Broadcasting Competitively Against Adaptive Adversary in Multi-Channel Radio Networks

Haimin Chen
State Key Laboratory for Novel Software Technology, Nanjing University, China
haimin.chen@smail.nju.edu.cn

Chaodong Zheng
State Key Laboratory for Novel Software Technology, Nanjing University, China
chaodong@nju.edu.cn

Abstract

Broadcasting in wireless networks is vulnerable to adversarial jamming. To thwart such behavior, *resource competitive analysis* is proposed. In this framework, sending, listening, or jamming on one channel for one time slot costs one unit of energy. The adversary can employ arbitrary strategy to disrupt communication, but has a limited energy budget T. The honest nodes, on the other hand, aim to accomplish broadcast while spending only $o(T)$. Previous work has shown, in a C-channels network containing n nodes, for large T values, each node can receive the message in $\tilde{O}(T/C)$ time, while spending only $\tilde{O}(\sqrt{T/n})$ energy. However, these multi-channel algorithms only work for certain values of n and C, and can only tolerate an oblivious adversary.

In this work, we provide new upper and lower bounds for broadcasting in multi-channel radio networks, from the perspective of resource competitiveness. Our algorithms work for arbitrary n, C values, require minimal prior knowledge, and can tolerate a powerful adaptive adversary. More specifically, in our algorithms, for large T values, each node's runtime is $O(T/C)$, and each node's energy cost is $\tilde{O}(\sqrt{T/n})$. We also complement algorithmic results with lower bounds, proving both the time complexity and the energy complexity of our algorithms are optimal or near-optimal (within a poly-log factor). Our technical contributions lie in using "epidemic broadcast" to achieve time efficiency and resource competitiveness, and employing coupling techniques in the analysis to handle the adaptivity of the adversary. At the lower bound side, we first derive a new energy complexity lower bound for 1-to-1 communication in the multi-channel setting, and then apply simulation and reduction arguments to obtain the desired result.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Broadcast, radio networks, resource competitive algorithms

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2020.22

Related Version Full paper is available on arXiv [9]: `https://arxiv.org/abs/2001.03936`.

Funding This work is supported by the Ministry of Science and Technology of China under Grant No.: 2018YFB1003200, and by the National Natural Science Foundation of China (NSFC) under Grant No.: 61702255 and 61672275.

Acknowledgements We would like to thank Weiming Feng and Guangxu Yang for the discussions.

1 Introduction

Consider a synchronous, time-slotted, single-hop wireless network formed by n devices (or, *nodes*). Each node is equipped with a radio transceiver, and these nodes communicate over a shared wireless medium containing C channels. In each time slot, each node can operate on one arbitrary channel, but cannot send and listen simultaneously. In this model, we study a fundamental communication problem – broadcasting – in which a designated *source* node wants to disseminate a message m to all other nodes in the network.

24th International Conference on Principles of Distributed Systems (OPODIS 2020).
Editors: Quentin Bramas, Rotem Oshman, and Paolo Romano; Article No. 22; pp. 22:1–22:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Lots of modern wireless devices are powered by battery and are able to switch between active and sleep states. Often, sending and listening occurring during active state dominate the energy expenditure, while sleeping costs much less [21]. Therefore, when running an algorithm, each node's energy complexity (or, *energy cost*) is often defined as the number of channel accesses [5–7, 17]; while time complexity is the number of slots till it halts.

The open and shared nature of wireless medium makes it vulnerable to jamming [16]. To thwart such behavior, one reasonable restriction is to bound the total amount of jamming, as injecting interfering signals also incurs operational cost. Specifically, we assume the existence of a jamming adversary called Eve. She can jam multiple channels in each slot, and jamming one channel for one slot costs one unit of energy. Eve has an energy budget T that is *unknown* to the nodes, and she can employ *arbitrary* strategy to disrupt communication.

This setting motivates the notation of *resource competitive algorithms* [1, 4, 8, 15, 18, 19] which focus on optimizing relative cost. Specifically, assume for each node the cost of sending or listening on one channel for one slot is one unit of energy (while idling is free),[1] can we design broadcast algorithms that ensure each node's cost is only $o(T)$? Such results would imply Eve cannot efficiently stop nodes from accomplishing the distributed computing task in concern. Interestingly enough, the answer is positive. In particular, Gilbert et al. [15] present a resource competitive broadcast algorithm in the single-channel radio network setting: with high probability, each node receives the message and terminates within $\tilde{O}(T+n)$ slots, while spending only $\tilde{O}(\sqrt{T/n}+1)$ energy.[2] This algorithm works even when Eve is adaptive and n is unknown to the nodes. Later, Chen and Zheng [8] consider the multi-channel setting: they show that when Eve is oblivious and $C = O(n)$, having multiple channels allows a linear speedup in time complexity, while the energy cost of each node remains to be $\tilde{O}(\sqrt{T/n}+1)$.

