A Local Broadcast Layer for the SINR Network Model

Citation: Magnus M. Halldorsson, Stephan Holzer, and Nancy Lynch. 2015. A Local Broadcast Layer for the SINR Network Model. In Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC '15). ACM, New York, NY, USA, 129-138.

As Published: http://dx.doi.org/10.1145/2767386.2767432

Publisher: Association for Computing Machinery (ACM)

Persistent URL: http://hdl.handle.net/1721.1/100842

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

A Local Broadcast Layer for the SINR Network Model*

Magnús M. Halldórsson[†]
mmh@ru.is
Reykjavik University

Stephan Holzer[‡]
holzer@csail.mit.edu
MIT

Nancy Lynch[†]
lynch@csail.mit.edu
MIT

Abstract

We present the first algorithm that implements an *abstract MAC (absMAC)* layer in the *Signal-to-Interference-plus-Noise-Ratio (SINR)* wireless network model. We first prove that efficient SINR implementations are not possible for the standard absMAC specification. We modify that specification to an "approximate" version that better suits the SINR model. We give an efficient algorithm to implement the modified specification, and use it to derive efficient algorithms for higher-level problems of global broadcast and consensus.

In particular, we show that the absMAC *progress* property has no efficient implementation in terms of the SINR strong connectivity graph $G_{1-\varepsilon}$, which contains edges between nodes of distance at most $(1-\varepsilon)$ times the transmission range, where $\varepsilon > 0$ is a small constant that can be chosen by the user. This progress property bounds the time until a node is guaranteed to receive *some* message when at least one of its neighbors is transmitting. To overcome this limitation, we introduce the slightly weaker notion of *approximate progress* into the absMAC specification. We provide a fast implementation of the modified specification, based on decomposing the algorithm of [9] into local and global parts. We analyze our algorithm in terms of local parameters such as node degrees, rather than global parameters such as the overall number of nodes. A key contribution is our demonstration that such a local analysis is possible even in the presence of global interference.

Our absMAC algorithm leads to several new, efficient algorithms for solving higher-level problems in the SINR model. Namely, by combining our algorithm with high-level algorithms from [25], we obtain an improved (compared to [9]) algorithm for global single-message broadcast in the SINR model, and the first efficient algorithm for multi-message broadcast in that model. We also derive the first efficient algorithm for network-wide consensus, using a result of [31]. This work demonstrates that one can develop efficient algorithms for solving high-level problems in the SINR model, using graph-based algorithms over a local broadcast abstraction layer that hides the technicalities of the SINR platform such as global interference. Our algorithms do not require bounds on the network size, nor the ability to measure signal strength, nor carrier sensing, nor synchronous wakeup.

[*] Full version [16] available at arxiv.org.

[†] Supported by Icelandic Research Fund grants 120032011 and 152679-051.

[‡] Supported by the following grants: AFOSR Contract Number FA9550-13-1-0042, NSF Award 0939370-CCF, NSF Award CCF-1217506, NSF Award CCF-AF-0937274.

1 Introduction

Two active areas in Distributed Computing Theory are the attempts to understand wireless network algorithms in the *Signal-to-Interference-plus-Noise-Ratio (SINR) model* and *abstract Medium Access Control layers (absMAC)*.

- The SINR model captures wireless networks in a more precise way than traditional graph-based models, taking into account the fact that signal strength decays according to geometric rules and interference and does not simply stop at a certain border.
- Abstract MAC layers (a.k.a Local Broadcast Layers), express guarantees for local broadcast while hiding the complexities of managing message contention. These guarantees include message delivery latency bounds: an *acknowledgment bound* on the time for a sender's message to be received by all neighbors, and a *progress bound* on the time for a receiver to receive some message when at least one neighbor is sending.

In this paper we combine the strengths of both models by abstracting and modularizing broadcast with respect to global interference and decay via the SINR formula. This marks the start of a systematic study that simplifies the development of algorithms for the SINR model. At the same time we provide an example that modularizing and abstracting broadcast using MAC layers is beneficial and does not necessarily result in worse time-bounds than those of the broadcast algorithm being decomposed.

Traditionally, SINR platforms are quite complicated (compared to graph-based platforms), and consequently are very difficult to use directly for designing and analyzing algorithms for higher-level problems[1]. We show how absMACs can help to mask their complexity and make algorithms easier to design. This demonstrates the potential power of absMACs with respect to algorithm design for the SINR model. During this process we point out and overcome inherent difficulties that at first glance seem to separate the MAC layers from the SINR model and other physical models. These difficulties arise because absMACs are graph-based interference models, while physical models capture (global) interference by specific signal-propagation formulas. Overcoming this mismatch is a key difficulty addressed in this work.

We tackle this mismatch by introducing the concept of *approximate progress* into the absMAC specification and analysis. The definition of approximate progress enables us to obtain a good implementation of an absMAC, which enables anyone to immediately transform generic algorithms designed for an absMAC into algorithms for the SINR model. The main observation that inspired the definition of approximate progress is a proof, that no SINR absMAC implementation is able to guarantee fast progress in an SINR-induced graph G, while fast progress can be guaranteed with respect to an approximation $\tilde{G}$ of G. Roughly speaking, as SINR-induced strong connectivity graphs are defined based on discs representing transmission ranges, we choose $\tilde{G} := G_{1-2\varepsilon}$ to approximate $G := G_{1-\varepsilon}$ by making the disc a tiny bit smaller than in G.

This abstraction makes it easier to design algorithms for higher-level problems in the SINR model and has further benefits. One of the most intriguing properties of abstract MAC layers is their separation of global from local computation. This is beneficial in two ways. On the one hand this separation allows us to expose useful SINR techniques in the simple setting of local broadcast. On the other hand this separation provides the basic structure to perform an analysis based on local parameters, such as the number of nodes in transmission/communication range

[1] We refer by *higher-level problems* to e.g. network-wide broadcast, consensus, or computing fast relaying-routes, max-flow and other problems whose solution requires a good understanding of lower-level problems. Here, we refer by *lower-level problems* to e.g. achieving connectivity, minimizing schedules and capacity maximization, which are better understood by now.

and the distance-ratios between them, which is beneficial as pointed out in the full version of this paper [16]. Due to this, and the plug-and-play nature of the absMAC theory, we obtain a faster algorithms for global single-message broadcast than [9] and fast algorithms for global multi-message broadcast and consensus in the SINR model. To achieve these results, we simply plug our absMAC implementation and bounds into the results of [25] and [31].

Future Benefits of Abstract MAC Layers in the SINR Model. Many higher-level problems such as global broadcast, routing and reaching consensus are not yet well understood in the SINR model and recently gained more attention [9, 11, 18, 19, 21, 34, 35]. Many of these problems in the algorithmic SINR can be attacked in a structured way by using and implementing absMACs that hide all complications arising from the SINR model and global interference. Using MAC layers, graph-based algorithms can be analyzed in the SINR model even without knowledge of the SINR model and might still lead to almost optimal algorithms as we demonstrate here.

2 Contributions and Related Work

We devote large parts of this article to prove theorems on implementing an absMAC in the SINR model and how to modify the absMAC specification to get better results. Based on these theorems we derive results on higher-level problems in the SINR model. We provide more details on contributions in the full version of this paper [16]. Our model assumptions in the SINR model and absMAC are listed in Section 3 and are adapted from [9] and [20]. Table 1 summarizes our algorithmic contributions.

