Bounds on Contention Management in Radio Networks

Mohsen Ghaffari[*], Bernhard Haeupler[*], Nancy Lynch[*] and Calvin Newport[**]

[*]Computer Science and Artificial Intelligence Lab, MIT
{ghaffari, haeupler, lynch}@csail.mit.edu

[**]Department of Computer Science, Georgetown University
cnewport@cs.georgetown.edu

Abstract. The local broadcast problem assumes that processes in a wireless network are provided messages, one by one, that must be delivered to their neighbors. In this paper, we prove tight bounds for this problem in two well-studied wireless network models: the *classical* model, in which links are reliable and collisions consistent, and the more recent *dual graph* model, which introduces unreliable edges. Our results prove that the *Decay* strategy, commonly used for local broadcast in the classical setting, is optimal. They also establish a separation between the two models, proving that the dual graph setting is strictly harder than the classical setting, with respect to this primitive.

1 Introduction

At the core of every wireless network algorithm is the need to manage contention on the shared medium. In the theory community, this challenge is abstracted as the *local broadcast problem*, in which processes are given messages, one by one, that must be delivered to their neighbors.

This problem has been studied in multiple wireless network models. The most common such model is the *classical* model, introduced by Chlamatac and Kutten [8], in which links are reliable and concurrent broadcasts by neighbors always generate collisions. The dominant local broadcast strategy in this model is the *Decay* routine introduced by Bar-Yehuda et al. [9]. In this strategy, nodes cycle through an exponential distribution of broadcast probabilities with the hope that one will be appropriate for the current level of contention (e.g., [9, 11–17, 22]). To solve local broadcast with high probability (with respect to the network size n), the *Decay* strategy requires $O(\Delta \log n)$ rounds, where Δ is the maximum contention in the network (which is at most the maximum degree in the network topology). It has remained an open question whether this bound can be improved to $O(\Delta + \text{polylog}(n))$. In this paper, we resolve this open question by proving the *Decay* bound optimal (notice, throughout this paper, when we call an upper bound "optimal" or a lower bound "matching," we mean within poly-$\log\log$ factors). This result also proves for the first time that existing constructions of *ad hoc selective families* [15, 16]—a type of combinatorial object used in wireless network algorithms—are optimal.

We then turn our attention to the more recent *dual graph* wireless network model introduced by Kuhn et al. [18, 20, 22, 25]. This model generalizes the classical model by

	Classical Model	Dual Graph Model
Ack. Upper	$O(\varDelta \log n)^{**}$	$O(\varDelta' \log n)^{*}$
Ack. Lower	$\varOmega\left(\frac{\varDelta \log n}{\log^2 \log n}\right)^{*}$	$\varOmega\left(\frac{\varDelta' \log n}{\log^2 \log n}\right)^{*}$
Prog. Upper	$O(\log \varDelta \log n)$	$O(\min\{k \log k \log n, \varDelta' \log n\})^{*}$
Prog. Lower	$\varOmega(\log \varDelta \log n)^{**}$	$\varOmega\left(\frac{\varDelta' \log n}{\log^2 \log n}\right)^{*}$

Fig. 1. A summary of our results for *acknowledgment* and *progress* for the local broadcast problem. Results that are new, or significant improvements over the previously best known result, are marked with an "*" while a "**" marks results that where obtained from prior work via minor tweaks.

allowing some edges in the communication graph to be unreliable. It was motivated by the observation that real wireless networks include links of dynamic quality (see [22] for more extensive discussion). We provide tight solutions to the local broadcast problem in this setting, using algorithms based on the *Decay* strategy. Our tight bounds in the dual graph model are larger (worse) than our tight time bounds for the classical model, formalizing a separation between the two settings (see Figure 1 and the discussion below for result details). We conclude by proving another separation: in the classical model there is no significant difference in power between centralized and distributed local broadcast algorithms, while in the dual graph model the gap is exponential.

These separation results are important because most wireless network algorithm analysis relies on the correctness of the underlying contention management strategy. By proving that the dual graph model is strictly harder with respect to local broadcast, we have established that an algorithm proved correct in the classical model will not necessarily remain correct or might loose its efficiency in the more general (and more realistic) dual graph model.

To summarize: This paper provides an essentially complete characterization of the local broadcast problem in the well-studied classical and dual graph wireless network models. In doing so, we: (1) answer the long-standing open question regarding the optimality of *Decay* in the classical model; (2) provide a variant of Decay and prove it optimal for the local broadcast problem in the dual graph model; and (3) formalize the separation between these two models, with respect to local broadcast.

Result Details: As mentioned, the *local broadcast* problem assumes processes are provided messages, one by one, which should be delivered to their neighbors in the communication graph. Increasingly, local broadcast solutions are being studied separately from the higher level problems that use them, improving the composability of solutions; e.g., [18, 21, 23, 24]. Much of the older theory work in the wireless setting, however, mixes the local broadcast logic with the logic of the higher-level problem being solved; e.g., [9,11–17,22]. This previous work can be seen as implicitly solving local broadcast.

The efficiency of a local broadcast algorithm is characterized by two metrics: (1) an *acknowledgment bound*, which measures the time for a sender process (a process that has a message for broadcast) to deliver its message to all of its neighbors; and (2) a *progress bound*, which measures the time for a receiver process (a process that has a sender neighbor) to receive at least one message [1]. The acknowledgment bound is obviously interesting; the progress bound has also been shown to be critical for analyzing algorithms for many problems, e.g., global broadcast [18] where the reception of *any* message is normally sufficient to advance the algorithm. The progress bound was first introduced and explicitly specified in [18,23] but it was implicitly used already in many previous works [9,11–14,17]. Both acknowledgment and progress bounds typically depend on two parameters, the maximum contention $\varDelta$ and the network size n. In the dual graph model, an additional measure of maximum contention, $\varDelta'$, is introduced to measure contention in the unreliable communication link graph, which is typically denser than the reliable link graph. In our progress result for the dual graph model, we also introduce k to capture the *actual* amount of contention relevant to a specific message. These bounds are usually required to hold with high probability.