In this paper, we develop two new multi-channel broadcast algorithms that can tolerate a stronger adaptive adversary and work for arbitrary n, C values, without sacrificing time efficiency or resource competitiveness. The first algorithm – called MultiCastAdp– needs to know n; while the other more complicated one – called MultiCastAdvAdp– does not. Both algorithms are randomized, and in the interesting case where T is large compared with n and C, each node's runtime is $O(T/C)$, while each node's energy cost is $\tilde{O}(\sqrt{T/n})$.[3]

▶ **Theorem 1.** *MultiCastAdp guarantees the following properties w.h.p.: (a) all nodes receive the message and terminate within* $O(T/C+\tau_{time}) = \tilde{O}(T/C+\max\{n/C, C/n\})$ *slots; and (b) the cost of each node is* $O(\sqrt{T/n}\cdot\sqrt{\lg T}\cdot\lg n+\tau_{cost}) = \tilde{O}(\sqrt{T/n}+C/n)$*.*

- *When* $C = O(n)$, $\tau_{time} = (n/C)\cdot\lg(n/C)\cdot\lg^2 n$, *and* $\tau_{cost} = \lg(n/C)\cdot\lg n$.
- *When* $C = \Omega(n)$, $\tau_{time} = (C/n)\cdot\lg(C/n)\cdot\lg^2 n$, *and* $\tau_{cost} = (C/n)\cdot\lg(C/n)\cdot\lg n$.

▶ **Theorem 2.** *MultiCastAdvAdp guarantees the following properties w.h.p.: (a) all nodes receive the message and terminate within* $O(T/C+(nC+C^2)\cdot\lg^4(nC)) = \tilde{O}(T/C+nC+C^2)$ *slots; and (b) the cost of each node is* $O(\sqrt{T/n}\cdot\lg^2 T+C^2\cdot\lg^5(nCT)+(nC+C^2)\cdot\lg^4(nC)) = \tilde{O}(\sqrt{T/n}+nC+C^2)$*.*

We also complement algorithmic results with lower bounds. Specifically, the $O(T/C)$ term in runtime is optimal, as Eve can jam all C channels continuously for T/C slots. Meanwhile, the $\tilde{O}(\sqrt{T/n})$ term in energy cost matches lower bound up to a poly-logarithmic factor. Thus our algorithms achieve (near) optimal time and energy complexity simultaneously.

[1] In reality, the cost for sending, listening, and jamming might differ, but they are often in the same order. The assumptions here are mostly for the ease of presentation, and are consistent with existing work. Moreover, allowing different actions to have different constant costs will not affect the results.

[2] We say an event happens *with high probability (w.h.p.)* if the event occurs with probability at least $1-1/n^c$, for some tunable constant $c \geq 1$. Moreover, we use $\tilde{O}$ to hide poly-log factors in n, C, and T.

[3] The primary goal of resource competitive algorithms is to optimize nodes' cost for large T values, see previous work (e.g., [4, 18]) and discussion on resource competitiveness in Section 1.2 for more details.

▶ **Theorem 3.** *For an adaptive adversary with budget T, any fair multi-channel broadcast algorithm that succeeds with constant probability imposes an expected cost of $\Omega(\sqrt{T/n})$ per node. Notice, an algorithm is* fair *if all participating nodes have the same expected cost; both* MultiCastAdp *and* MultiCastAdvAdp *are fair.*

1.1 Related Work

Broadcasting in radio networks is non-trivial due to collisions. Classical results often rely on variants of the Decay procedure [3], while recent ones (e.g., [10, 14]) tend to employ more advanced techniques (e.g., network decomposition) to improve performance. Besides time complexity, energy cost has also been taken into consideration when building communication primitives (e.g., [5–7, 13]), but usually without assuming the existence of a jamming adversary.

Distributed computing in jamming-prone environment has attracted a lot of attention as well. Researchers from the theory community usually pose certain restrictions on the behavior of the malicious user(s), and then develop corresponding countermeasures (e.g., [2, 11, 20, 22]). Unfortunately, these restrictions somewhat limit the adversary's strategy, and many of the proposed algorithms also require honest nodes to spend a lot of energy. In view of these, *resource competitive analysis* [4] is proposed. This framework allows more flexibility for the adversary, hence potentially better captures reality. However, it also brings new challenges to the design and analysis of algorithms.

In 2011, King, Saia, and Young [19] developed the first resource competitive algorithm, in the context of 1-to-1 communication. (That is, Alice wants to send a message to Bob.) Specifically, the proposed Las Vegas algorithm ensures the expected cost of Alice and Bob is only $O(T^{0.62} + 1)$. As mentioned earlier, Gilbert et al. [15] later devise a single-channel broadcast algorithm that is resource competitive against jamming. They have also proved several lower bounds showing the algorithm's energy cost is near optimal. The work that is most closely related to ours is by Chen and Zheng [8], in which several multi-channel broadcast algorithms are developed. However, an important drawback of [8] is that it only considers an oblivious adversary, while all other previous results can tolerate an adaptive (or even reactive) adversary. In this paper, we close the gap by considering an adaptive adversary, and provide similar or better results than [8] that work for arbitrary values of n and C. We also prove our results are (near) optimal by deriving new lower bounds.