Efficient implementation of acknowledgments. Theorem 4.1 transfers Algorithm 1 of [17] and its analysis to implement fast acknowledgments of the absMAC and modifies it to use local parameters. The full version of this paper [16] provides a close lower bound.

Proof of impossibility of efficient progress. Theorem 4.2 shows that one cannot expect an efficient implementation of progress using the standard definition of absMAC. In particular one cannot implement an absMAC in the SINR model that achieves progress much faster than acknowledgments.

The notion of approximate progress. Achieving progress faster than acknowledgment is key to several algorithms designed for absMACs. Motivated by the above lower bound, we relax the notion of progress in the specification of an absMAC to *approximate progress*. Definition 5.1 introduces approximate progress with respect to an approximation (or some subgraph) of the graph in which local broadcast is performed. Although this new notion of approximate progress is weaker than the usual (single-graph) notion of progress, bounds on approximate progress turn out to be strong enough to yield, e.g., good bounds for global broadcast as long as G is, e.g., connected—see Theorem 7.2. The introduction of approximate progress is the main conceptual contribution of this article.

Efficient implementation of approximate progress. We modify the global single-message broadcast algorithm of [9] to guarantee approximate progress in an absMAC (a local multi-message environment). Our modifications make this algorithm suitable for a localized analysis, which bounds the runtime in terms of local parameters and the desired success probability, see Theorem 6.1. This analysis, which removes the parameter n from the runtime is the main technical contribution of this article and leads to the improved global broadcast algorithms mentioned below.

Global consensus, single-message and multi-message broadcast in the SINR model. We immediately derive an algorithm for global consensus (CONS) in Corollary 4.3 by combining our acknowledgment-bound with a result of [31]. Section 7 combines our absMAC implementation with results of [25] in a straightforward way to derive algorithms for global single-message broadcast (SMB) and global multi-message broadcast (MMB).

Task/Bound	Lower bound	Upper bound presented here
f_{ack}	$\Delta_{G_{1-\varepsilon}}\ ^{(+)}$	$\mathcal{O}\left(\Delta_{G_{1-\varepsilon}} \cdot \log\left(\frac{\Lambda}{\varepsilon_{ack}}\right) \quad + \quad \log(\Lambda)\log\left(\frac{\Lambda}{\varepsilon_{ack}}\right)\right)$
f_{prog}	$\Delta_{G_{1-\varepsilon}}^{(*)}$	$\mathcal{O}\left(\Delta_{G_{1-\varepsilon}} \cdot \log\left(\frac{\Lambda}{\varepsilon_{ack}}\right) \quad + \quad \log(\Lambda)\log\left(\frac{\Lambda}{\varepsilon_{ack}}\right)\right)$
f_{approg}	–	$\mathcal{O}\left(\left(\log^{\alpha}(\Lambda) + \log^{*}\left(\frac{1}{\varepsilon_{approg}}\right)\right)\log(\Lambda)\log\left(\frac{1}{\varepsilon_{approg}}\right)\right)$
global SMB	$\Omega\Big(D_{G_{1-\varepsilon}} \log\left(\frac{n}{D_{G_{1-\varepsilon}}}\right)$ $+ \quad \log^2(n)\Big)^{(\ddagger)}$	$\mathcal{O}\left(\left(D_{G_{1-2\varepsilon}} \quad + \quad \log\left(\frac{n}{\varepsilon_{SMB}}\right)\right)\log^{\alpha+1}(\Lambda)\right)^{(\dagger)}$
global MMB	$\Omega\Big(D_{G_{1-\varepsilon}} \log\left(\frac{n}{D_{G_{1-\varepsilon}}}\right)$ $+ \quad k\log(n) + \log^2(n)\Big)^{(\ddagger)}$	$\mathcal{O}\Big(D_{G_{1-2\varepsilon}} \log^{\alpha+1}(\Lambda) \quad + \quad k\Delta_{G_{1-\varepsilon}} \log\left(\frac{nk}{\varepsilon_{MMB}}\right)$ $+ \quad \mathrm{polylog}\left(\frac{nk\Lambda}{\varepsilon_{MMB}}\right)\Big)^{(\dagger)}$
global CONS	–	$\mathcal{O}\left(D_{G_{1-\varepsilon}}(\Delta_{G_{1-\varepsilon}} + \log(\Lambda))\log\left(\frac{n\Lambda}{\varepsilon_{CONS}}\right)\right)^{(\dagger)}$

Table 1: Summary of algorithmic results, see Section 3 for details on notation. The table compares our new upper bounds to (known and new) lower bounds. Known lower bounds are graph-based and transfer to our setting, as we use weaker assumptions. To compare graph-based lower bounds with our upper bounds, one might choose $\Lambda = n$ to account for possible high degree and choose $\varepsilon_{SMB} = \varepsilon_{MMB} = n^{-c}$ to achieve w.h.p. correctness. (*) Lower bound proven in this paper using absMAC assumptions of [25]. (†) Lower bounds require runtimes of global broadcast to depend on n even though we perform a local analysis. (‡) Combinations of lower bounds of [1, 12, 29] for graph based models. (+) Trivial lower bound (the full version of this paper [16] provides more details.)

2.1 Comparison of Algorithmic Results with Previous Work

Global single-message broadcast. Table 2 compares the runtime of our algorithm for global SMB with previous work. Currently [9] and [20] provide the best implementations of global SMB in the SINR model (see the runtimes in Table 2). The result of [9] is as good or better than [20] in case $\log^{\alpha+1}(\Lambda) \leq \log(n)$ and vice versa. To make it possible to compare our result to theirs, we need to choose $\varepsilon_{SMB} = 1/n^c$ such that global SMB is correct w.h.p.. Furthermore, we execute our algorithm with $\varepsilon' := \varepsilon/2$ instead of ε, while algorithms in previous work are executed without changing ε. This ensures that our bounds are stated in terms of the same parameter $D_{G_{1-\varepsilon}}$ rather than the possibly larger parameter $D_{G_{1-2\varepsilon}}$. At the same time the choice of ε' affects the runtime only by a constant factor. This results in a runtime of our algorithm of $\mathcal{O}\left((D_{G_{1-\varepsilon}} + \log(n))\log^{\alpha+1}(\Lambda)\right)$ in the strong connectivity graph $G_{1-\varepsilon}$. This improves over the algorithm presented in [9] in the full range of all parameters, and improves in case of $\log^{\alpha+1}(\Lambda) \leq \min(D_{G_{1-\varepsilon}} \log(n), \log^2(n))$ over the algorithm of [20]. Note that compared to [20] we (and [9]) assume knowledge of a bound on Λ. The key-ingredient of this improvement is our localized analysis in combination with [25].

Global multi-message broadcast. The algorithm for global MMB derived from [17] runs in $\mathcal{O}((D_{G_{1-\varepsilon}} + k)(\Delta_{G_{1-\varepsilon}} \cdot \log n + \log^2 n))$ time. Roughly speaking, our algorithm replaces the dependency on the potentially large multiplicative term $D_{G_{1-\varepsilon}}\Delta_{G_{1-\varepsilon}}$ by $D_{G_{1-\varepsilon}}$ up to polylog factors. Section 2.2 summarizes global MMB in related models.