Our upper and lower bound results for the local broadcast problem in the classical and dual graph models are summarized in Figure 1. Here we highlight three key points regarding these results. First, in both models, the upper bounds are within $O(\log^2 \log n)$ of the lower bounds. Second, we show that $\varOmega(\frac{\varDelta \log n}{\log^2 \log n})$ rounds are necessary for acknowledgment in the classical model. This answers in the negative the open question of whether a $O(\varDelta + \text{polylog}(n))$ solution is possible. Third, the separation between the classical and dual graph models occurs with respect to the progress bound, where the tight bound for the classical model is *logarithmic* with respect to contention, while in the dual graph model it is *linear*—an exponential gap. Finally, in addition to the results described in Figure 1, we also prove the following additional separation between the two models: in the dual graph model, the gap in progress between distributed and centralized local broadcast algorithms is (at least) linear in the maximum contention $\varDelta'$, whereas no such gap exists in the classical model.

Before starting the technical sections, we remark that due to space considerations, the full proofs are omitted from the conference version and can be found in [27].

2 Model

To study the local broadcast problem in synchronous multi-hop radio networks, we use two models, namely the *classical radio network model* (also known as the radio network model) and the *dual graph model*. The former model assumes that all connections in the network are reliable and it has been extensively studied since 1980s [8–18, 18, 23]. On the other hand, the latter model is a more general model, introduced more recently in 2009 [18–20], which includes the possibility of unreliable edges. Since the former model is simply a special case of the latter, we use dual graph model for explaining the model and the problem statement. However, in places where we want to emphasize on a result in the classical model, we focus on the classical model and explain how the result specializes for this specific case.

In the dual graph model, radio networks have some reliable and potentially some unreliable links. Fix some $n \geq 1$. We define a network (G, G') to consist of two undirected graphs, $G = (V, E)$ and $G' = (V, E')$, where V is a set of n wireless nodes

[1] Note that with respect to these definitions, a process can be both a sender and a receiver, simultaneously.

and $E \subseteq E'$, where intuitively set E is the set of reliable edges while E' is the set of all edges, both reliable and unreliable. In the classical radio network model, there is no unreliable edge and thus, we simply have $G = G'$, i.e., $E = E'$.

We define an algorithm $\mathcal{A}$ to be a collection of n randomized processes, described by probabilistic automata. An execution of $\mathcal{A}$ in network (G, G') proceeds as follows: first, we fix a bijection $proc$ from V to $\mathcal{A}$. This bijection assigns processes to graph nodes. We assume this bijection is defined by an adversary and is not known to the processes. We do not, however, assume that the definition of (G, G') is unknown to the processes (in many real world settings it is reasonable to assume that devices can make some assumptions about the structure of their network). In this study, to strengthen our results, our upper bounds make no assumptions about (G, G') beyond bounds on maximum contention and polynomial bounds on size of the network, while our lower bounds allow full knowledge of the network graph.

An execution proceeds in synchronous rounds $1, 2, ...,$ with all processes starting in the first round. At the beginning of each round r, every process $proc(u), u \in V$ first receives inputs (if any) from the environment. It then decides whether or not to transmit a message and which message to send. Next, the adversary chooses a *reach set* that consists of E and some subset, potentially empty, of edges in $E' - E$. Note that in the classical model, set $E' - E$ is empty and therefore, the reach set is already determined. This set describes the links that will behave reliably in this round. We assume that the adversary has full knowledge of the state of the network while choosing this reach set. For a process v, let $B_{v,r}$ be the set all graph nodes u such that, $proc(u)$ broadcasts in r and $\{u, v\}$ is in the reach set for this round. What $proc(v)$ receives in this round is determined as follows. If $proc(v)$ broadcasts in r, then it receives only its own message. If $proc(v)$ does not broadcast, there are two cases: (1) if $|B_{v,r}| = 0$ or $|B_{v,r}| > 1$, then $proc(v)$ receives $\bot$ (indicating *silence*); (2) if $|B_{v,r}| = 1$, then $proc(v)$ receives the message sent by $proc(u)$, where u is the single node in $B_{v,r}$. That is, we assume processes cannot send and receive simultaneously, and also, there is no collision detection in this model. However, to strengthen our results, we note that our lower bound results hold even in the model with collision detection, i.e., where process v receives a special collision indicator message $\top$ in case $|B_{v,r}| > 1$. After processes receive their messages, they generate outputs (if any) to pass back to the environment.

Distributed vs. Centralized Algorithms: The model defined above describes distributed algorithms in a radio network setting. To strengthen our results, in some of our lower bounds we consider the stronger model of *centralized* algorithms. We formally define a centralized algorithm to be defined the same as the distributed algorithms above, but with the following two modifications: (1) the processes are given $proc$ at the beginning of the execution; and (2) the processes can make use of the current state and inputs of *all* processes in the network when making decisions about their behavior.

Notation & Assumptions: For each $u \in V$, the notations $N_G(u)$ and $N_{G'}(u)$ describe, respectively, the neighbors of u in G and G'. Also, we define $N^+_G(u) = N_G(u) \cup \{u\}$ and $N^+_{G'}(u) = N_{G'}(u) \cup \{u\}$. For any algorithm $\mathcal{A}$, we assume that each process $\mathcal{A}$ has a unique identifier. To simplify notation, we assume the identifiers are from $\{1, ..., n\}$. We remark that our lower bounds hold even with such strong identifiers, whereas for the upper bounds, we just need the identifiers of different processes to be different. Let $id(u), u \in V$ describe the id of process $proc(u)$. For simplicity, throughout this paper we often use the notation *process* u, or sometimes just u, for some $u \in V$, to refer to $proc(u)$ in the execution in question. Similarly, we sometimes use *process* i, or sometimes just i, for some $i \in \{1, ..., n\}$, to refer to the process with id i. We sometimes use the notation $[i, i']$, for integers $i' \geq i$, to indicate the sequence $\{i, ..., i'\}$, and the notation $[i]$ for integer i to indicate $[1, i]$. Throughout, we use the the notation *w.h.p.* (*with high probability*) to indicate a probability at least $1 - \frac{1}{n}$. Also, unless specified, all logarithms are natural log. Moreover, we ignore the integral part signs whenever it is clear that omitting them does not effect the calculations more than a change in constants.