1.2 Additional Model Details

All nodes in the network start execution simultaneously and can independently generate random bits. In each slot, each node either sends a message on a channel, or listens on a channel, or remains idle. Only listening nodes get feedback regarding channel status. The adversary Eve is adaptive: at the beginning of each slot, she is given all past execution history and can use these information to determine her behavior. However, she does *not* know honest nodes' random bits or behavior of the current slot.

In each slot, for each listening node, the channel feedback is determined by the number of sending nodes on that channel and the behavior of Eve. Specifically, consider a slot and a channel ch. If no node sends on ch and Eve does not jam ch, then nodes listening on ch hear silence. If exactly one node sends a message on ch and Eve does not jam ch, then nodes listening on ch receive the unique message. Finally, if at least two nodes send on ch or Eve jams ch, then nodes listening on ch hear noise. Note that we assume nodes cannot tell whether noise is due to jamming or message collision (or both).

We adopt the following definition of resource competitive algorithms introduced in [4]:

▶ **Definition 4.** *Consider an execution π in which nodes execute algorithm $\mathcal{A}_N$ and Eve employs strategy $\mathcal{A}_E$. Let $\texttt{cost}_u(\pi)$ denote the energy cost of node u, and $T(\pi)$ denote the energy cost of Eve. We say $\mathcal{A}_N$ is (ρ, τ)-*resource competitive *if* $\max_u\{\texttt{cost}_u(\pi)\} \leq \rho(T(\pi)) + \tau$ *for any execution π.*

In above, ρ is a function of T and possibly other parameters (such as n, C). It captures the additional cost nodes incur due to jamming. The other function τ captures the cost of the algorithm when Eve is absent, thus τ should not depend on T. Most resource competitive algorithms aim to minimize ρ, while keeping τ reasonably small.

1.3 Overview of Techniques

Fast and competitive broadcast against jamming. Most resource competitive broadcast algorithms group slots into consecutive *epochs*, and execute a jamming-resistant broadcast scheme within each epoch. In the single-channel setting, often the core idea is to broadcast "sparsely" [18, 19]. Consider 1-to-1 communication as an example. If both nodes send and listen in $\Theta(\sqrt{R})$ random slots in an epoch of length R, then by a birthday-paradox argument, successful transmission will occur with constant probability even if Eve jams constant fraction of all R slots. In the multi-channel setting, "*epidemic broadcast*" is employed [8]. In the simplest form of this scheme, in each time slot, each node will choose a random channel from $[C] = \{1, 2, \cdots, C\}$. Then, each informed node (i.e., the node knows the message m) will broadcast m with a constant probability, while each uninformed node will listen with a constant probability. If $C = n/2$, broadcast will complete in $O(\lg n)$ slots w.h.p., and this claim holds even if Eve jams constant fraction of all channels for constant fraction of all slots.

In designing MultiCastAdp and MultiCastAdvAdp, one key challenge is to extend the basic epidemic broadcast scheme to guarantee an optimal $O(T/C)$ runtime for arbitrary n, C values, without increasing energy expenditure. To that end, we note that in the single-channel setting, [15] has shown $\Theta(1/\sqrt{Rn})$ is roughly an optimal working probability (i.e., sending/listening probabilities). When C channels are available, a good way to adjust the probability would be to multiply it by a factor of $\sqrt{C}$ (i.e., $\Theta(\sqrt{C/(Rn)})$). Intuitively, the reason being: if each node works on $\sqrt{C}$ random channels simultaneously in each slot, then again by a birthday-paradox argument, each pair of nodes will meet on at least one channel with at least constant probability, which effectively means the optimal single-channel analysis could be applied again. Of course nodes do not have multiple transceivers and cannot work on multiple channels simultaneously, but over a period of time, multiplying the single-channel working probability by $\sqrt{C}$ achieves similar effect. On the other hand, although the working probability of nodes is increased by a factor of $\sqrt{C}$, the energy expenditure of Eve will increase by a factor of $\Theta(C)$. As a result, compared with single-channel solutions, our algorithms have a $\Theta(C)$ speedup in time, yet the resource competitive ratio is unchanged.

Termination and the coupling technique. Termination mechanism is another key integrant, it ensures nodes stop execution correctly and timely. For each node u, a helpful termination criterion is comparing N_u – the number of silent slots it observed during the current epoch – to some pre-defined threshold. To argue the correctness of our algorithms, we often need to show N_u is close to its expected value. However, this is non-trivial if Eve is adaptive.

