game.

It is easy to see that one of the necessary conditions for a handshake between some $v \in \mathbb{L}'$ and $u \in \mathbb{H}'$ to be successful is that h must be the only herald candidate trying to perform a handshake with ℓ at that time. Hence, the nodes that enter states $\mathbb{H}$ and $\mathbb{L}$ can be viewed as *leader-herald pairs*.

6.4 The Red-Blue Game

Ideally, we would like the leaders to form an independent set (and to also be independent of nodes in $\mathbb{M}$). This would allow us to send the leaders directly to the MIS. However, this is not always the case as multiple leaders can be adjacent. The goal of the red-blue game is to detect such bad leaders (i.e., adjacent leaders) and *knock them out*, back to state $\mathbb{A}$, along with their heralds.

For this purpose, we use a simple algorithm which we call the *red-blue game*. The red-blue game uses a designated channel $\mathcal{G}$, along with the handshake channel $\mathcal{H}$ and the report channels.

A single red-blue game is a 6-round protocol that is executed by a leader-herald pair (ℓ, h). During each game, it is possible that the pair is *knocked out*, meaning that both nodes go back to state $\mathbb{A}$. If the pair finishes $\Theta(\log n)$ red-blue games without getting knocked out, then ℓ assumes that it does not have an adjacent leader and joins the MIS.

The 6 rounds of a red-blue game are as follows: In rounds 1, 3 and 5 of the game, both ℓ and h transmit on the handshake channel $\mathcal{H}$. These transmissions block channel $\mathcal{H}$ so that adjacent nodes cannot perform a successful handshake and thus, no new adjacent leader-herald pair can be created until either ℓ joins the MIS or the pair is knocked out.

The main rounds of the game are rounds 2 and 4. In both rounds, h broadcasts ℓ's id on channel $\mathcal{G}$. ℓ picks a random color in the set $\{red, blue\}$ at the beginning of the 6-round protocol. In round 2, if ℓ chose *red*, then it transmits its id on channel $\mathcal{G}$, and if it chose *blue*, it listens to $\mathcal{G}$. In round 4, the behavior is reversed: ℓ listens if it chose *red* and it transmits if it chose *blue*.

Each time ℓ is listening to channel $\mathcal{G}$, by default, it should receive the message of h. If ℓ does not receive that message, it means that another node is also transmitting on channel $\mathcal{G}$—either a leader, another herald or an MIS node. If this happens, ℓ gets knocked out.

In round 6, ℓ transmits on the meeting channel, while h listens on it. The content of the sent message is whether the red-blue game succeeded (i.e., whether ℓ detected any collisions) and the meeting channel for the next red-blue game chosen uniformly at random among the report channels (for the first red-blue game, the meeting channel is fixed during the handshake). If h does not receive a message from ℓ indicating that the game succeeded, then h gets knocked out. (Notice that h may not receive such a message due to a collision, in which case ℓ gets knocked out in the next red-blue game when it fails to receive a message from h.) Note that the nodes that are knocked out go back to state $\mathbb{A}$ only after they have finished the 6 rounds of their red-blue game.

The objective of the even rounds is that if two leaders are adjacent and act synchronously (round-wise), then with probability 1/2, *both* leaders get knocked out. This is because if both leaders choose different colors *red* and *blue*, then they fail to receive the message from their respective heralds in rounds 2 and 4. Thus, if a leader-herald pair passes the red-blue game $O(\log n)$ times, then, w.h.p., there is no synchronized neighboring leader.

In the analysis, we show that because of the handshake rules, there are only very few configurations for two leader-herald pairs to be adjacent. Basically either the two leaders or the two heralds neighbor each other and operate synchronously, or if the leader of one and the herald of another pair are neighboring, then their starts of the red-blue games are shifted by exactly 2 rounds. When combined with the properties of the red-blue game, this ensures that only one leader moves on to the MIS.