3 Problem

Our first step in formalizing the local broadcast problem is to fix the input/output interface between the *local broadcast module* (automaton) of a process and the higher layers at that process. In this interface, there are three actions as follows: (1) $bcast(m)_v$, an input action that provides the local broadcast module at process v with message m that has to be broadcast over v's local neighborhood, (2) $ack(m)_v$, an output action that the local broadcast module at v performs to inform the higher layer that the message m was delivered to all neighbors of v successfully, (3) $rcv(m)_u$, an output action that local broadcast module at u performs to transfer the message m, received through the radio channel, to higher layers. To simplify definitions going forward, we assume w.l.o.g. that every $bcast(m)$ input in a given execution is for a unique m. We also need to restrict the behavior of the environment to generate $bcast$ inputs in a *well-formed* manner, which we define as strict alternation between $bcast$ inputs and corresponding ack outputs at each process. In more detail, for every execution and every process u, the environment generates a $bcast(m)_u$ input only under two conditions: (1) it is the first input to u in the execution; or (2) the last input or non-rcv output action at u was an ack.

We say an algorithm *solves the local broadcast problem* if and only if in every execution, we have the following three properties: (1) for every process u, for each $bcast(m)_u$ input, u eventually responds with a single $ack(m)_u$ output, and these are the only ack outputs generated by u; (2) for each process v, for each message m, v outputs $rcv(m)_v$ at most once and if v generates a $rcv(m)_v$ output in round r, then there is a neighbor $u \in N_{G'}(v)$ such that following conditions hold: u received a $bcast(m)_u$ input before round r and has not output $ack(m)_u$ before round r (3) for each process u, if u receives $bcast(m)_u$ in round r and respond with $ack(m)_u$ in round $r' \geq r$, then w.h.p.: $\forall v \in N_G(u)$, v generates output $rcv(m)_v$ within the round interval $[r, r']$. We call an algorithm that solves the local broadcast problem a *local broadcast algorithm.*

Time Bounds: We measure the performance of a local broadcast algorithm with respect to the two bounds first formalized in [18]: *acknowledgment* (the worst case bound on the time between a $bcast(m)_u$ and the corresponding $ack(m)_u$), and *progress* (informally speaking the worst case bound on the time for a process to receive at least one message when it has one or more G neighbors with messages to send). The first bound represents standard ways of measuring the performance of local communication. The progress bound is crucial for obtaining tight performance bounds in certain classes of applications. See [18, 23] for examples of places where progress bound proves crucial explicitly. Also, [9, 11–14, 17] use the progress bound implicitly throughout their analysis.

In more detail, a local broadcast algorithm has two *delay functions* which describe these delay bounds as a function of the relevant contention: f_{ack}, and f_{prog}, respectively. In other words, every local broadcast algorithm can be characterized by these two functions which must satisfy properties we define below. Before getting to these properties, however, we first present a few helper definitions that we use to describe local contention during a given round interval. The following are defined with respect to a fixed execution. (1) We say a process u is *active* in round r, or, alternatively, *active with* m, iff it received a $bcast(m)_u$ output in a round $\leq r$ and it has not yet generated an $ack(m)_u$ output in response. We furthermore call a message m active in round r if there is a process that is active with it in round r. (2) For process u and round r, contention $c(u, r)$ equals the number of active G' neighbors of u in r. Similarly, for every $r' \geq r$, $c(u, r, r') = max_{r'' \in [r, r']}\{c(u, r'')\}$. (3) For process v and rounds $r' \geq r$, $c'(v, r, r') = max_{u \in N_G(v)}\{c(u, r, r')\}$. We can now formalize the properties our delay functions, specified for a local broadcast algorithm, must satisfy for any execution:

1. *Acknowledgment bound:* Suppose process v receives a $bcast(m)_v$ input in round r. Then, if $r' \geq r$ is the round in which process v generates corresponding output $ack(m)_v$, then with high probability we have $r' - r \leq f_{ack}(c'(v, r, r'))$.
2. *Progress bound:* For any pair of rounds r and $r' \geq r$, and process u, if $r' - r > f_{prog}(c(u, r, r'))$ and there exists a neighbor $v \in N_G(u)$ that is active throughout the entire interval $[r, r']$, then with high probability, u generates a $rcv(m)_u$ output in a round $r'' \leq r'$ for a message m that was active at some round within $[r, r']$.

We use notation Δ' (or Δ for the classical model) to denote the maximum contention over all processes.[2] In our upper bound results, we assume that processes are provided with upper bounds on contention that are within a constant factor of Δ' (or Δ for the classical model). Also, for the sake of concision, in the results that follow, we sometimes use the terminology "*has an acknowledgment bound of*" (resp. *progress bound*) to indicate "*specifies the delay function* f_{ack}" (resp. f_{prog}). For example, instead of saying "the algorithm specifies delay function $f_{ack}(k) = O(k)$," we might instead say "the algorithm has an acknowledgment bound of $O(k)$."