To see this, consider an epoch containing R slots. Define G_i as the *behavior* (i.e., channels choices and actions) of all nodes in slot i, and define Q_i – the set of channels that are *not* jammed by Eve – as the *jamming result* of slot i. Note that N_u can be written as the sum of R indicator random variables: $N_u = \sum_{i=1}^{R} N_{u,i}$, where $N_{u,i} = 1$ iff u hears silence in the i^{th} slot. $N_{u,i}$ is determined by G_i and Q_i, but in general Q_i can be arbitrary function of $\{G_1, G_2, \cdots, G_{i-1}, Q_1, Q_2, \cdots, Q_{i-1}\}$. Nonetheless, in case Eve is oblivious (i.e., an offline adversary), her optimal strategy would be a fixed vector of jamming results $\langle q_1, q_2, \cdots, q_R \rangle$, thus $\{N_{u,1}, N_{u,2}, \cdots, N_{u,R}\}$ are mutually independent when $\{G_1, G_2, \cdots, G_R\}$ are mutually independent (this can be easily enforced by the algorithm). Therefore, if Eve is oblivious, we can directly apply powerful concentration inequalities like Chernoff bounds to show N_u is close to its expectation. However, once Eve becomes adaptive, Q_i could depend on $\{G_1, \cdots, G_{i-1}\}$ and above observations no longer hold: $\{N_{u,1}, \cdots, N_{u,R}\}$ could be dependent!

In this paper, we leverage the *coupling* technique (see, e.g., [12]) extensively to resolve the dependency issue. Specifically, for each vector of jamming results over one epoch, we create a coupled execution and relate N_u to a corresponding random variable in the coupled execution. By carefully crafting the coupling, the random variable in the coupled execution can be interpreted as the sum of a set of independent random variables, allowing us to bound the probability that N_u deviates a lot from its expectation. However, there is a catch in this approach: bounding the probability that N_u deviates a lot from its expectation requires us to sum the failure probability over all jamming results vectors, but there may be $\Theta(2^{CR})$ such vectors! Our solution to this new problem is to group all vectors into fewer categories, so that vectors within one category have identical or similar effects on the metric we concern.

▶ Remark. Techniques like "principle of deferred decision", or the ones used in previous work, cannot resolve the dependency issue directly in our setting. See full version of our paper for more discussion.

Lower bound. Existing result [15] indicates fair broadcast in the single-channel settings requires each node spending $\Omega(\sqrt{T/n})$ energy, but could it be the case that having multiple channels also reduces the energy complexity of the problem? We show the answer is negative.

Specifically, for any multi-channel broadcast algorithm $\mathcal{A}_n$, we devise a corresponding multi-channel 1-to-1 communication algorithm $\mathcal{A}_2$ that simulates $\mathcal{A}_n$ internally. We also devise a jamming strategy $\mathcal{S}$ for disrupting $\mathcal{A}_n$ and $\mathcal{A}_2$: in each slot, for each channel, Eve jams that channel iff a successful transmission will occur on that channel with a probability exceeding $1/T$. $\mathcal{A}_2$ and $\mathcal{S}$ are carefully constructed so that algorithms' success probabilities and nodes' energy expenditure in the two executions (i.e., in $(\mathcal{A}_n, \mathcal{S})$ and $(\mathcal{A}_2, \mathcal{S})$) are closely connected. Then, we derive an energy complexity lower bound for multi-channel 1-to-1 communication assuming Eve uses $\mathcal{S}$. (This result, Theorem 17 in Section 7, could be of independent interest and is strong in two aspects: (a) the bound holds even if the two nodes has multiple transceivers; (b) its proof uses a novel approach to handle adaptive Monte Carlo algorithms.) Finally, an energy complexity lower bound for $\mathcal{A}_n$ is obtained via reduction.

2 Notations

Let V be the set of all nodes. Since all algorithms developed in this paper proceed in epochs, consider a slot i in an epoch of length R, where $1 \leq i \leq R$. Denote $Q_i \in 2^{[C]}$ as the jamming result of the i^{th} slot: Q_i is the set of channels that are *not* jammed by Eve in the i^{th} slot. Denote $G_i = \langle (G^{ch}_{i,v})_{v \in V}, (G^{act}_{i,v})_{v \in V} \rangle$ as the behavior (i.e., channel choices and actions) of the n nodes in the i^{th} slot: $G_i \in \Omega = [C]^n \times \{\text{send}, \text{listen}, \text{idle}\}^n$.[4] Since Eve is adaptive, Q_i may depend on $\boldsymbol{G}_{<i} = (G_1, \cdots, G_{i-1})$. Lastly, define $\boldsymbol{Q}_{\leq i} = (Q_1, \cdots, Q_i)$.

[4] There is a technical subtlety worth clarifying. The "behavior" here does not care about the exact content to be broadcast if some node(s) choose to send message(s) in a slot. That is, for each slot, the "behavior" here is *not* some element in $[C]^n \times (M \cup \{\text{listen}, \text{idle}\})^n$, where M is the set of all possible messages. This is for the ease of presentation and will not affect the correctness of our results.

To quantify the severity of jamming from Eve, for a given slot, we use $\mathcal{E}(> x)$ (respectively, $\mathcal{E}(\geq x)$, $\mathcal{E}(< x)$, $\mathcal{E}(\leq x)$) to denote that in a slot, more than (respectively, at least, less than, at most) x fraction of the C channels are *not* jammed by Eve. In the following, we use $\mathcal{E}(\cdot x)$ to represent one of the above four forms. (I.e., "$\cdot$" denotes "$>$", "$\geq$", "$<$", or "$\leq$".)