Article	Runtime bound for global SMB	We improve this runtime in case of
this	$\mathcal{O}\left(\left(D_{G_{1-\varepsilon}} + \log(n)\right)\log^{\alpha+1}(\Lambda)\right)$	
[9]	$\mathcal{O}\left(D_{G_{1-\varepsilon}}\log^{\alpha+1}(\Lambda)\log(n)\right)$	all parameters and ranges
[20]	$\mathcal{O}\left(D_{G_{1-\varepsilon}}\log^{2}(n)\right)$	$\log^{\alpha+1}(\Lambda) \leq \min(D_{G_{1-\varepsilon}}\log(n)\ ,\ \log^2(n))$

Table 2: Comparison of the runtime of our global SMB protocol with previous results.

2.2 Related Work

We provide more details on related work in the full version of this paper [16].

Graph Based Wireless Networks (Chlamtac et al. [5]). Upper bounds for global SMB [8, 27] in networks of unknown topology are tight due to a lower bound of $\Omega(D\log(n/D) + \log^2 n)$ by Alon et al. [1, 29]. The sequence of work [3, 13, 24] considered global MMB. Ghaffari et al. [12] presented a lower bound of $\Omega(k\log n)$ for global broadcast of k messsages. These lower bounds can be transferred to the SINR-model using SINR-induced graphs.

Abstract MAC layer (Kuhn et al. [28]). The probabilistic absMAC we consider was defined by Khabbazian et al. [25]. AbsMAC implementations were provided in [25, 26] and applications were provided in [6, 7, 14, 25, 28, 31]. We use optimal algorithms for global SMB and MMB in the probabilistic absMAC due to [25] and results on CONS by Newport [31].

SINR model (e.g. Moscibroda and Wattenhofer [30]). Local broadcast was studied in various models, e.g in [15, 17, 33]. We modify the analysis of [17] to use purely local parameters. Global MMB algorithms can be implied once local broadcast is available. Yu et al. [34, 35] obtained almost optimal bounds using arbitrary power control. Arbitrary power control used in [35] can yield arbitrary speed ups compared to our model [22, 30] and we get close to their runtime. Global SMB was studied in the sequence of papers [9, 19, 20, 21] using various model assumptions. Daum et al. [9] proposed a model that uses weak model assumptions, which we use as well. Thanks to a completely new approach they show how to perform global broadcast in $G_{1-\varepsilon}$ within $\mathcal{O}(D\log^{\alpha+1}(\Lambda)\log(n))$ rounds w.h.p.. We transfer and modify this algorithm to implement approximate progress in a probabilistic absMAC and provide a significantly extended analysis. Shortly after [9], Jurdzinski et al. [20] came up with a $\mathcal{O}(D\log^2 n)$ algorithm that improves over the one of [9] for a range of parameters. Table 2 compares these results to ours. Power control was also used in [4] to achieve connectivity and aggregation, which in turn can be used for broadcast as well.

3 Model and Definitions

Graphs and their properties. Let $G = (V, E)$ be a graph over n nodes V and edges E. We denote by $d_G(v, w)$ the hop-distance between w and v (the number of edges on a shortest (u, v)-path), and by $D_G := \max_{u,v\in V} d_G(u, v)$ the *diameter* of graph G. All neighbors of v in G are called *G-neighbors* of v. We define v's neighborhood to be $N_G(v) := \{u|(v, u) \in E\}$ and extend this to $N_{G,r}(v) := \{u|d_G(v, u) \leq r\}$ for the r-neighborhood, $r \in \mathbb{N}$. For any set $W \subseteq V$ we generalize this to $N_{G,r}(W) := \bigcup_{w\in W} N_{G,r}(w)$. $\delta_G(v) := |N_G(v) \setminus \{v\}|$ denotes the degree of v and $\Delta_G := \max_{v\in V} \delta_G(v)$ the degree of G. Let $S \subseteq V$ be a subset of G's vertices, then $G|_S = (S, E|_S)$ denotes the subgraph of G induced by nodes S, where $E|_S := \{(u, v) \in E|u, v \in S\}$. A set $S \subseteq S' \subseteq V$ is called a *maximal independent set (MIS)* of S' in G if 1) any two nodes $u, v \in S$ are independent, that is $(u, v) \notin E$, and 2) any node $v \in S'$ is covered by some neighbor in S, that is $N_G(v) \cap S \neq \emptyset$. A graph $G = (V, E)$ is *(polynomial) growth-bounded* if there is a polynomial bounding function $f(r)$ such that for each node $v \in V$, the number of nodes in the neighborhood

$N_{G,r}(v)$ that are in any independent set of G is at most $f(r)$ for all $r \geq 0$. This allows us to bound the size of neighborhoods depending on the maximal degree of the network. When performing a localized analysis this yields union-bounds depending on the maximal degree, rather than the size of the network.

Lemma 3.1. *Let G be polynomially growth-bounded by function f, then it holds that $|N_{G,r}(v)| \leq \Delta f(r)$ for all $v \in V$ and $r \in \mathbb{N}$.*

Proof. The proof appears in the full version of this paper [16]. □

The SINR model. Nodes are located in a plane and we write $d(v, w)$ for the Euclidean distance between points v, w (often corresponding to node's positions). It is clear from the context when d refers to hop-distance or Euclidean distance. When a node v (of a wireless network) sends a message, it transmits with (uniform) power $P > 0$. A transmission of v is received successfully at a node u, if and only if $SINR_u(v) := \frac{P/d(v,u)^\alpha}{\sum_{w \in S\setminus\{u,v\}} P/d(w,u)^\alpha + N} \geq \beta$, where N is a universal constant denoting the ambient noise. The parameter $\beta > 1$ denotes the minimum SINR (signal-to-interference-noise-ratio) required for a message to be successfully received, α is the so-called path-loss constant. Typically it is assumed that $\alpha \in (2, 6]$, see [15]. Here, S is the subset of nodes in V that are sending. By $R := (P/\beta N)^{1/\alpha}$ we denote the transmission range, i.e. the maximum distance at which two nodes can communicate assuming no other nodes are sending at the same time. For $a \in \mathbb{R}^+$, we define $R_a := a \cdot R$. If $d(v, u) \leq R_a$ and $a < 1$, we say u and v are connected by a *a-strong* link. Like previous literature [2, 9, 10, 15, 23] we consider a link to be strong if it is $(1 - \varepsilon)$-strong for constant $\varepsilon > 0$. If $R_a < d(u, v) \leq R_1$, we say u and v are connected by an *a-weak* link. A $(1-\varepsilon)$-weak link is just called weak link. We consider the strong connectivity graph $G_{1-\varepsilon} = (V, E_{1-\varepsilon})$, where $(u, v) \in E_{1-\varepsilon}$, if $u, v \in V$ are connected by a strong link. Given a graph G, we denote by Λ_G the ratio between the maximum and minimum Euclidean length of an edge in E. In case that G is $G_{1-\varepsilon}$, we simply write Λ instead of $\Lambda_{G_{1-\varepsilon}}$.