Simplified One-Shot Setting for Lower Bounds: The local broadcast problem as just described assumes that processes can keep receiving messages as input forever and in an arbitrary asynchronous way. This describes the practical reality of contention management, which is an on going process. All our algorithms work in this general setting. For our lower bounds, we use a setting in which we restrict the environment to only issue broadcast requests at the beginning of round one. We call this the *one-shot setting*. Also, in most of our lower bounds, we consider, G and G' to be bipartite graphs,

[2] Note that since the maximum degree in the graph is an upper bound on the maximum contention, this notation is consistent with prior work, see e.g. [18, 23, 24].

where nodes of one part are called *senders* and they receive broadcast inputs, and nodes of the other part are called *receivers*, and each have a sender neighbor. In this setting, when referring to contention $c(u)$, we furthermore mean $c(u, 1)$. Note that in this setting, for any r, r', $c(u, [r, r'])$ is less than or equal to $c(u, 1)$. The same holds for $c'(u)$. Also, in these bipartite networks, the maximum G'-degree (or G-degree in the classical model) of the receiver nodes provides an upper bound on the maximum contention Δ' (or Δ in the classical model). When talking about these networks, and when it is clear from the context, we sometimes use the phrase *maximum receiver degree* instead of the maximum contention.

4 Related Work

Chlamatac and Kutten [8] were the first to introduce the classical radio network model. Bar-Yehuda et al. [9] studied the theoretical problem of local broadcast in synchronized multi-hop radio networks as a submodule for the broader goal of global broadcast. For this, they introduced *Decay* procedure, a randomized distributed procedure that solves the local broadcast problem. Since then, this procedure has been the standard method for resolving contention in wireless networks (see *e.g.* [17,18,23,24]). In this paper, we prove that a slightly modified version of Decay protocol achieves optimal progress and acknowledgment bounds in both the classical radio network model and the dual graph model. A summary of these time bounds is presented in Figure 1.

Deterministic solutions to the local broadcast problem are typically based on combinatorial objects called *Selective Families*, see *e.g.* [12]- [16]. Clementi et al. [14] construct (n, k)-selective families of size $O(k \log n)$ ([14, Theorem 1.3]) and show that this bound is tight for these selective families ([14, Theorem 1.4]). Using these selective families, one can get local broadcast algorithms that have progress bound of $O(\Delta \log n)$, in the classical model. These families do not provide any local broadcast algorithm in the dual graph model. Also, in the same paper, the authors construct (n, k)-strongly-selective families of size $O(k^2 \log n)$ ([14, Theorem 1.5]). They also show (in [14, Theorem 1.6]) that this bound is also, in principle, tight for selective families when $k \leq \sqrt{2n} - 1$. Using these strongly selective families, one can get local broadcast algorithms with acknowledgment bound of $O(\Delta^2 \log n)$ in the classical model and also, with acknowledgment bound of $f_{ack}(k) = O((\Delta')^2 \log n)$ in the dual graph model. As can be seen from our results (summarized in Figure 1), all three of the above time bounds are far from the optimal bounds of the local broadcast problem. This shows that when randomized solutions are admissible, solutions based on these notions of selective families are not optimal.

In [15], Clementi et al. introduce a new type of selective families called Ad-Hoc Selective Families which provide new solutions for the local broadcast problem, if we assume that processes know the network. Clementi et al. show in [15, Theorem 1] that for any given collection $\mathcal{F}$ of subsets of set $[n]$, each with size in range $[\Delta_{min}, \Delta_{max}]$, there exists an ad-hoc selective family of size $O((1 + \log(\Delta_{max}/\Delta_{min})) \cdot \log |F|)$. This, under the assumption of processes knowing the network, translates to a deterministic local broadcast algorithm with progress bound of $O(\log \Delta \, \log n)$, in the classical model. This family do not yield any broadcast algorithms for the dual graph model.

Also, in [16], Clementi et al. show that for any given collection $\mathcal{F}$ of subsets of set $[n]$, each of size at most Δ, there exists a Strongly-Selective version of Ad-Hoc Selective Families that has size $O(\Delta \log |F|)$ (without using the name ad hoc). This result shows that, again under the assumption of knowledge of the network, there exists a deterministic local broadcast algorithms with acknowledgment bounds of $O(\Delta \log n)$ and $O(\Delta' \log n)$, respectively in the classical and dual graph models. Our lower bounds for the classical model show that both of the above upper bounds on the size of these objects are tight.

5 Upper Bounds for Both Classical and Dual Graph Models

In this section, we show that by slight modifications to Decay protocol, we can achieve upper bounds that match the lower bounds that we present in the next sections. Due to space considerations, the details of the related algorithms are omitted from the conference version and can be found in [27].

Theorem 5.1. *In the classical model, there exists a distributed local broadcast algorithm that gives acknowledgment bound of $f_{ack}(k) = O(\Delta \log n)$ and progress bound of $f_{prog}(k) = O(\log \Delta \log n)$.*

Theorem 5.2. *There exists a distributed local broadcast algorithm that, in the classical model, gives bounds of $f_{ack}(k) = O(\Delta \log n)$ and $f_{prog}(k) = O(\log \Delta \log n)$, and in the dual graph model, gives bounds of $f_{ack}(k) = O(\Delta' \log n)$ and $f_{prog}(k) = O(\min\{k \log \Delta' \log n, \Delta' \log n\})$.*

Theorem 5.3. *In the dual graph model, there exists a distributed local broadcast algorithm that gives acknowledgment bound of $f_{ack}(k) = O(\Delta' \log n)$ and progress bound of $f_{prog}(k) = O(\min\{k \log k \log n, \Delta' \log n\})$.*

6 Lower Bounds in the Classical Radio Broadcast Model

In this section, we focus on the problem of local broadcast in the classical model and present lower bounds for both progress and acknowledgment times. We emphasize that all these lower bounds are presented for centralized algorithms and also, in the model where processes are provided with a collision detection mechanism. Note that these points only strengthen these results. These lower bounds prove, for the first time, that the optimized decay protocol, as presented in the previous section, is optimal with respect to progress and acknowledgment times in the classical model. These lower bounds also show that the existing constructions of Ad Hoc Selective Families are optimal. Moreover, in future sections, we use the lower bound on the acknowledgment time in the classical model that we present here as a basis to derive lower bounds for progress and acknowledgment times in the dual graph model.

6.1 Progress Time Lower Bound

In this section, we remark that following the proof of the lower bound of Alon et al. [10] on the time needed for global broadcast of one message in radio networks, and with slight modifications, one can get a lower bound of $\Omega(\log \Delta \log n)$ on the progress bound in the classical model.