6.5 The MIS State

Nodes in the MIS state need to continue to broadcast to prevent neighboring nodes from joining the MIS. This is accomplished by broadcasting with constant probability on $\mathcal{H}$, $\mathcal{G}$ and the report channels. More specifically, each node v that is in state $\mathbb{M}$ (i.e., that has joined MIS) performs one of the following two steps: (i) If v did not broadcast its id on channel $\mathcal{H}$ in the previous round, then it does so in the current round. (ii) If v did broadcast on channel $\mathcal{H}$ in the previous round, then with probability:

a. 1/2 it broadcasts its id and status on channel $\mathcal{H}$,
b. 1/4 it broadcasts its id and status on channel $\mathcal{G}$,
c. 1/4 it broadcasts its id and status on channel $\mathcal{R}_k$, with k chosen uniformly at random in $\{1, \ldots, 3\alpha(1)\}$.

Case (a) blocks any ongoing handshakes. Case (b) knocks back neighboring leaders to state $\mathbb{A}$, preventing the red-blue game from succeeding. Case (c) knocks back neighboring heralds to state $\mathbb{A}$, and also eliminates neighboring nodes in state $\mathbb{A}$, causing them to move to $\mathbb{E}$. These ongoing broadcasts ensure that we satisfy guarantee (G1) introduced in Section 4.

Note that channel $\mathcal{H}$ is blocked at least once every two rounds. Thus, after v has been in state $\mathbb{M}$ for 6 rounds, no new neighbors of v can switch to state $\mathbb{L}$. On the other hand, note that in every period of two rounds, with constant probability, v transmits once on channel $\mathcal{G}$. The transmissions on channel $\mathcal{G}$ knock back adjacent leaders that might have been created when (or immediately after) v switched to state $\mathbb{M}$ due to the *lonely* counter. Finally, the transmissions on the report channels let the neighboring nodes of v know that they are dominated by v, causing them to halt. Note that those transmissions can also knock back neighboring heralds to state $\mathbb{A}$.

7. HERALD ANALYSIS

Here, we present and overview of the analysis of the herald filter. For detailed proofs we refer to [9].

7.1 The Analysis of the Active State

We first present some facts about the transitions of nodes from state $\mathbb{A}$ to states $\mathbb{L}'$ and $\mathbb{H}'$. We show that for the k-neighborhood of some node u, the probability that no node in $N^k(u)$ is being elected as a herald candidate (switching from state $\mathbb{A}$ to $\mathbb{H}'$) is constant, and, by adjusting π_ℓ, arbitrarily close to 1. We then give some conditions under which

the creation of a single herald candidate happens with constant probability.

DEFINITION 7.1. (**Activity Sum**) *For a node u we define* $\Gamma(u) := \sum_{v \in N^1(u)} \gamma(v)$. *We call this the* activity sum *or* activity mass *of node u.*

LEMMA 7.2. *Fix a constant positive integer k. For any round t and node u, with probability $1 - O(\pi_\ell \alpha(k))$, no node $v \in N^k(u)$ switches from state $\mathbb{A}$ to state $\mathbb{H}'$ in round t.*

PROOF SKETCH. For the proof we solely focus on non-isolated nodes in the graph $G_\mathbb{A}$ induced by nodes in state $\mathbb{A}$. For a node v to become a herald candidate it has to receive a message from one of its neighbors w (event $B^{v,w}$). Careful analysis of those events gives us that the probability that v receives a message from *any* neighbor on channel $\mathcal{A}_m$ is at most $\pi_\ell 2^{-2m+2} \gamma(v) \Gamma(v) e^{-2^{-m-1}\Gamma(v)}$. This probability is maximized on channel $\mathcal{A}_\lambda$ for $\lambda :\approx \log \Gamma(v)$ (or $\lambda := 1$ if $\Gamma(v) < 1$) where the probability is $O\big(\pi_\ell \frac{\gamma(v)}{\Gamma(v)}\big)$. The probability falls rapidly as we move away from channel $\mathcal{A}_\lambda$. A union bound over all $v \in N^k(u)$, combined with Lemma 5.1, gives the desired result. $\square$

DEFINITION 7.3. (**Fatness**) *We call a node u (or respectively its neighborhood $N(u)$) η-fat for some value $\eta > 0$, if it holds that $\Gamma(u) \geq \eta \cdot \max_{v \in N(u)}\{\Gamma(v)\}$.*

LEMMA 7.4. *Let t be a round in which for a node u in state $\mathbb{A}$ in the herald filter it holds that there is no herald, leader, or herald candidate in $N^2(u)$. Furthermore, all neighbors of MIS nodes in $N^2(u)$ are in state $\mathbb{E}$, $\Gamma(u) \geq 1$, and either*

(a) $\Gamma(u) < 3\alpha(1)$, u is $\frac{1}{3\alpha(1)}$-fat, and $\gamma(u) = \frac{1}{2}$, or

(b) u is $\frac{1}{2}$-fat and $\Gamma(u) \geq 3\alpha(1)$.