Abstract MAC layers. We use the definitions of Ghaffari et al. [14] adapted to the probabilistic setting of [25]. To initiate a broadcast in a graph G, the MAC layer provides an interface to higher layers via input $bcast(m)_i$ for any node $i \in V$ and message $m \in M$. To simplify the definition of this primitive, assume w.l.o.g. that all local broadcast messages are unique. When a node $u \in V$ broadcasts a message m, the model delivers the message to all neighbors in E. It then returns an acknowledgment of m to u indicating the broadcast is complete, denoted by $ack(m)_u$. In between it returns a $rcv(m)_v$ event for each node v that received message m. This model provides two timing bounds . The first is the acknowledgment bound, which guarantees that each broadcast will complete and be acknowledged within f_{ack} time. The second is the progress bound, which guarantees : fix some $(u, v) \in E$ and interval of length f_{prog} throughout which u is broadcasting a message m; during this interval v must receive some message (though not necessarily m, but a message that some location is currently working on, not just some ancient message from the distant past). The progress bound, in other words, bounds the time for a node to receive some message when at least one of its neighbors is broadcasting. In both theory and practice f_{prog} is typically much smaller than f_{ack} [25]. Further motivation and power of these delay bounds is demonstrated e.g. in [14, 25, 28]. We emphasize that in abstract MAC layer models the order of receive events is determined non-deterministically by an arbitrary message scheduler. The timing of these events is also determined nondeterministically by the scheduler, constrained only by the above time bounds.

The Standard Abstract MAC Layer. Nodes are modeled as event-driven automata. While [14] assumes that an environment abstraction fires a wake-up event at each node at the beginning of each execution, we assume conditional wake-up to be consistent with the model of [9], see Definition 3.2. This is a weaker wake-up assumption with respect to upper bounds when compared to synchronous wake-up [14]. This strengthens our algorithmic results. In contrast to this our lower bounds assume synchronized wake-up, which is in turn the weaker assumption with respect to lower bounds. The environment is also responsible for any events specific to the problem being solved. In multi-message broadcast, for example, the environment provides the broadcast messages to nodes at the beginning.

Definition 3.2 (Conditional (a.k.a non-spontaneous) wake-up of [9] adapted to absMACs)**.** *Once a node receives an input from the MAC-environment (above the MAC layer) or a transmission from the network below the MAC layer, the node wakes up and participates in the algorithm.*

The enhanced abstract MAC layer. The enhanced abstract MAC layer model differs from the standard model in two ways. First, it allows nodes access to time (formally, they can set timers that trigger events when they expire), and assumes nodes know f_{ack} and f_{prog}. Second, the model also provides nodes an abort interface that allows them to abort a broadcast in progress.

The probabilistic abstract MAC layer. We use parameters ε_{prog} and ε_{ack} to indicate the error probabilities for satisfying the delay bounds f_{prog} and f_{ack}. Roughly speaking this means that the MAC layer guarantees that progress is made with probability $1 - \varepsilon_{prog}$ within f_{prog} time. With probability $1 - \varepsilon_{ack}$ the MAC layer correctly outputs an acknowledgment within f_{ack} time steps. More details can be found in Section 4.2 of [25].

Problems. We derive algorithms in the SINR-model that perform the tasks listed below correctly with probability $1 - \varepsilon_{task}$. When choosing $\varepsilon_{task} \leq n^{-c}$ we say that an algorithm performs a task with high probability (w.h.p.). Here, $c > 0$ is an arbitrary constant provided to the algorithm as an input-parameter. We use the notation w.h.p. only to compare our results with previous work.

Multi-message broadcast (MMB) problem [25]. This problem inputs $k \geq 1$ messages into the network at the beginning of an execution, perhaps providing multiple messages to the same node. We assume k is not known in advance. The problem is solved once every message m, starting at some node u, reaches every node in G. Note that we assume G is connected to be consistent with previous work in the SINR model, while in [14] this is not assumed. We treat messages as black boxes that cannot be combined.

Single-message broadcast (SMB) problem [25]. The SMB problem is the special case of MMB with $k = 1$. The single node at which the message is input is denoted by i_0.

Consensus problem (CONS) problem [31]. In this problem each node begins an execution with an initial value from $\{0, 1\}$. Every node has the ability to perform a single irrevocable *decide* action for a value in $\{0, 1\}$. To solve consensus, an algorithm must guarantee the following three properties: 1) *agreement*: no two nodes decide different values; 2) *validity*: if a node decides value v, then some node had v as its initial value; and 3) *termination*: every non-faulty process eventually decides.

General model assumptions. As in [9] wake up of nodes is conditional, see Definition 3.2. Nodes are located in the Euclidean plane[2] and locations are unknown. Nodes send with uniform power, where the fixed power level P is not known to the nodes. We use the common assumption that $\alpha > 2$, see [15]. No collision detection mechanism is provided. As previous work we assume $G_{1-\varepsilon}$ is connected. MAC-layer based work [25] requires us to assume that nodes can detect if a received message originates from a neighbor in a graph G–in our setting this is $G_{1-\varepsilon}$–(only one

[2] Our results can be generalized to any growth-bounded metric space when revising the assumption on α.

graph G is used in [25], while messages from any sender in the network might arrive but do not cause rcv-events). We remark that the assumption that nodes can detect if a received message originated in the $G_{1-\varepsilon}$-neighborhood is not used by any of the algorithms presented in this paper. Therefore this assumption could be dropped if one is not interested in implementing an absMAC that performs local broadcast exactly in $G_{1-\varepsilon}$. In particular, our absMAC implementation outputs *rcv* events for all *bcast*-messages received, which can be modified if required by other higher-level algorithms designed for absMACs and when the $G_{1-\varepsilon}$-neighborhood is known. The reader might consider the full version of this paper [16] for further details.

4 Efficient Acknowledgments and Impossibility of Fast Progress

Theorem 4.1. *In the SINR model using the assumptions of Section 3, acknowledgments of an absMAC can be implemented w.r.t. graph $G_{1-\varepsilon}$ with probability guarantee $1-\varepsilon_{ack}$ in time $f_{ack} = \mathcal{O}\left(\Delta_{G_{1-\varepsilon}} \log\left(\frac{\Lambda}{\varepsilon_{ack}}\right) + \log(\Lambda)\log\left(\frac{\Lambda}{\varepsilon_{ack}}\right)\right)$.*

The proof of Theorem 4.1 is a straightforward modification of [17] to local parameters.

Many algorithms that are implemented in an absMAC benefit from the fact that typically f_{prog} is much smaller than f_{ack}. Often it is the case that $f_{prog} = \mathcal{O}(\text{polylog}\,(f_{ack}))$. We show that for any implementation of the absMAC [25] for $G_{1-\varepsilon}$ in the SINR model such a difference of the runtime is impossible. As the bound on f_{ack} in Theorem 4.1 is close to our lower bound on f_{prog}, we conclude that this algorithm is an almost optimal implementation of absMAC in the SINR-model with respect to both f_{ack} and f_{prog}.