Lemma 6.1. *For any n and any $\Delta \leq n$, there exists a one-shot setting with a bipartite network of size n and maximum contention of at most Δ such that for any transmission schedule, it takes at least $\Omega(\log \Delta \log n)$ rounds till each receiver receives at least one message.*

6.2 Acknowledgment Time Lower Bound

In this section, we present the main technical result of the paper which is a lower bound of $\Omega(\frac{\Delta \log n}{\log^2 \log n})$ on the acknowledgment time in the classical radio broadcast model.

Theorem 6.2. *In the classical radio broadcast model, for any large enough n and any $\Delta \in [20 \log n, n^{0.1}]$, there exists a one-shot setting with a bipartite network of size n and maximum receiver degree at most Δ such that it takes at least $\Omega(\frac{\Delta \log n}{\log^2 \log n})$ rounds until all receivers have received all messages of their sender neighbors.*

In other words, in this one-shot setting, any algorithm that solves the local broadcast problem has a acknowledgment bound of $\Omega(\frac{\Delta \log n}{\log^2 \log n})$. To prove this theorem, instead of showing that randomized algorithms have low success probability, we show a stronger variant by proving an impossibility result: we prove that there exists a one-shot setting with the above properties such that, even with a centralized algorithm, it is *not possible* to schedule transmissions of nodes in $o(\frac{\Delta \log n}{\log^2 \log n})$ rounds such that each receiver receives the message of each of its neighboring senders successfully. In particular, this result shows that in this one-shot setting, for any randomized local broadcast algorithm, the probability that an execution shorter than $o(\frac{\Delta \log n}{\log^2 \log n})$ rounds successfully delivers message of each sender to all of its receiver neighbors is zero.

In order to make this formal, let us define a transmission schedule σ of length $L(\sigma)$ for a bipartite network to be a sequence $\sigma_1, \ldots, \sigma_{L(\sigma)} \subseteq S$ of senders. Having a sender $u \in \sigma_r$ indicates that at round r the sender u is transmitting its message. For a network G, we say that transmission schedule σ *covers* G if for every $v \in S$ and $u \in \mathcal{N}_G(v)$, there exists a round r such that $\sigma_r \cap \mathcal{N}_G(v) = \{u\}$, that is using transmission schedule σ, every receiver node receives all the messages of all of its sender neighbors. Also, we say that a transmission schedule σ is *short* if $L(\sigma) = o(\frac{\Delta \log n}{\log^2 \log n})$. With these notations, we are ready to state the main result of this section.

Lemma 6.3. *For any large enough n and $\Delta \in [20 \log n, n^{0.1}]$, there exists a bipartite network G with size n and maximum receiver degree at most Δ such that no short transmission schedule covers G.*

Proof (Proof Sketch for Lemma 6.3). Fix an arbitrary n and a $\Delta \in [20 \log n, n^{0.1}]$. Also let $\eta = n^{0.1}$, $m = \eta^9$. We use the probabilistic method [7] to show the existence of the network G with the aforementioned properties.

First, we present a probability distribution over a particular family of bipartite networks with maximum receiver degree Δ. To present this probability distribution, we show how to draw a random sample from it. Before getting to the details of this sampling, let us present the structure of this family. All networks of this family have a fixed set of nodes V. Moreover, V is partitioned into two nonempty disjoint sets S and R, which are respectively the set of senders and the set of receivers. We have $|S| = \eta$ and $|R| = m$. The total number of nodes in these two sets is $\eta + m = n^{0.1} + n^{0.9}$. We adjust the number of nodes to exactly n by adding enough isolated senders to the graph. To draw a random sample from this family, each receiver node $u \in R$ chooses Δ random senders from S uniformly (with replacement) as its neighbors. Also, choices of different receivers are independent of each other.

Having this probability distribution, we study the behavior of short transmission schedules over random graphs drawn from this distribution. For each fixed transmission schedule σ, let $P(\sigma)$ be the probability that σ covers a random graph G. Using a union bound, we can infer that for a random graph G, the probability that there exists a short transmission schedule that covers G is at most sum of the $P(\sigma)$-s, when σ ranges over all the short transmission schedules. Let us call this probability *the total coverage probability*. In order to prove the lower bound, we show that "the total coverage probability is in $e^{-\Omega(\eta)}$" and therefore, less than 1. Proving this claim completes the proof as with this claim, using the probabilistic method [7], we can conclude that there exists a bipartite network with maximum receiver degree of at most Δ such that no short transmission schedule covers it. To prove that the total coverage probability is $e^{-\Omega(\eta)}$, since the total number of short transmission schedules is less than 2^{η^3}, it is enough to show that for each short transmission schedule σ, $P(\sigma) = e^{-\Omega(\eta^5)}$.

Proving that for any fixed short schedule σ, $P(\sigma) = e^{-\Omega(\eta^5)}$ is the core part of the proof and also the hardest one. For this part, we use techniques similar to those that we are using in [26] for getting a lower bound for multicast in known radio networks. Let us first present some definitions. Fix a short transmission schedule σ. For each round r of σ, we say that this round is *lightweight* if $|\sigma(r)| < \frac{\eta}{2\Delta \log \eta}$. Since σ is a short transmission schedule, i.e., $L(\sigma) < \Delta \log \eta$, the total number of senders that transmit in at least one lightweight round of σ is less than $\frac{\eta}{2}$. Therefore, there are at least $\frac{\eta}{2}$ senders that never transmit in lightweight rounds of σ. We call these senders the *principal* senders of σ.

Throughout the rest of the proof, we focus on the principal senders of σ. For this, we divide the short transmission schedules into two disjoint categories, *adequate* and *inadequate*. We say that σ is an *adequate* transmission schedule if throughout σ, each principal node transmits in at least $\frac{\log \eta}{\log \log \eta}$ rounds. Otherwise we say that σ is an *inadequate* transmission schedule. We study inadequate and adequate transmission schedules in two separate lemmas (Lemmas 6.4 and 6.5), and prove that in each case $P(\sigma) = e^{-\Omega(\eta^5)}$.