For an epoch, we use $\mathcal{E}^{(>y)}(\cdot x)$ (respectively, $\mathcal{E}^{(\geq y)}(\cdot x)$, $\mathcal{E}^{(<y)}(\cdot x)$, $\mathcal{E}^{(\leq y)}(\cdot x)$) to denote the event that for more than (respectively, at least, less than, at most) y fraction of the R slots, $\mathcal{E}(\cdot x)$ happen. For example, $\mathcal{E}^{(>0.1)}(> 0.2)$ means in an epoch, for more than 0.1 fraction of all slots, Eve leaves more than 0.2 fraction of all channels unjammed.

Define negation operation in the following manner: $(\overline{> x}) = (\leq x)$ and vice versa; $(\overline{< x}) = (\geq x)$ and vice versa. Further define complement operation in the following manner: $\mathtt{C}(> x) = (< 1 - x)$ and vice versa; $\mathtt{C}(\geq x) = (\leq 1 - x)$ and vice versa. It is easy to verify $\mathcal{E}^{(\cdot y)}(\cdot x) = \mathcal{E}^{(\mathtt{C}(\cdot y))}(\overline{\cdot x})$ and $\overline{\mathcal{E}^{(\cdot y)}(\cdot x)} = \mathcal{E}^{(\overline{\cdot y})}(\cdot x)$. Therefore:

$$\overline{\mathcal{E}^{(\geq y)}(\geq x)} = \overline{\mathcal{E}^{(\leq 1-y)}(\overline{\geq x})} = \mathcal{E}^{(>1-y)}(\overline{\geq x}) = \mathcal{E}^{(>1-y)}(< x)$$

Again, as a simple example, the above equality implies "if in an epoch, it is not the case that in at least 0.1 fraction of all slots Eve leaves at least 0.2 fraction of all channels unjammed, then it must be the case that in more than 0.9 fraction of all slots, Eve leaves less than 0.2 fraction of all channels unjammed; and vice versa". (I.e., $\overline{\mathcal{E}^{(\geq 0.1)}(\geq 0.2)} = \mathcal{E}^{(>0.9)}(< 0.2)$.)

3 The MultiCastAdp Algorithm

Each node u maintains a Boolean variable M_u to indicate whether it knows the message m (in which case M_u is *true* and u is *informed*) or not (in which case M_u is $false$ and u is *uninformed*). Initially, only the source node sets $M_u = true$. The algorithm proceeds in epochs and the i^{th} epoch contains $R_i = a \cdot 4^i \cdot i \cdot \lg^2 n$ slots, where a is some large constant. In each slot in epoch i, for each node u that is still executing the algorithm (i.e., the node is still *active*), it will hop to a uniformly chosen random channel. Then, u will choose to broadcast or listen each with probability $p_i = (\sqrt{C/n})/2^i$. If u decides to broadcast and $M_u = true$, it sends m; otherwise, u sends a special beacon message $\pm$. On the other hand, if in a slot u decides to listen, it will record the channel feedback. Finally, by the end of an epoch i, for a node u, if among the slots it listened within this epoch, at least $(p_i R_i)/2$ are silent slots, then u will halt. One point worth noting is, the first epoch number is not necessarily one; instead, it is chosen as a sufficiently large integer to ensure $p_i \leq 1/2$ and $p_i \leq C/(4n)$. Hence, the first epoch number is $I_b = 2 + \lceil \max\{\lg(\sqrt{n/C}), \lg(\sqrt{C/n})\} \rceil$. Complete pseudocode of MultiCastAdp is provided in the full version of the paper.

4 Analysis of MultiCastAdp

Effectiveness of epidemic broadcast. The first technical lemma states if in an epoch jamming from Eve is not strong and every node is active, then all nodes will be informed by the end of the epoch. More specifically:

▶ **Lemma 5.** *If all nodes are active at the beginning of epoch i, and during epoch i event $\mathcal{E}^{\geq y_1}(\geq x_1)$ occurs, then by the end of this epoch, all nodes will be informed, with probability at least $1 - n^{-\Theta(i)}$. Here, $x_1 = y_1 = 0.1$, and $\mathcal{E}^{\geq y_1}(\geq x_1)$ is defined in Section 2.*

This lemma highlights the effectiveness of the epidemic broadcast scheme. Intuitively, it holds because when less than $n/2$ nodes know message m, the number of informed nodes will increase by some constant factor every so often; and once at least $n/2$ nodes know m, remaining uninformed nodes will quickly learn the message too. To prove this intuition rigorously, however, we need to apply the coupling technique.