Theorem 4.2. *For worst-case locations of points there is no implementation of the absMAC in the SINR model that provides local broadcast in $G_{1-\varepsilon}$ and achieves fast progress. In particular it holds that $f_{prog} \geq \Delta_{G_{1-\varepsilon}}$. This is true even for an optimal schedule computed by an (even central) entity that has unbounded computational power, has full knowledge as well as control of the network and can choose an arbitrary power assignment.*

We defer the full proof to the full paper [16]. The key idea is to have two sets U and V of nodes, each set of nodes on a line with unit distance between nodes. These two lines are located at distance $R_{1-\varepsilon} := 10\Delta$ to each other such that at most one node in set V can receive a message from U at a time. Note that this is independent of ε.

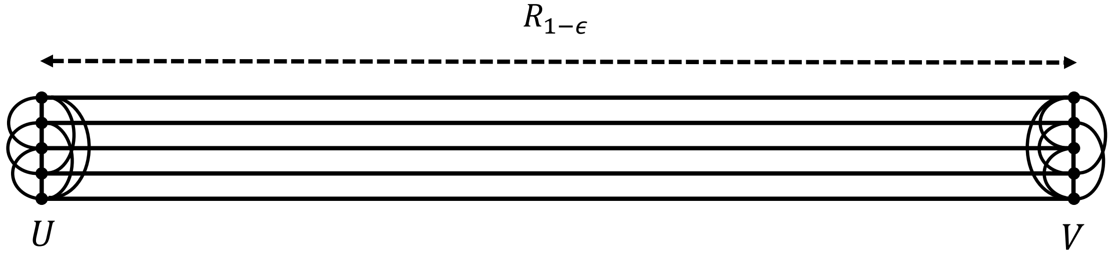

Figure 1: Graph $G_{1-\varepsilon}$ based on the construction used in the proof of Theorem 4.2. Here we choose $\Delta = 5$.

Despite this lower bound we can already provide a first application of designing an absMAC for the SINR. Corollary 4.3 is an application of Theorem 4.1 to [31], see the full version of this paper [16].

Corollary 4.3 (Theorem 4.2. of [31] transferred to our setting)**.** *In the SINR model using the assumptions of Section 3, network-wide consensus can be solved with probability* $1-\varepsilon_{CONS}$ *in time* $f_{CONS} = \mathcal{O}\left(D_{G_{1-\varepsilon}}(\Delta_{G_{1-\varepsilon}} + \log(\Lambda))\log\left(\frac{n\Lambda}{\varepsilon_{CONS}}\right)\right)$.

5 Approximate Progress

Motivated by the lower bound of the previous section we modify the absMAC specification. An easy way would be to relax the progress bound and output a rcv-event not only for messages sent by $G_{1-\varepsilon}$-neighbors, but for all message received (i.e. sent by any G_1 neighbor). This is problematic when considering randomized algorithms. In particular when computing e.g. overlay networks. It might happen that only $G_1 \setminus G_{1-\varepsilon}$-neighbors of a node v are chosen for the overlay due to the random event of low interference. This could of course be avoided by directly implementing the absMAC with respect to G_1 rather than $G_{1-\varepsilon}$, which in turn results in a $\Omega(n)$ lower bound for f_{prog} and f_{ack} (e.g. when all nodes are located at distance at least R_1 such that messages can only be received when exactly one node is sending). Later these overlay nodes might not be able to serve v. To avoid such a setting, we introduce an *approximate* progress bound into the absMAC specification, where we use a graph G and an *approximation* (or any subgraph) $\tilde{G}$ of G in which progress is measured.

In the next sections we show that this generalization of progress has three desirable properties, it 1) captures SINR behavior in the sense that we present an absMAC implementation in the SINR model that provides fast (approximate) progress, and 2) replaces (with minor assumptions and effects) the progress bound in the runtime-analysis of e.g. global single-message and multi-message broadcast in the MAC layer [25], and 3) does not affect the correctness of these algorithms. Therefore we consider this notion of approximate progress to be a good modification of the specification of abstract MAC layers with respect to the SINR model.

Definition 5.1 (Approximate progress)**.** *Let there be (reliable[3]) broadcast implemented with respect to a graph G and let $\tilde{G} := (V, \tilde{E})$ be a subgraph[4] of G. Consider a node i and assume that a $\tilde{G}$-neighbor of i is broadcasting a message. The approximate progress bound guarantees that a rcv event with a message originating in a G-neighbor occurs at node i within time f_{approg} with probability $1-\varepsilon_{approg}$. We say that approximate progress is implemented with respect to graphs G and (its approximation) $\tilde{G}$.*

We formalize this using the notation of [25]: Let β be a closed execution that ends at time t. Let I be the set of $\tilde{G}$-neighbors of j that have active bcasts at the end of β, where bcast$(m_i)_i$ is the bcast at i. Suppose that I is nonempty. Let I' be the set of G-neighbors of j that have active bcasts at the end of β. Suppose that no rcv$(m_i)_j$ event occurs in β, for any $i \in I'$. Define the following sets A and B of time-unbounded executions that extend β. Set A contains all executions in which no abort$(m_i)_i$ occurs for any $i \in I$. Set B contains all executions in which, by time $t + f_{approg}$, at least one of the following occurs: 1) an ack$(m_i)_i$ for every $i \in I$, 2) a rcv$(m_i)_j$ for some $i \in I'$, or 3) A rcv$_j$ for some message whose bcast occurs after β. If $\mathbb{P}_\beta[A] > 0$, then $\mathbb{P}_\beta[B|A] \geq 1-\varepsilon_{approg}$.

This notation is useful, as there are settings where it is not crucial that progress is made with respect to exactly G. Already progress in subgraph $\tilde{G}$ might yield good overall bounds for solving a problem on G especially when e.g. (depending on the problem at hand) $D_{\tilde{G}} \approx D_G$ or $\tilde{G}^2$. As we show in Theorem 7.1, in the global SMB and MMB algorithms of [25] local broadcast does not need

[3] The notation of approximate progress might later be extended to unreliable broadcast [14].

[4] Graph $\tilde{G}$ can be any subgraph of G but will typically be an approximation of G, which results in the name approximate progress. Later we consider graph $\tilde{G} := G_{1-2\varepsilon}$, which approximates $G := G_{1-\varepsilon}$ with respect to the SINR formula and Euclidean distances in the sense that it contains all, but the longest edges of G.

to be precise such that under some conditions progress can be replaced by approximate progress. In the SINR model one might choose, e.g., $G := G_{1-\varepsilon} \supseteq G_{1-2\varepsilon} =: \tilde{G}$, as we do. This choice captures that any $G_{1-\varepsilon}$-neighbor is almost a $G_{1-2\varepsilon}$-neighbor. In addition its signal has a similar strength when it arrives at the receiver and in reality might even be the same, as signal strengths can vary slightly. We discuss differences to the dual-graph model for unreliable communication of [14] in the full version of this paper [16].