Lemma 6.4. *For each inadequate short transmission schedule σ, the probability that σ covers a random graph is $e^{-\Omega(\eta^5)}$, i.e., $P(\sigma) = e^{-\Omega(\eta^5)}$.*

Proof (Proof Sketch). Let σ be an arbitrary inadequate short transmission schedule. Since σ is inadequate, there exists a principal sender node v that transmits in less than

$\frac{\log \eta}{\log \log \eta}$ rounds of σ. Also, since v is a principal sender, it does not transmit in any lightweight round. That is, in each round that v transmits, the number of sender nodes that transmit is at least $\frac{\eta}{2\Delta \log \eta}$. We show that in a random graph, v is unlikely to deliver its message to all its neighbors, i.e., that in a random graph, with probability $1-e^{-\Omega(\eta^5)}$, there exists a receiver neighbor of v that does not receive the message of v.

A formal proof for this claim requires rather careful probability arguments but the main intuition is as follows. In each round that v transmits, there is a high contention, i.e., at least $\frac{\eta}{2\Delta \log \eta}$ senders transmit. Thus, in a random graph, in most of those rounds, neighbors of v receive collisions. On the other hand, the number of rounds that v transmits in them is at most $\frac{\log \eta}{\log \log \eta}$. These two observations suggest that it is unlikely for all neighbors of v to receive its message.

Lemma 6.5. *For each adequate short transmission schedule σ, the probability that σ covers a random graph is $e^{-\Omega(\eta^5)}$, i.e., $P(\sigma) = e^{-\Omega(\eta^5)}$.*

Proof (Proof Sketch). Let σ be an arbitrary adequate short transmission schedule. Recall that principal senders of σ are defined as senders that do not transmit in lightweight rounds of σ. Let us say that a message is a principal message if its sender is a principal sender. Note that in a random graph, in expectation, each receiver is adjacent to at least $\frac{\Delta}{2}$ principal senders. Therefore, if σ covers a random graph, each receiver should receive, in expectation, at least $\frac{\Delta}{2}$ principal messages. Hence, since there are m different receivers, if σ covers a random graph, there are, in expectation, at least $\frac{m\Delta}{2}$ successful deliveries. Then, using a Chernoff bound, we can infer that if σ covers a random graph, with probability $1 - e^{-\Omega(\eta^9)}$, there are at least $\frac{m\Delta}{4}$ successful deliveries. To prove the lemma, we show that the probability that for a random graph, σ has $\frac{m\Delta}{4}$ successful deliveries is $e^{-\Omega(\eta^5)}$. Then, a union bound completes the proof of lemma.

Hence, the remaining part of the proof is to show that on a random graph, with probability $e^{-\Omega(\eta^5)}$, σ has less than $\frac{m\Delta}{4}$ successful deliveries. This part is the core part of the proof of this lemma. The formal reasoning for this part requires a careful potential argument but the intuition is based on the following simple observations. Suppose that σ has at least $\frac{m\Delta}{4}$ successful deliveries with probability $e^{-\Omega(\eta^5)}$. Since σ is an adequate transmission schedule, each principal sender transmits in at least $\frac{\log \eta}{\log \log \eta}$ rounds and because there are at least $\frac{\eta}{2}$ principal senders, there has to be at least $\frac{\eta \log \eta}{2 \log \log \eta}$ transmissions by principal senders. Now in each round σ, the number of transmitting senders should be at most $\Theta(\frac{\eta}{\Delta})$, or otherwise, the number of successful deliveries drops down exponentially as a function of the multiplicative distance from $\frac{\eta}{\Delta}$, and hence the total sum of them over all the rounds would not accumulate to $\frac{m\Delta}{4}$. If we assume that in each round roughly at most $\Theta(\frac{\eta}{\Delta})$ senders transmit, we directly get a lower bound of $\frac{\frac{\eta \log \eta}{2 \log \log \eta}}{\frac{\eta}{\Delta}} = \Theta(\frac{\Delta \log \eta}{\log \log \eta})$ on the number of rounds of σ which is in contradiction with the fact that σ is short. The formal proof of this part replaces this simplistic assumption by a more careful argument that, essentially, takes all the possibilities of the number of transmitters in each of the rounds into consideration, using a potential argument. This formal argument is omitted due to the space considerations.

7 Lower Bounds in the Dual Graph Model

In this section, we show a lower bound of $\Omega(\frac{\Delta' \log n}{\log^2 \log n})$ on the progress time of centralized algorithms in the dual graph model with collision detection. This lower bound directly yields a lower bound with the same value on the acknowledgment time in the same model. Together, these two bounds show that the optimized decay protocol presented in section 5 achieves almost optimal acknowledgment and progress bounds in the dual graph model. On the other hand, this result demonstrates a big gap between the progress bound in the two models, proving that progress is unavoidably harder (slower) in the dual graph model.

Theorem 7.1. *In the dual graph model, for each n and each $\Delta' \in [20 \log n, n^{\frac{1}{11}}]$, there exists a bipartite network $H^*(n, \Delta')$ with n nodes and maximum receiver G'-degree at most Δ' such that no algorithm can have progress bound of $o(\frac{\Delta' \log n}{\log^2 \log n})$ rounds. In the same network, no algorithm can have acknowledgment bound of $o(\frac{\Delta' \log n}{\log^2 \log n})$ rounds.*

Proof (Proof Outline). In order to prove this lower bound, in Lemma 7.2, we show a reduction from acknowledgment in the bipartite networks of the classical model to the progress in the bipartite networks of the dual graph model. In particular, this means that if there exists an algorithm with progress bound of $o(\frac{\Delta' \log n}{\log^2 \log n})$ in the dual graph model, then for any bipartite network H in the classical broadcast model, we have a transmission schedule $\sigma(H)$ with length $o(\frac{\Delta \log n}{\log^2 \log n})$ that covers H. Then, we use Theorem 6.2 to complete the lower bound.