Then by round $t' \in [t, t+7]$, with probability $\Omega(\pi_\ell)$ either a node in $N^2(u)$ joins the MIS or a pair $(l, h) \in \mathbb{L} \times \mathbb{H}$ is created in $N^1(u)$ such that $(N(\{l, h\}) \setminus \{l, h\}) \cap (\mathbb{H}' \cup \mathbb{H} \cup \mathbb{L}) = \emptyset$.

PROOF SKETCH. The proof is much more delicate, but partially follows the lines of the proof of Lemma 7.2. First we make sure that w.c.p. either in round t or $t + 1$ in $N^1(u)$ nodes contributing a constant fraction to $\Gamma(u)$ are in state $\mathbb{A}$, including u itself. Then we first lower bound the probability that some nodes $v, w \in N^1_\mathbb{A}(u)$ in that round meet alone on some channel and no other nodes nearby receive a message there. For case (a) we analyze the probability that this event happens for channel $\mathcal{A}_1$ and manage to lower bound it by $\Omega(1)$. For case (b) we use Lemma 5.1 to ensure the existence of neighbors v of u with high activity levels in u's and v's joint neighborhoods. Here we choose channel $\mathcal{A}_\lambda$ with $\lambda :\approx \log \Gamma(u)$ and show that for the previously mentioned neighbors the probability of meeting someone from $N^1(u)$ is in $\Omega\big(\frac{\gamma(v)}{\Gamma(u)}\big)$. Lemma 5.1 also gives us a lower bound of how much activity levels such neighbors provide and in total we can prove that exactly one of u's neighbors v meets another node w on $\mathcal{A}_\lambda$ is in $\Omega(1)$. Lemma 7.2 gives us a $1 - O(\pi_\ell)$ probability that no herald candidate is created at all. Lower bounding the probability that no herald candidate is created on a different channel than $\mathcal{A}_\lambda$ and combining this with the previous results gives the creation of a single herald candidate in round t resp. $t + 1$. Lemma 7.2 applied to the remaining rounds in $[t, t+7]$ finishes the proof. $\square$

7.2 The Analysis of the Handshake

In the following lemma, we study the circumstances under which two adjacent leader-herald pairs can coexist.

LEMMA 7.5. *In round r consider two leader-herald pairs (l_1, h_1) and (l_2, h_2) and suppose that the pairs started their most recent handshakes in rounds r_1 and r_2, $r_1 \leq r_2$, respectively. Say that edge e is* crossing *if one of its endpoints is in $\{l_1, h_1\}$ and its other endpoint is in $\{l_2, h_2\}$. Then, either no crossing edge exists or exactly one of the following conditions is satisfied: (1) $r_1 = r_2$ and crossing edges are $\{l_1, l_2\}$ and/or $\{h_1, h_2\}$, (2) $r_2 = r_1 + 2$ and the only crossing edge is $\{l_1, h_2\}$.*

7.3 The Analysis of the Red-Blue Game

We next study the exact guarantees of when and how pairs get knocked out in the red-blue games.

DEFINITION 7.6. *(**Maturity**) We say that candidate v is* mature *in round t, if v is in round 5 or 6 of its respective handshake.*

DEFINITION 7.7. *(**Good Pair, Bad Pair**) Consider a leader-herald pair (l, h) in round t. We say pair (l, h) is a* good pair *if in round t none of the neighbors of l (other than h) is in state $\mathbb{L}$, $\mathbb{H}$ or is a mature candidate. Otherwise we say that (l, h) is a* bad pair*.*

LEMMA 7.8. *If a pair (l, h) is good in round t and they started their first red-blue game in round r, then, w.h.p., either*