6 Implementation of Fast Approximate Progress

We implement approximate progress with respect to $G := G_{1-\varepsilon}$ and $\tilde{G} := G_{1-2\varepsilon}$. In the full version of this paper [16] we show that the Decay method cannot achieve fast approximate progress in the SINR model. Therefore we describe a method different from Decay and obtain:

Theorem 6.1. *In the SINR model using the assumptions of Section 3, we implement approximate progress of an absMAC with respect to graphs $G_{1-\varepsilon}$ and its approximation $G_{1-2\varepsilon}$ with probability at least $1-\varepsilon_{approg}$ in time approximate progress of an absMAC with respect to graphs $G_{1-\varepsilon}$ and its approximation $G_{1-2\varepsilon}$ with probability at least $1-\varepsilon_{approg}$ in time*

$$f_{approg} = \mathcal{O}\left(\left(\log^{\alpha}(\Lambda) + \log^{*}\left(\frac{1}{\varepsilon_{approg}}\right)\right)\log(\Lambda)\log\left(\frac{1}{\varepsilon_{approg}}\right)\right).$$

6.1 Algorithm

The algorithm presented by [9] achieves w.h.p. global SMB in $G_{1-\varepsilon}$. We review this algorithm and show how to modify it to guarantee fast (probabilistic) approximate progress with respect to $G_{1-2\varepsilon}$. In the following, set S_1 contains all nodes with an ongoing broadcast. Set S_1 changes after each epoch depending on the algorithm using the absMAC.

High-level description of Algorithm 1 of [9] and the intuition behind it. This algorithm performs $D_{G_{1-\varepsilon}}$ many epochs. For $\Phi = \Theta(\log \Lambda)$, each epoch computes approximations $\tilde{H}_1, \tilde{H}_2, \ldots, \tilde{H}_\Phi$ of a sequence of constant degree graphs $H_1, H_2, \ldots, H_\Phi$. Each H_ϕ is defined based on nodes S_ϕ, s.t. when each node in S_ϕ transmits with probability $p \in (0, 1/2]$, the transmission corresponding to an edge of H_ϕ is successful with probability $\mu \in (0, p)$. Sets S_ϕ, $\phi \in [2, \Phi]$, are maximal independent sets in $\tilde{H}_{\phi-1}$ and the algorithm of [32] is simulated to compute such MIS (and uses a node's unique ID $\in$ poly n as an input). Each transmission during the computations of S_ϕ and $\tilde{H}_\phi$ is repeated $T := \Theta(\log n)$ to ensure w.h.p. correctness. Finally for each ϕ, all nodes S_ϕ transmit their bcast-message $\Theta(\log^{\alpha}(\Lambda)\log(n))$ times. Intuitively S_ϕ is a sparser version of $S_{\phi-1}$ and [9] shows that S_Φ contains only nodes that cannot communicate with each other. Using this and further insights they argue that for any node in $N_{G_{1-\varepsilon}}(S_1)$ there is a $\phi \in [1, \Phi]$ such that 1) there is a node $u \in S_\phi$ at distance at most $R_{1-\varepsilon}$, and 2) the density of S_ϕ is so low that interference from other nodes allows u_ϕ's message to reaches u w.h.p. (when the transmission is repeated sufficiently often). This shows that in each epoch all nodes in $N_{G_{1-\varepsilon}}(S_1)$ receive the (single!) bcast-message[5] w.h.p.. We provide more details in the full version of this paper [16].

Our modifications and motivation behind these changes: (I) We replace the inputs for the MIS algorithm. Instead of unique ID $\in$ poly n we use temporary labels $l_{i,\phi} \in [1, (\text{poly }\Lambda)/\varepsilon_{approg}]$.

[5] We denote by *bcast-message* any messages that contains information to be broadcast due to a bcast-event. By *messages*, we refer to messages sent for coordination among the nodes.

(II) We replace $T = \Theta(\log n)$ by $\Theta(\log(\Lambda/\varepsilon_{approg}))$ and reduce the number of repeated transmissions of bcast-messages from $\mathcal{O}(\log^{\alpha}(\Lambda)\log(n))$ to $\mathcal{O}(\log^{\alpha}(\Lambda)\log(1/\varepsilon_{approg}))$. (III) We rename the computed graphs from $\tilde{H}_\phi$ to $\tilde{\tilde{H}}_\phi$. (IV) We execute the algorithm with respect to $G_{1-2\varepsilon}$ instead of $G_{1-\varepsilon}$. If $\varepsilon_{approg} > n^{-c}$, these modifications reduce the runtime of an epoch, but also lower the probability of correctness. Therefore computed graphs are very unlikely to be global approximations of H_ϕ (and we change their name to $\tilde{\tilde{H}}_\phi$). Still, the parameters are chosen such that we can show that the probability of approximate progress is at least $1 - \varepsilon_{approg}$ as outlined in Section 6.2.

6.2 Outline of the Analysis

We analyze the effect of the two main modifications of the algorithm of [9] with respect to their analysis and put it into the context of approximate progress. More details of this careful analysis are provided in the full version of this paper [16].

First modification: non-unique labels in the MIS computation. We argue in the model of [32] the sets S_ϕ computed by our modified MIS-algorithm are independent sets in $\tilde{H}_{\phi-1}$. Furthermore, for any given node v, with probability $1 - \varepsilon_{approg}/3$, this set is maximal in a neighborhood around v "large enough" to ensure that this part of computations involved in approximate progress at node v is correct.

Second modification: fewer repetitions of transmissions. In the algorithm of [9] each node sends every bcast-message $\mathcal{O}(\log^{\alpha}(\Lambda)\log n)$ times, while we use only $\mathcal{O}(\log^{\alpha}(\Lambda)\log(1/\varepsilon_{approg}))$ repeated transmissions. This implies that [9] can assume that all communication is successful at any point w.h.p.. For large ε_{approg} we only have weak probability guarantees for success of communication. One side-effect is that with very high probability the computed graphs $\tilde{\tilde{H}}_\phi$ are not the desired global approximations of graphs H_ϕ. This in turn affects correctness of approximate progress and we need to analyze local and global implications caused by reducing the number of repeated transmissions.

1. **Global implications of unsuccessful transmissions:** Global interference might increase in the long term and we need to bound this. Unsuccessful transmissions during the computation of $\tilde{\tilde{H}}_\phi$ might remain undetected and cause that edges are missing in $\tilde{\tilde{H}}_\phi$. This event influences future computations of nearby nodes until the current epoch ends. Influenced nodes might cause additional global interference. In the full version of this paper [16] we bound the expected additional interference from these nodes. It turns out that T is chosen such that this interference can be tolerated in other parts of our proof and when transferring the analysis of [9]. After $\tilde{\tilde{H}}_\phi$ is computed, all transmissions are successful. They use the same schedule used to compute $\tilde{\tilde{H}}_\phi$.
2. **Local implications of unsuccessful transmissions:** Transmissions of messages need to be successful in all "large enough" neighborhoods of v in graphs $\tilde{\tilde{H}}_\phi$ to guarantee approximate progress at point v. These unsuccessful transmissions can only appear during the computation of $\tilde{\tilde{H}}^{\mu}_{p}[S_\phi]$ and while transmitting the bcast-message. Only if communication is locally successful, it is guaranteed that graph $\tilde{\tilde{H}}_\phi$ is an approximation of H_ϕ w.r.t. the above mentioned neighborhood of v, which is necessary in order to transfer the analysis of [9]. We analyze the probability that $\tilde{\tilde{H}}_\phi$ is locally an approximation in the full version of this paper [16]. Finally, approximate progress is made only if communication of bcast-messages succeeds locally.