Lemma 7.2. *Consider arbitrary n_2 and Δ_2 and let $n_1 = n_2\Delta_2$ and $\Delta'_1 = \Delta_2$. Suppose that in the dual graph model, for each bipartite network with n_1 nodes and maximum receiver G'-degree Δ'_1, there exists a local broadcast algorithm A with progress bound of at most $f(n_1, \Delta'_1)$. Then, for each bipartite network H with n_2 nodes and maximum receiver degree Δ_2 in the classical radio broadcast model, there exists a transmission schedule $\sigma(H)$ with length at most $f(n_2\Delta_2, \Delta_2)$ that covers H.*

Proof (Proof Sketch). Let H be a network in the classical radio broadcast model with n_2 nodes and maximum receiver degree at most Δ_2. We use algorithm A to construct a transmission schedule σ_H of length at most $f(n_2\Delta_2, \Delta_2)$ that covers H. We first construct a new bipartite network, *Dual(H)* = (G, G'), in the dual graph model with at most n_1 nodes and maximum receiver G'-degree Δ'_1. The set of sender nodes in the Dual(H) is equal to that in H. For each receiver u of H, let $d_H(u)$ be the degree of node u in graph H. Let us call the senders that are adjacent to u 'the *associates* of u'. In the network Dual(H), we replace receiver u with $d_H(u)$ receivers and we call these new receivers 'the *proxies* of u'. In graph G of Dual(H), we match proxies of u with associates of u, i.e., we connect each proxy to exactly one associate and vice versa. In graph G' of Dual(H), we connect all proxies of u to all associates of u. It is easy to check that Dual(H) has the desired size and maximum receiver degree.

Now we present a special adversary for the dual graph model. Later we construct transmission schedule σ_H based on the behavior of algorithm A in network Dual(H) against this adversary. This special adversary activates the unreliable links using the following procedure. Consider round r and receiver node w. (1) If exactly one G'-neighbor of w is transmitting, then the adversary activates only the links from w to its G-neighbors, (2) otherwise, adversary activates all the links from w to its G'-neighbors.

We focus on the executions of algorithm A on the network Dual(H) against the above adversary. By assumption, there exists an execution α of A with length at most $f(n_2\Delta_2, \Delta_2)$ rounds such that in α, every receiver receives at least one message. Let transmission schedule σ_H be the transmission schedule of execution α. Note that because of the above choice of adversary, in the execution α, each receiver can receive messages only from its G-neighbors. Suppose that w is a proxy of receiver u of H. Then because of the construction of Dual(H), each receiver node has exactly one G-neighbor and that neighbor is one of associates of u (the one that is matched to w). Therefore, in execution α, for each receiver u of H, in union, the proxies of u receive all the messages of associates of u. On the other hand, because of the choice of adversary, if in round r of σ a receiver w receives a message, then using transmission schedule σ_H in the classical radio broadcast model, u receives the message of the same sender in round r of σ_H. Therefore, using transmission schedule σ_H in the classical broadcast model and in network H, every receiver receives messages of all of its associates. Hence, σ_H covers H and we are done with the proof of lemma.

8 Centralized vs. Distributed Algorithms in the Dual Graph Model

In this section, we show that there is a gap in power between distributed and centralized algorithms in the dual graph model, but not in the classical model—therefore highlighting another difference between these two settings. Specifically, we produce dual graph network graphs where centralized algorithms achieve $O(1)$ progress while distributed algorithms have unavoidable slow progress. In more detail, our first result shows that distributed algorithms will have *at least one process* experience $\Omega(\frac{\Delta' \log n}{\log^2 \log n})$ progress, while the second result shows the *average* progress is $\Omega(\Delta')$. Notice, such gaps do not exist in the classical model, where our distributed algorithms from Section 5 can guarantee fast progress in all networks.

Theorem 8.1. *For any k and $\Delta' \in [20 \log k, k^{1/10}]$, there exists a dual graph network of size n, $k < n \leq k^4$, with maximum receiver degree Δ', such that the optimal centralized local broadcast algorithm achieves a progress bound of $O(1)$ in this network while every distributed local broadcast algorithm has a progress bound of $\Omega(\frac{\Delta' \log n}{\log^2 \log n})$.*

Our proof argument leverages the bipartite network proven to exist in Lemma 7.2 to show that all algorithms have slow progress in the dual graph model. Here, we construct a network consisting of many copies of this counter-example graph. In each copy, we leave one of the reliable edges as reliable, but *downgrade* the others to unreliable edges that act reliable. A centralized algorithm can achieve fast progress in each of these copies as it only needs the processes connected to the single reliable edge to broadcast. A distributed algorithm, however, does not know which edge is actually reliable, so it still has slow progress. We prove that in one of these copies, the last message to be delivered comes across the only reliable edge, w.h.p. This is the copy that provides the slow progress needed by the theorem.

Notice, in some settings, practioners might tolerate a slow worst-case progress (e.g., as established in Theorem 8.1), so long as *most* processes have fast progress. In our next theorem, we show that this ambition is also impossible to achieve. To do so, we first need a definition that captures the intuitive notion of many processes having slow progress. In more detail, given an execution of the one-shot local broadcast problem (see Section 2), with processes in *sender set* S being passed messages, label each receiver that neighbors S in G with the round when it first received a message. The *average progress* of this execution is the average of these values. We say an algorithm has an *average progress of* $f(n)$, with respect to a network of size n and sender set S, if executing that algorithm in that network with those senders generates an average progress value of no more than $f(n)$, w.h.p. We now bound this metric in the same style as above

Theorem 8.2. *For any n, there exists a dual graph network of size n and a sender set, such that the optimal centralized local broadcast algorithm has an average progress of $O(1)$ while every distributed local broadcast algorithm has an average progress of $\Omega(\Delta')$.*

Our proof uses a reduction argument. We show how a distributed algorithm that achieves fast average progress in a specific type of dual graph network can be transformed to a distributed algorithm that solves global broadcast fast in a different type of dual graph network. We then apply a lower bound from [20] that proves no fast solution exists for the latter—providing our needed bound on progress.