- *the related leader l joins the MIS by the end of round $r + \tau_{red-blue} = r + O(\log N)$, or*
- *a node $v \in N(l) \cup N(h)$ joins the MIS before round $r + \tau_{red-blue} = r + O(\log N)$ by increasing its lonely counter above τ_{lonely}.*

LEMMA 7.9. *Consider a node v and suppose that in an arbitrary round t, there is a leader or herald of a bad pair in $N^3(v)$. Then, with constant probability, in round $t + 12$, no node in $N^3(v)$ is in state $\mathbb{H}'$ and all leaders and heralds are part of a good pair.*

PROOF SKETCH. First we look at the leaders (or leaders-to-be) that are *not* neighboring other leaders. The pairs related to these leaders either become good or the leaders neighbor heralds of other pairs. Those leaders form an independent set, thus, their amount is limited. W.c.p., all of them choose the wrong color in their respective red-blue game which causes them to be kicked out by those heralds. Second we look at leaders that have other leaders nearby, which by Lemma 7.5 have to by completely synchronized. If we just look at the graph induced by these leaders (synced to one specific round), then we can find two disjoint independent sets S and T, and by adding at most $O(1)$ further nodes to T we get that each such leader either is in S or T and neighbors at least one node of the other, or it has each a neighbor in each of those sets. If all nodes in S choose blue and all nodes in T choose red, which happens w.c.p., then all nodes get kicked out in their red-blue game. $\square$

7.4 The Analysis of the MIS State

Here we present the main safety guarantee of our MIS algorithm.

LEMMA 7.10. *W.h.p., the nodes in state $\mathbb{M}$ always form an independent set. Moreover, if a node v enters state $\mathbb{M}$ in round t and a node $w \in N(v)$ is awake in round t, then in round $t' = t + O(\log n)$, w.h.p., w is in state $\mathbb{E}$.*

7.5 Putting the Pieces Together

In this section, we wrap everything up to show that guarantees (G1), (G4) and (G5) hold. Together with the guarantees (G2) and (G3) handled in Section 5 we finalize the proof of Theorem 4.1.

Lemma 7.10 immediately proves (G1) for all nodes currently in the herald filter. All nodes in the decay filter, both at the beginning as well as during the main body part, listen to the report channels w.c.p. in every round. Thus, either they learn of a neighboring MIS node within $O(\log n)$ rounds, or they move forward to the herald filter in that time bound. In the latter case, after that transition, Lemma 7.10 takes care of those nodes.

Lemma 7.10 also immediately gives us (G5). The only thing that remains to be shown is the progress guaranteed by (G4). To do so, we use the following lemma.

LEMMA 7.11. *Consider two neighboring nodes u and v such that both are in the herald filter at time t and assume that no node in $N(u) \cup N(v)$ has joined the MIS by time t. Then, w.h.p., some node u' in the $O(\log \log N)$-neighborhood of u joins the MIS between times t and $t + O(\log N)$.*

We now sketch how to obtain guarantee (G4). A node u in the herald filter in round r does not leave the herald filter before becoming dominating or dominated. If u has no neighboring nodes in the herald filter for $O(\log n \log \log n)$ consecutive rounds, u joins the MIS. Also, u can only have $\alpha(2)$ periods in which it is alone and where MIS nodes in $N^2(u)$ have already eliminated their neighbors. Each such period starts when some neighbor in the herald filter gets eliminated by a new MIS node in distance 2 from u. Whenever u has a neighbor in the herald filter Theorem 7.11 implies that within $O(\log n)$ rounds, w.h.p., a new MIS node is created in the $O(\log \log n)$-neighborhood of u. Clearly this can only happen $O(\alpha(\log \log n)) = O(\text{poly} \log \log n)$ times and thus after $O(\log(n)\alpha(\log \log n))$ rounds, either u or one of its neighbors is dominating and thus guarantee (G4) is satisfied.

8. OTHER PROBLEMS

In this section, we use our MIS algorithm solution as a building block in solving other problems efficiently in the multichannel environment. Our main technical result is a new algorithm that uses the MIS solution as a subroutine to build a constant-degree connected dominating set (CDS) in $O\left(\frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds. We then use this structure as an overlay to derive solutions to broadcast and leader election that run in $O\left(D + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds, and to k-message multi-broadcast that runs in $O\left(D + k + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ and $O\left(D + k \log n + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds for unrestricted and restricted message sizes, respectively.