6.3 Key Lemmas of the Analysis

Full proofs of the following lemmas appear in the full version of this paper [16]. We start by analyzing the effect of using (potentially) non-unique labels chosen uniformly at random $\in \left[1, \frac{\text{poly}\,\Lambda}{\varepsilon_{approg}}\right]$ in the modified MIS computation, which is the first difference to [9], as pointed out in Section 6.2.

Lemma 6.2. *Let $H = (V, E)$ be a constant degree growth-bounded graph and let $U \subseteq V$ be a set of nodes of size at most $\mathcal{O}(\Lambda^2)$. Consider an execution of our modification of the MIS-algorithm of [32] on H in the CONGEST model using random labels $\in \left[1, \frac{poly\,\Lambda}{\varepsilon_{approg}}\right]$. Then the set of nodes in state dominator is 1) an independent set, and 2) with probability at least $1 - \frac{\varepsilon_{approg}}{3\Phi}$ this set is maximal with respect to $N_{H, c \cdot 4^{\Phi} \cdot \log^*(\Lambda/\varepsilon_{approg})}(U)$, the $c \cdot 4^{\Phi} \cdot \log^*(\Lambda/\varepsilon_{approg})$-neighborhood of U in H.*

We analyze Case 1 pointed out in Section 6.2, i.e. we bound the global interference from nodes with undetectable unsuccessful transmissions.

Definition 6.3 (Set W of nodes with wrong neighborhoods (due to unsuccessful transmissions))**.** *Denote by $W \subseteq S_1$ the set of all those nodes v such that for at least one $\phi \in \{1, \cdots, \Phi\}$ it is not the case that $N_{H_p^{\mu}[S_\phi]}(v) \subseteq N_{\tilde{\tilde{H}}_p^{\mu}[S_\phi]}(v) \subseteq N_{H_p^{\mu(1-\gamma)}[S_\phi]}(v)$, i.e. v's direct neighborhood does not $(1-\gamma)$-approximate $N_{H_p^{\mu}[S_\phi]}(v)$.*

Lemma 6.4. *Given point i in space, the expected total additional interference $I_W(i)$ that point i receives from all nodes in W at any given time is less than $\left(\frac{\varepsilon_{approg}}{\Lambda}\right)^{\Theta(1)}$.*

The full paper [16] introduces the notion of a successful epoch at a given point i in a formal way. In summary, given point i, an epoch is successful at point i, if three properties are satisfied: 1) all computations of graphs $\tilde{\tilde{H}}_p^{\mu}[S_1], \ldots, \tilde{\tilde{H}}_p^{\mu}[S_{\Phi-1}]$, and 2) the corresponding sets S_ϕ are correct within a certain area around i and 3) some message was transmitted to i successfully at some point. We show that

Lemma 6.5. *Given set S_1 and a node i and let there be a $G_{1-2\varepsilon}$-neighbor of i with an ongoing broadcast event $bcast(m)_j$. Assume Properties 1 and 2 of the definition [16] of a successful epoch at point i are satisfied. Then there is a phase $\phi' \in \{1, \ldots, \Phi\}$ such that in phase ϕ node i receives a bcast-message from a $G_{1-\varepsilon}$-neighbor of i with probability $1 - \varepsilon_{approg}/3$.*

These are the key lemmas that are used together with a bound on the runtime of an epoch to derive Theorem 6.1.

7 Application: Improved Network-Wide Broadcast

In the full version of this paper [16] we implement the probabilistic absMAC of [25] in a formal way using Theorems 4.1 and 6.1 and the corresponding algorithms. We combine this absMAC implementation with algorithms from [25] for global broadcast in this absMAC. In Theorem 7.1 we argue that we can replace f_{prog} and ε_{prog} in the relevant Theorems of [25] by f_{approg} and ε_{approg} under certain conditions and state the effect that this replacement has on other parameters of the runtime.

Theorem 7.1. *Let G be a graph in which local broadcast is available via the probabilistic absMAC of [25]. Let $\tilde{G}$ be the graph in which approximate progress is measured and let the vertex sets of the connected components of $\tilde{G}$ and G be the same. Then one can replace $f_{prog}, \varepsilon_{prog}$ and D_G in Theorems 7.7 and 8.20 of [25] concerning their global SMB and MMB algorithms by $f_{approg}, \varepsilon_{approg}$ and $D_{\tilde{G}}$.*

In the algorithms of [25], once a node i receives a message, node i broadcasts the message if it did not broadcast it before. The result of global broadcast is independent of whether a message was received due to transmission from a $\tilde{G}$-neighbor or a G-neighbor as long as the components of $\tilde{G}$ and G are the same. Only the runtime changes. In time f_{prog} with probability $1 - \varepsilon_{prog}$ a message is received by a node v when a $G_{1-\varepsilon}$-neighbor of v is sending. Therefore the runtime presented in [25] depends on $D_{G_{1-\varepsilon}}$. Compared to this, with probability $1 - \varepsilon_{approg}$ in time f_{approg} a message arrives when a $G_{1-2\varepsilon}$-neighbor is sending. Therefore $D_{G_{1-\varepsilon}}$ needs to be replaced by $D_{G_{1-2\varepsilon}}$, see the full version of this paper [16] for details.

Theorem 7.2. *Consider the SINR model using the model assumptions stated in Section 3. We present an algorithm that performs global SMB in graph $G_{1-\varepsilon}$ with probability at least $1 - \varepsilon_{SMB}$ in time $\mathcal{O}\left(\left(D_{G_{1-2\varepsilon}} + \log\left(\frac{n}{\varepsilon_{SMB}}\right)\right) \cdot \log^{\alpha+1}(\Lambda)\right)$.*

The second algorithm presented in the proof performs global MMB in graph $G_{1-\varepsilon}$ with probability at least $1 - \varepsilon_{MMB}$ in time $\mathcal{O}\left(D_{G_{1-2\varepsilon}} \log^{\alpha+1}(\Lambda) \quad + \quad k'\left(\Delta_{G_{1-\varepsilon}} + polylog\left(\frac{nk\Lambda}{\varepsilon_{MMB}}\right)\right) \log\left(\frac{nk}{\varepsilon_{MMB}}\right)\right)$

The proof applies our Theorems 4.1, 6.1 and 7.1 to Theorems 7.7 and 8.20 of [25], see the full version of this paper [16] for details.

Acknowledgments: We thank Sebastian Daum, Mohsen Ghaffari, Fabian Kuhn and Calvin Newport for answering questions concerning their earlier work and helpful discussions. In particular we thank Erez Kantor for many helpful discussions—especially in the early stages of this work.

References

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

[2] M. Andrews and M. Dinitz. Maximizing capacity in arbitrary wireless networks in the SINR model: Complexity and game theory. In *Proc. of the 28th International Conference on Computer Communications, INFOCOM, Rio de Janeiro*, pages 1332–1340. IEEE, 2009.

[3] R. Bar-Yehuda, A. Israeli, and A. Itai. Multiple communication in multihop radio networks. *SIAM J. Comput.*, 22(4):875–887, 1993.

[4] M. H. L. Bodlaender, M. M. Halldórsson, and P. Mitra. Connectivity and aggregation in multihop wireless networks. In *ACM Symposium on Principles of Distributed Computing, PODC '13, Montreal, QC, Canada, July 22-24, 2013*, pages 355–364, 2013.