To construct the coupling, we first specify how nodes' behavior is generated. Fix an epoch, imagine two sufficiently long bit strings $\boldsymbol{T}_{high}$ and $\boldsymbol{T}_{low}$ in which each bit is generated independently and uniformly at random. Divide $\boldsymbol{T}_{high}$ and $\boldsymbol{T}_{low}$ into consecutive *chunks* of equal size, such that each chunk provides enough random bits for n nodes to determine their behavior in a slot. More formally, $\boldsymbol{T}_{high} = (T_{hi}^{(1)}, T_{hi}^{(2)}, \cdots, T_{hi}^{(R)})$ and $\boldsymbol{T}_{low} = (T_{lo}^{(1)}, T_{lo}^{(2)}, \cdots, T_{lo}^{(R)})$, where each $T_{hi}^{(*)}$ or $T_{lo}^{(*)}$ is a chunk. Next, we introduce three processes that are used during the coupling: β, β', and γ.

We begin with β, which is an execution of MultiCastAdp with adversary Eve. The tricky part about β is: in the i^{th} slot, nodes' behavior G_i is *not* determined by $T_{hi}^{(i)}$ or $T_{lo}^{(i)}$ directly. Instead, it is generated in a more complicated way. Specifically, at the beginning of slot i, Eve first computes its jamming result Q_i (i.e., the set of unjammed channels) based on $\boldsymbol{Q}_{<i}$ and $\boldsymbol{G}_{<i}$. If $|Q_i| \geq x_1 C$ and the number of previously used chunks from $\boldsymbol{T}_{high}$ is no more than $y_1 R$, then we pick the next unused chunk from $\boldsymbol{T}_{high}$; otherwise, we pick the next unused chunk from $\boldsymbol{T}_{low}$. Assume $T^{(j)}$ is the chosen chunk, and it computes to nodes' behavior $\langle(\hat{G}_v^{ch})_{v\in V}, (\hat{G}_v^{act})_{v\in V}\rangle$. Still, we do not use $\langle(\hat{G}_v^{ch})_{v\in V}, (\hat{G}_v^{act})_{v\in V}\rangle$ as nodes' behavior. Instead, we permute the channel choices according to the jamming result. Specifically, for each $q \in 2^{[C]}$, define permutation π_q on $[C]$ as follows: for $1 \leq k \leq |q|$, $\pi_q(k)$ is the k^{th} smallest element in q; and for $|q|+1 \leq k \leq C$, $\pi_q(k)$ is the $(k-|q|)^{\text{th}}$ smallest element in $[C]\backslash q$. (For example, if $C = 5$ and $q = \{2, 4\}$, then π_q permutes $\langle 1, 2, 3, 4, 5\rangle$ to $\langle 2, 4, 1, 3, 5\rangle$.) Further define bijection $\Psi_q : \Omega \to \Omega$ using π_q:

$$\Psi_q\left(\left\langle \left(\hat{G}_v^{ch}\right)_{v\in V}, \left(\hat{G}_v^{act}\right)_{v\in V} \right\rangle\right) = \left\langle \left(\pi_q\left(\hat{G}_v^{ch}\right)\right)_{v\in V}, \left(\hat{G}_v^{act}\right)_{v\in V} \right\rangle$$

Now, we use $\langle(\pi_q(\hat{G}_v^{ch}))_{v\in V}, (\hat{G}_v^{act})_{v\in V}\rangle$ as nodes' behavior G_i in slot i. Formally, let $K(\boldsymbol{Q}_{\leq i}) = \sum_{j=1}^{i} \mathbb{I}[|Q_j| \geq x_1 C]$ count the number of weakly jammed slots (i.e., $|Q_j| \geq x_1 C$) among the first i slots, where each $\mathbb{I}[|Q_j| \geq x_1 C]$ is an indicator random variable. Then, G_i can be defined as:

$$G_i = \begin{cases} \Psi_{Q_i}\left(T_{hi}^{(K(\boldsymbol{Q}_{\leq i}))}\right), & |Q_i| \geq x_1 C \text{ and } K(\boldsymbol{Q}_{\leq i}) \leq y_1 R \\ \Psi_{Q_i}\left(T_{lo}^{(i-K(\boldsymbol{Q}_{\leq i}))}\right), & |Q_i| < x_1 C \text{ and } K(\boldsymbol{Q}_{\leq i}) \leq y_1 R \\ \Psi_{Q_i}\left(T_{lo}^{(i-y_1 R)}\right), & \text{otherwise} \end{cases}$$

Careful readers might suspect does $\boldsymbol{G} = (G_1, G_2, \cdots, G_R)$ in process β really has the correct distribution $\boldsymbol{\mathcal{G}}$ we want. (That is, $\boldsymbol{\mathcal{G}}$ is the distribution in which the behavior of the nodes are determined by, say $\boldsymbol{T}_{low}$, directly.) After all, just by looking at the definition, it seems G_i depends on Q_i, which is controlled by Eve. Interestingly enough, indeed $\boldsymbol{G} \sim \boldsymbol{\mathcal{G}}$. To understand this intuitively, consider the following simple game played between Alice and Eve. In each round, Alice tosses a fair coin but does not reveal it to Eve (this coin plays similar role as $T^{(j)}$). However, Eve can decide whether to flip the coin or not (this is like permuting channel assignments according to q). Finally, the coin is revealed and the game continues into the next round. Now, a simple but important observation is: the coin is still a fair coin in each round, although Eve can decide whether to flip it or not. Similarly, back to our setting, we can show $\boldsymbol{G} \sim \boldsymbol{\mathcal{G}}$.