Connected Dominating Set. First, we show how to construct a constant-degree CDS in $O\left(\frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds. At a high-level: our solution builds an MIS then connects every pair of MIS nodes that are within 3 hops using a constant-length path. The result is a constant-degree

CDS (see [7]). We provide a brief description of the algorithm here. See [9] for further details.

Special care must be taken to deal with two factors related to the transition of nodes from the MIS stage to the CDS stage: (1) MIS nodes must keep revisiting the MIS algorithm to prevent newly activated nodes from joining the MIS; (2) since nodes might end the MIS stage at different rounds, they might also start the CDS stage at different rounds, causing synchronization issues (the CDS stage cycles through fixed-length phases). In [9], we detail our strategy for addressing these issues. For concision, in the summary below, assume that nodes start the CDS stage in the same round and MIS nodes do not revisit the MIS algorithms.

The CDS stage consists of repeatedly iterating four phases made of six total rounds. During the first phase, an MIS node broadcasts on a designated *CDS channel* with constant probability. This announcement notifies nearby neighbors of its existence, informs them that a new iteration is beginning, and sends out a random sequence of bits, the *coin sequence*, that is used to synchronize neighbors in the remaining rounds of the iteration. It also includes a list of nearby MIS nodes (that it knows about so far) and (short) paths to these nodes. Any node that learns that it is on such a path adds itself to the CDS. Also during this phase, any non-MIS node that is already in the CDS broadcasts a message with constant probability in the announcement phase on the CDS channel, propagating its information about the currently selected paths. (Once an MIS node selects a path, this allows the information to propagate down the path.)

Rounds 2 and 3 are the *search phase*. To reduce contention, a non-MIS node u broadcasts in this phase only if there is a single MIS neighbor w that has a 1 in its coin sequence for this round. Otherwise it will only receive. In round 2, which handles sparse regions, a node broadcasts the id of MIS node w on the CDS channel: in each iteration, it cycles through the $\log \log n$ probabilities $\{\frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{\log n}\}$. In round 3, which handles dense regions, a node chooses a channel $c \in [\log n]$ with probability 2^{-c} (i.e., as in the herald protocol from the MIS algorithm), and then broadcasts MIS node w with constant probability. In both search rounds, a non-broadcaster listens—on the CDS channel (round 2), or a channel chosen uniformly from $[\log n]$ (round 3). If a node learns about a new MIS node, it adds it to the set of *discovered* MIS nodes.

The remaining rounds 4–6 are the *report phase*. If a non-MIS node u has discovered a new MIS node (and a path to this node), and if u has a single MIS neighbor w with a 1 in its coin sequence for this round, then u will report to w. In round 3, it broadcasts its new knowledge on the CDS channel with constant probability. In round 5, it chooses channel $c \in [\log n]$ with probability 2^{-c} and then broadcasts its new information with constant probability. If u decides to listen and it receives a message it then acts as a herald for this message, rebroadcasting it on the CDS channel with constant probability in the next round.

The basic argument is as follows: imagine two MIS nodes u and z are connected by a two-hop path. in this case they have a neighbor in common, and will learn about each other in a subsequent report phase. Assume, on the other hand, they are connected by a three-hop path. Let V be the neighbors of u and let W be the neighbors of z. If some node $w \in W$ has only a few neighbors in V (i.e., $< \log n$), then w receives a message from V during the sparse search phase

rounds within $O(\log \log n)$ rounds, with constant probability. The same holds symmetrically if some node $v \in V$ has only a few neighbors in W. Otherwise, if every node $w \in W$ has at least $\log n$ neighbors in V, and every node $v \in V$ has at least $\log n$ neighbors in W, then, with constant probability, during the dense search phase rounds where W nodes are broadcasting but not V nodes, there will be a single broadcaster from W on some channel with at least one listener from V. This information will be relayed to u in subsequent report phases. This leads to the following conclusion:

THEOREM 8.1. *We can construct a constant-degree CDS in $O\left(\frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds, w.h.p.*