[5] I. Chlamtac and S. Kutten. On broadcasting in radio networks–problem analysis and protocol design. *IEEE Transactions on Communications*, 33(12):1240–1246, 1985.

[6] A. Cornejo, N. Lynch, S. Viqar, and J. L. Welch. Neighbor discovery in mobile ad hoc networks using an abstract mac layer. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 1460–1467. IEEE, 2009.

[7] A. Cornejo, S. Viqar, and J. L. Welch. Reliable neighbor discovery for mobile ad hoc networks. *Ad Hoc Networks*, 12:259–277, 2014.

[8] A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. *J. Algorithms*, 60(2):115–143, 2006.

[9] S. Daum, S. Gilbert, F. Kuhn, and C. C. Newport. Broadcast in the ad hoc SINR model. In *Distributed Computing - 27th International Symposium, DISC 2013, Jerusalem, Israel, October 14-18, 2013. Proceedings*, pages 358–372, 2013.

[10] M. Dinitz. Distributed algorithms for approximating wireless network capacity. In *INFOCOM 2010. 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 15-19 March 2010, San Diego, CA, USA*, pages 1397–1405, 2010.

[11] G. Even, Y. Matsri, and M. Medina. Multi-hop routing and scheduling in wireless networks in the SINR model. In *Algorithms for Sensor Systems - 7th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, ALGOSENSORS 2011, Saarbrücken, Germany, September 8-9, 2011, Revised Selected Papers*, pages 202–214, 2011.

[12] M. Ghaffari, B. Haeupler, and M. Khabbazian. A bound on the throughput of radio networks. *CoRR*, abs/1302.0264, 2013.

[13] M. Ghaffari, B. Haeupler, and M. Khabbazian. Randomized broadcast in radio networks with collision detection. In *ACM Symposium on Principles of Distributed Computing, PODC '13, Montreal, QC, Canada, July 22-24, 2013*, pages 325–334, 2013.

[14] M. Ghaffari, E. Kantor, N. Lynch, and C. Newport. Multi-message broadcast with abstract MAC layers and unreliable links. In *Proceedings of the 33rd Annual ACM Symposium on Principles of Distributed Computing, PODC 2014, Paris, France, July 15-18, 2014*, pages 56–65, extended version available at http://arxiv.org/abs/1405.1671, 2014.

[15] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local broadcasting in the physical interference model. In *DIALM-POMC*, pages 35–44, 2008.

[16] M. M. Halldorsson, S. Holzer, and N. Lynch. A local broadcast layer for the sinr network model. *arXiv preprint arXiv:1505.tba*, 2015.

[17] M. M. Halldórsson and P. Mitra. Towards tight bounds for local broadcasting. In *FOMC'12, The Eighth ACM International Workshop on Foundations of Mobile Computing (part of PODC 2012), Funchal, Portugal, July 19, 2012, Proceedings*, page 2, 2012.

[18] N. Hobbs, Y. Wang, Q.-S. Hua, D. Yu, and F. C. M. Lau. Deterministic distributed data aggregation under the SINR model. In M. Agrawal, S. B. Cooper, and A. Li, editors, *TAMC*, volume 7287 of *Lecture Notes in Computer Science*, pages 385–399. Springer, Heidelberg, 2012.

[19] T. Jurdzinski, D. R. Kowalski, M. Rozanski, and G. Stachowiak. Distributed randomized broadcasting in wireless networks under the SINR model. In *Distributed Computing*, pages 373–387. 2013.

[20] T. Jurdzinski, D. R. Kowalski, M. Rozanski, and G. Stachowiak. On the impact of geometry on ad hoc communication in wireless networks. In M. M. Halldórsson and S. Dolev, editors, *Proceedings of the Thirty-Third Annual ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 357–366, 2014.

[21] T. Jurdzinski, D. R. Kowalski, and G. Stachowiak. Distributed deterministic broadcasting in wireless networks of weak devices. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, pages 632–644, 2013.

[22] T. Kesselheim. A constant-factor approximation for wireless capacity maximization with power control in the SINR model. In D. Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1549–1559, 2011.

[23] T. Kesselheim and B. Vöcking. Distributed contention resolution in wireless networks. In N. A. Lynch and A. A. Shvartsman, editors, *Distributed Computing (proceedings of DISC)*, volume 6343 of *Lecture Notes in Computer Science*, pages 163–178. Springer, Heidelberg, 2010.

[24] M. Khabbazian and D. R. Kowalski. Time-efficient randomized multiple-message broadcast in radio networks. In C. Gavoille and P. Fraigniaud, editors, *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011, San Jose, CA, USA, June 6-8, 2011*, pages 373–380, 2011.

[25] M. Khabbazian, D. R. Kowalski, F. Kuhn, and N. A. Lynch. Decomposing broadcast algorithms using abstract MAC layers. *Ad Hoc Networks*, 12:219–242, 2014.

[26] M. Khabbazian, F. Kuhn, N. A. Lynch, M. Médard, and A. ParandehGheibi. MAC design for analog network coding. In A. Chaintreau and D. R. Kowalski, editors, *FOMC'11, The Seventh ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing (part of FCRC 2011), San Jose, CA, USA, June 9, 2011, Proceedings*, pages 42–51, 2011.

[27] D. R. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. *Distributed Computing*, 18(1):43–57, 2005.

[28] F. Kuhn, N. A. Lynch, and C. C. Newport. The abstract MAC layer. *Distributed Computing*, 24(3-4):187–206, 2011.

[29] E. Kushilevitz and Y. Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM J. on Computing*, 27:702–712, 1998.

[30] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *Proceedings of the 25th annual IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2006, Barcelona, Catalunya, Spain, 23-29 April 2006*, pages 1–13, 2006.

[31] C. C. Newport. Consensus with an abstract MAC layer. In M. M. Halldórsson and S. Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 66–75, 2014.

[32] J. Schneider and R. Wattenhofer. A log-star distributed maximal independent set algorithm for growth-bounded graphs. In R. A. Bazzi and B. Patt-Shamir, editors, *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008, Toronto, Canada, August 18-21, 2008*, pages 35–44, 2008.

[33] D. Yu, Q. Hua, Y. Wang, and F. C. M. Lau. An o(log n) distributed approximation algorithm for local broadcasting in unstructured wireless networks. In *IEEE 8th International Conference on Distributed Computing in Sensor Systems, DCOSS 2012, Hangzhou, China, 16-18 May, 2012*, pages 132–139, 2012.

[34] D. Yu, Q.-S. Hua, Y. Wang, H. Tan, and F. C. M. Lau. Distributed multiple-message broadcast in wireless ad-hoc networks under the SINR model. In G. Even and M. M. Halldórsson, editors, *Structural Information and Communication Complexity - 19th International Colloquium,*

SIROCCO 2012, Reykjavik, Iceland, June 30-July 2, 2012, Revised Selected Papers, volume 7355 of *Lecture Notes in Computer Science*, pages 111–122. Springer, Heidelberg, 2012.

[35] D. Yu, Q.-S. Hua, Y. Wang, J. Yu, and F. C. M. Lau. Efficient distributed multiple-message broadcasting in unstructured wireless networks. In *Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14-19, 2013*, pages 2427–2435, 2013.