We continue to introduce process β'. In β', still there are n nodes executing MultiCastAdp, along with a jamming adversary Carlo. However, for each slot i, if in β nodes use $\Psi_{Q_i}(T_{hi}^{(j)})$ (resp., $\Psi_{Q_i}(T_{lo}^{(j)})$) to determine their behavior, then in β' nodes directly use $T_{hi}^{(j)}$ (resp., $T_{lo}^{(j)}$), and Carlo leaves channels $\{1, 2, \cdots, x_1 C\}$ unjammed (resp., jams all channels).

Finally, in γ, again there are n nodes executing MultiCastAdp, yet the adversary uses a fixed strategy: in the first $y_1 R$ slots, channels $\{1, 2, \cdots, x_1 C\}$ are unjammed; and in the remaining $(1 - y_1)R$ slots, all channels are jammed. Besides, in the i^{th} slot, nodes directly use chunk $T_{hi}^{(i)}$ to compute their behavior if $i \le y_1 R$, and use chunk $T_{lo}^{(i-y_1 R)}$ otherwise.

We are now ready to sketch the proof of Lemma 5. (Complete proofs can be found in the full version of the paper.)

Proof sketch of Lemma 5. Define $\mathcal{E}_X$ (respectively, $\mathcal{E}_{X'}$, and $\mathcal{E}_Y$) be the event that some node is still uninformed by the end of process β (respectively, β', and γ). Let $\mathcal{E}_{\geq}$ be event $\mathcal{E}^{\geq y_1}(\geq x_1)$. The following claims capture the relationship between these events:

Claim I: $\mathcal{E}_X$ implies $\mathcal{E}_{X'}$. Consider a slot i and two nodes u and v, assume in β node u broadcasts on channel ch and node v listens on ch. Then, by definition of our permutation function π, in that same slot in β', u must broadcast on $\pi_{Q_i}^{-1}(ch)$ and v must listen on $\pi_{Q_i}^{-1}(ch)$. We argue a failed transmission attempt on ch in β will also fail on $\pi_{Q_i}^{-1}(ch)$ in β': (a) if some third node w also broadcasts on ch in slot i in β, then w must also broadcast on $\pi_{Q_i}^{-1}(ch)$ in slot i in β'; (b) if Eve jams ch in slot i in β, then Carlo must also jam $\pi_{Q_i}^{-1}(ch)$ in slot i in β'. Thus, assuming v is uninformed in both β and β' at the beginning of slot i, then by the end of slot i, if v is still uninformed in β, it must be the case that v is also uninformed in β'. A simple induction immediately leads to the claim.

Claim II: $(\mathcal{E}_{X'} \wedge \mathcal{E}_{\geq})$ implies $\mathcal{E}_Y$. If $\mathcal{E}_{\geq}$ happens in β, then in both β' and γ, Eve leaves channels $\{1, 2, \cdots, x_1 C\}$ unjammed in $y_1 R$ slots, and jams all channels in remaining slots. Observe that we can ignore the slots in which all channels are jammed; and in each remaining slot, nodes' behavior and channel feedback are identical in the two processes.

Therefore, $\Pr[\mathcal{E}_X \wedge \mathcal{E}_{\geq}] \le \Pr[\mathcal{E}_Y]$. The effectiveness of the epidemic broadcast scheme is easy to demonstrate in process γ, as the jamming strategy of the adversary in γ is not adaptive. Specifically, we conclude $\Pr[\mathcal{E}_Y] \le \exp(-\Theta(i \cdot \lg n))$. ◀

Competitiveness and Correctness. We prove two other key lemmas in this part. The first one shows Eve cannot stop nodes from halting without spending a lot of energy, thus guaranteeing the resource competitiveness of the termination mechanism.

▶ **Lemma 6.** *Fix an epoch i and a node u, assume u is alive at the beginning of this epoch. By the end of this epoch, with probability at most $\exp(-\Theta(i \cdot \lg^2 n))$, the following two events happen simultaneously: (a) $\mathcal{E}^{\geq y_2}(\geq x_2)$ occurs during the epoch; and (b) node u does not halt. Here, $x_2 = y_2 = 0.99$, and $\mathcal{E}^{\geq y_2}(\geq x_2)$ is defined in Section 2.*

Proof sketch. Arrange the randomness of nodes as what we do in the proof of Lemma 5, except that we use parameter $x_2 = 0.99$ to replace x_1, and $y_2 = 0.99$ to replace y_1. Let R be the length of the epoch, p be nodes' working probability, and $\mathcal{E}_{\geq}$ be event $\mathcal{E}^{\geq y_2}(\geq x_2)$. Define X_i (respectively, X'_i, and Y_i) be an indicator random variable taking value one iff u hears silence in the i^{th} slot in β (respectively, β', and γ). Following random variables are what we intend to couple: $X = \sum_{i=1}^{R} X_i$, $X' = \sum_{i=1}^{R} X'_i$, and $Y = \sum_{i=1}^{R} Y_i$. Specifically:

Claim I: For any integer $t \ge 0$, $\Pr[X \le t] \le \Pr[X' \le t]$. Similar to the proof of Claim I in the proof of Lemma 5, for each slot, if in that slot u hears silence in β', then by definition of our permutation function π and the construction of β and β', it must be the case that u also hears silence in β. Thus, $X'_i = 1$ implies $X_i = 1$, resulting in $X' \le X$.