The problems below use a CDS as an overlay network. To best match the typical assumptions for these problems, we will assume synchronous starts—i.e., the CDS algorithm starts and ends at the same rounds for all nodes. Our algorithms all work without this assumption as well, requiring in this case only that the theorem statements be rewritten to guarantee their running time holds *after the first round in which a complete CDS is constructed.*

Broadcast. First, build a constant-degree CDS using the above algorithm. Then, the source delivers the message to its CDS neighbors. On receiving the message, a CDS node re-broadcasts it with constant probability in each round. Because the CDS nodes have constant degree, a standard Chernoff analysis shows the message will reach every CDS node in $O(D + \log n)$ rounds (and therefore every node within $O(\log n)$ more rounds), w.h.p. Combined with the running time of the CDS algorithm, the total running time is $O\left(D + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds, nearly reaching the $\Omega\left(D + \frac{\log^2 n}{\mathcal{F}}\right)$ centralized lower bound for the multi-channel setting [16]. Formally:

THEOREM 8.2. *We can solve the problem of global broadcast in $O\left(D + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds, w.h.p.*

Multi-Message Broadcast. The multi-message broadcast problem assumes k messages must be propagated to the entire network. As before, first build a constant-degree CDS. We then use the same logic as the report phase of the CDS algorithm to propagate the k messages from their sources to nearby CDS nodes. This routine uses $\log n$ channels and can deliver all k messages to nearby nodes in $O(k + \log n)$ rounds, w.h.p. Once the messages are in the CDS, how we propagate depends on our assumption on message size. For unrestricted message size, we can run the above simple broadcast algorithm, simply combining all messages a node has received into a single message, in each round. This requires $O(D + \log n)$ rounds to propagate all k once we have our CDS. If we assume restricted message size (i.e., $O(\text{poly}(\log n))$ bits), we can use the algorithm and analysis of [21]. As established in [21], this will require $O(D + k \log n)$ rounds (formally, F_{prog} in the relevant theorem is $O(1)$ while F_{ack} is $O(\log n)$). From this we conclude:

THEOREM 8.3. *It is possible to solve k-multi-broadcast in time $O\left(D + k + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ with unrestricted messages sizes, and in $O\left(D + k \log n + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ rounds with restricted message sizes, w.h.p.*

Leader Election. To elect a leader, run the broadcast algorithm with *all* nodes initiating broadcast with a message

containing their own ID, and having each node update its broadcast message in each round to include the largest ID it has received so far. Using a standard Chernoff analysis, we can show that the largest ID will propagate to all nodes within $O(D + \log n)$ rounds. Formally:

THEOREM 8.4. *W.h.p., leader election can be solved in* $O\left(D + \frac{\log^2 n}{\mathcal{F}}\right) + \tilde{O}(\log n)$ *rounds.*

9. REFERENCES

[1] I. 802.11. Wireless LAN MAC and Physical Layer Specifications, June 1999.

[2] N. Alon, L. Babai, and A. Itai. A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem. *Journal of Algorithms*, 1986.

[3] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A Lower Bound for Radio Broadcast. *Journal of Computer and System Sciences*, 43(2):290–298, 1991.

[4] R. Bar-Yehuda, O. Goldreich, and A. Itai. On the Time-Complexity of Broadcast in Multi-Hop Radio Networks: An Exponential Gap Between Determinism and Randomization. *J. Computer and System Sciences*, 45(1):104–126, 1992.

[5] L. Barenboim, M. Elkin, S. Pettie, and J. Schneider. The Locality of Distributed Symmetry Breaking. In *Proc. Symp. on Foundations of Computer Science (FOCS)*, pages 321–330, 2012.

[6] Bluetooth Consortium. *Bluetooth Specification Version 2.1*, July 2007.

[7] K. Censor-Hillel, S. Gilbert, F. Kuhn, N. Lynch, and C. Newport. Structuring Unreliable Radio Networks. In *Proc. 30th Symp. on Principles of Distributed Computing (PODC)*, pages 79–88, 2011.

[8] I. Chlamtac and S. Kutten. On Broadcasting in Radio Networks–Problem Analysis and Protocol Design. *IEEE Transactions on Communications*, 33(12):1240–1246, 1985.