References

1. Bachir, A., Dohler, M., Wattayne, T., and Leung, K.: " MAC Essentials for Wireless Sensor Networks". IEEE Communications Surveys and Tutorials 12, 2 (2010), 222-248.
2. Shan, H., Zhuang, W., and Wand, Z.: "Distributed Cooperative MAC for Multihop Wireless Networks". IEEE Communications Magazine 47, 2 (February 2009), 126-133.
3. Sato, N., and Fujii, T.: "A MAC Protocol for Multi-Packet Ad-Hoc Wireless Network Utilizing Multi-Antenna". In Proceedings of the IEEE Conference on Consumer Communications and Networking (2009).
4. Sayed, S., and Yand, Y.: "BTAC: A Busy Tone Based Cooperative MAC Protocol for Wireless Local Area Networks". In Proceedings of the Interneational Conference on Communications and Networking in China (2008).
5. Sun, Y., Gurewitz, O., and Johnson, D. B.: "RI-MAC: a Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks", Proceedings of the ACM Conference on Embedded Network Sensor Systems, 2008.
6. Rhee, I., Warrier, A., Aia, M., Min, J., and Sichitiu, M. L.: "Z-MAC: a Hybrid MAC for Wireless Sensor Networks". IEEE/ACM Trans. on Net. 16 (June 2008), 511-524.
7. Alon, N. and Spencer, J. H.: "The probabilistic method". John Wiley & Sons, New York, 1992.
8. Chlamtac, I., Kutten, S. : "On Broadcasting in Radio Networks–Problem Analysis and Protocol Design". IEEE Trans. on Communications (1985).
9. Bar-Yehuda, R., Goldreich, O., and Itai, A.: "On the time-complexity of broadcast in radio networks: an exponential gap between determinism randomization". In PODC 87: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing, pages 98-108, New York, NY, USA, 1987. ACM.
10. Alon, N., Bar-Noy, A., Linial, N., and Peleg., D.: "A lower bound for radio broadcast". J. Comput. Syst. Sci., 43(2):290-298, 1991.

11. Chrobak, M., Gasieniec, L., and Rytter, W.: "Fast broadcasting and gossiping in radio networks", J. Algorithms 43, 2 (May 2002), 177-189.
12. Chlebus, B. S., Gasieniec, L., Gibbons, A., Pelc, A., and Rytter, W.: "Deterministic broadcasting in unknown radio networks". In Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms (SODA '00). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 861-870.
13. Chlebus, B. S., Gasieniec, L., Ostlin, A., and Robson, J. M.: "Deterministic Radio Broadcasting" ICALP 2000.
14. Clementi, A., Monti, A. , and Silvestri, R.: "Selective families, superimposed codes, and broadcasting on unknown radio networks". In the annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 709-718, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.
15. Clementi, A., Crescenzi, P., Monti, A., Penna, P., and Silvestri, R.: "On Computing Adhoc Selective Families". In Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 5th International Workshop on Randomization and Approximation Techniques in Computer Science: Approximation, Randomization and Combinatorial Optimization, pages 211–222, 2001.
16. Clementi, A., Monti, A. , and Silvestri, R.: "Round robin is optimal for fault-tolerant broadcasting on wireless networks". J. Parallel Distrib. Comput., 64(1):89-96, 2004.
17. Gasieniec, L., Peleg, D., and Xin, Q.: "Faster communication in known topology radio networks". In Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing (PODC '05). ACM, New York, NY, USA, 129-137.
18. Kuhn, F., Lynch, N., and Newport, C. : "The Abstract MAC Layer". Technical Report MIT-CSAIL-TR-2009-009, MIT CSAIL, Cambridge, MA, February 20, 2009.
19. Kuhn, F., Lynch, N., and Newport, C.: "The Abstract MAC Layer". Distributed Computing, 24(3):187-296, 2011. Special issue from DISC 2009 23rd International Symposium on Distributed Computing.
20. Kuhn, F., Lynch, N., and Newport, C.: "Brief Announcement: Hardness of Broadcasting in Wireless Networks with Unreliable Communication". Proceedings of the ACM Symposium on the Principles of Distributed Computing (PODC), Calgary, Alberta, Canada, August 2009.
21. Cornejo, A., Lynch, N., Viqar, S., and Welch, J.: "A Neighbor Discovery Service Using an Abstract MAC Layer". Forty-Seventh Annual Allerton Conference, Champaign-Urbana, IL, October, 2009. Invited paper.
22. Kuhn, F., Lynch, N., and Newport, C., Oshman, R., and Richa, A.: "Broadcasting in unreliable radio networks". In Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing (PODC '10). ACM, New York, NY, USA, 336-345.
23. Khabbazian, M., Kuhn, F., Kowalski, D. R., and Lynch. N.: "Decomposing broadcast algorithms using abstract MAC layers". In Proceedings of the 6th International Workshop on Foundations of Mobile Computing (DIALM-POMC '10). ACM, New York, NY, USA, 13-22.
24. Khabbazian, M., Kuhn, F., Lynch, N., Medard, M., and ParandehGheibi, A.: "MAC Design for Analog Network Coding". FOMC 2011: The Seventh ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing, San Jose, CA, June 2011.
25. Censor-Hillel, K., Gilbert, S., Kuhn, F., Lynch, N., and Newport, C.: "Structuring Unreliable Radio Networks". Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, San Jose, California, June 6-8, 2011.
26. Ghaffari, M., and Haeupler, B., and Khabbazian, M.: "The complexity of Multi-Message Broadcast in Radio Networks with Known Topology". Manuscript in preparation, 2012
27. Ghaffari, M., and Haeupler, B., Lynch, N., and Newport, C.: "Bounds on Contention Management in Radio Networks". http://arxiv.org/abs/1206.0154.