[9] S. Daum, M. Ghaffari, S. Gilbert, F. Kuhn, and C. Newport. Maximal independent sets in multichannel radio networks. Technical Report 275, University of Freiburg, Department of Computer Science, 2013.

[10] S. Daum, S. Gilbert, F. Kuhn, and C. Newport. Leader Election in Shared Spectrum Networks. In *Proc. Symp. on Principles of Dist. Computing (PODC)*, pages 215–224, 2012.

[11] S. Daum, F. Kuhn, and C. Newport. Efficient Symmetry Breaking in Multi-Channel Radio Networks. In *Proc. 26th International Symposium on Distributed Computing (DISC)*, pages 238–252, 2012.

[12] S. Dolev, S. Gilbert, R. Guerraoui, D. R. Kowalski, C. Newport, F. Kuhn, and N. Lynch. Reliable Distributed Computing on Unreliable Radio Channels. In *the Proc. of the 2009 MobiHoc S³ Workshop*, 2009.

[13] S. Dolev, S. Gilbert, R. Guerraoui, F. Kuhn, and C. Newport. The Wireless Synchronization Problem. In *Proceedings of the Principles of Distributed Computing*, pages 190–199, 2009.

[14] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport. Gossiping in a Multi-Channel Radio Network: An Oblivious Approach to Coping with Malicious Interference. In *Proc. Symp. on Dist. Comp. (DISC)*, 2007.

[15] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport. Secure Communication Over Radio Channels. In *Proc. Symp. on Principles of Dist. Computing (PODC)*, 2008.

[16] S. Dolev, S. Gilbert, M. Khabbazian, and C. Newport. Leveraging Channel Diversity to Gain Efficiency and Robustness for Wireless Broadcast. In *Proc. Symp. on Dist. Comp. (DISC)*, 2011.

[17] A. Ephremides, J. E. Wieselthier, and D. J. Baker. A Design Concept for Reliable Mobile Radio Networks With Frequency Hopping Signaling. *Proc. of the IEEE*, 75(56–73), 1987.

[18] M. Farach-Colton, R. J. Fernandes, and M. A. Mosteiro. Lower Bounds for Clear Transmissions in Radio Networks. In *Proc. Latin American Symposium on Theoretical Informatics*, 2006.

[19] S. Gilbert, R. Guerraoui, D. Kowalski, and C. Newport. Interference-Resilient Information Exchange. In *Proc. Conf. on Computer Comm. (INFOCOM)*, 2009.

[20] T. Jurdzinski and G. Stachowiak. Probabilistic Algorithms for the Wakeup Problem in Single-Hop Radio Networks. In *Proc. Int. Symp. on Algorithms and Computation*, pages 535–549, 2002.

[21] F. Kuhn, N. Lynch, and C. Newport. The Abstract MAC Layer. In *Proc. Symp. on Dist. Comp. (DISC)*, 2009.

[22] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs. In *Proc. 19th International Symposium on Distributed Computing (DISC)*, 2005.

[23] M. Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM Journal on Computing*, 15(4):1036–1053, 1986.

[24] T. Moscibroda and R. Wattenhofer. Maximal Independent Sets In Radio Networks. In *Proc. 24th Symp. on Principles of Distributed Computing (PODC)*, pages 148–157, 2005.

[25] C. Newport. *Distributed Computation on Unreliable Radio Channels*. PhD thesis, MIT, 2009.

[26] S. Schmid and R. Wattenhofer. Algorithmic Models for Sensor Networks. In *Proc. 14th Int. Workshop on Parallel and Distributed Real-Time Systmes*, pages 1–11, 2006.

[27] J. Schneider and R. Wattenhofer. A Log-Star Distributed Maximal Independent Set Algorithm for Growth-Bounded Graphs. In *Proc. Symp. on Principles of Dist. Computing (PODC)*, pages 35–44, 2008.

[28] M. Strasser, C. Pöpper, and S. Capkun. Efficient Uncoordinated FHSS Anti-jamming Communication. In *Proc. Symp. on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2009.

[29] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj. Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping. In *the Proceedings of the IEEE Symposium on Security and Privacy*, 2008.