Claim II: For any integer $t \ge 0$, $\Pr[(X' \le t) \wedge \mathcal{E}_{\geq}] \le \Pr[Y \le t]$. If $\mathcal{E}_{\geq}$ happens in β', then in both β' and γ, Eve leaves channels $\{1, 2, \cdots, x_2 C\}$ unjammed for $y_2 R$ slots, and jam all channels in other slots. Note that in each of the R slots, nodes' behavior are independent and are sampled from an identical distribution, so the indices of the $y_2 R$ slots does not matter.

Therefore, $\Pr[(X \le t) \wedge \mathcal{E}_{\ge}] \le \Pr[Y \le t]$. Since $\{Y_1, Y_2, \cdots, Y_R\}$ is a set of mutually independent random variables, bounding $\Pr[Y < Rp/2]$ is easy. Specifically, $\mathbb{E}[Y] = y_2 \cdot x_2 \cdot p \cdot (1-p/C)^{n-1} \ge 0.99^2 \cdot Rp \cdot (1-p/C)^n \ge 0.99^2 \cdot Rp \cdot e^{-2np/C} \ge 0.99^2 \cdot Rp \cdot e^{-0.5} > 0.59Rp$. Apply a Chernoff bound, we know $\Pr[Y < Rp/2] \le \exp(-\Theta(Rp)) \le \exp(-\Theta(i \cdot \lg n))$. ◀

The second lemma states that all nodes must have been informed before any node decides to halt, thus message dissemination must have completed before any node stops execution. To prove the lemma, we consider two complement cases: either Eve jams a lot in the epoch, or she does not. If jamming is not strong, Lemma 5 implies no node remains uninformed. Otherwise, u should not hear a lot of silent slots and will not halt. Notice, handling the strong jamming case also relies on the coupling technique.

▶ **Lemma 7.** *Fix an epoch i in which all nodes are active, fix a node u. By the end of this epoch, with probability at most $\exp(-\Theta(i \cdot \lg n))$, the following two events happen simultaneously: (a) node u halts; and (b) some node is still uninformed.*

Main theorem. We sketch the proof of Theorem 1 in this last part.

Fix a node u, we begin by computing how long u remains active. Let L be the total runtime of u. Since epoch length increases geometrically, we only need to focus on the last epoch in which u is active. Also, notice that Lemma 6 suggests Eve must jam a lot in an epoch – the amount of which can be described as some function of epoch length – to stop u from halting. Putting these pieces together, we show $\Pr(L > \Theta(1) \cdot T/C) \le n^{-\Omega(1)}$. By a union bound, we know when $T = \Omega(C)$ w.h.p. all nodes halt within $O(T/C)$ slots.

Next, we analyze the cost of nodes. Again fix a node u, let F denote its total cost. By an argument similar to above, we are able to prove $\Pr(F > \Theta(\lg n) \cdot \sqrt{\lg T \cdot (T/n)}) \le n^{-\Omega(1)}$. By a union bound, we know when $T = \Omega(C)$ w.h.p. the cost of each node is $O(\sqrt{T/n} \cdot \sqrt{\lg T} \cdot \lg n)$.

The last step is to show with high probability each node must have been informed when it halts, and this can be proved via an application of Lemma 7.

Finally, we note that when $T = o(C)$, all nodes will halt by the end of the first epoch, with high probability. This results in the τ_{time} and τ_{cost} terms in the theorem statement.

5 The MultiCastAdvAdp Algorithm

Our second algorithm – called MultiCastAdvAdp– works even if knowledge of n is absent. However, its design and analysis are much more involved than that of MultiCastAdp.

Building MultiCastAdvAdp. When the value of n is unknown, the principal obstacle lies in properly setting nodes' working probabilities. In view of this, we let MultiCastAdvAdp contain multiple *super-epochs*, each of which contains multiple *phases*, and nodes may use different working probabilities in different phases. Notice, for each super-epoch, we need to ensure it contains sufficiently many "good" phases, in the sense that within each such good phase broadcast will succeed if Eve does not heavily jam it. Another challenge posed by the unknown n value is that the simple termination criterion – large fraction of silent slots – no longer works, as this can happen when the working probability is too low.

Gilbert et al. [15] provide a solution to the above two challenges in the single-channel setting. Specifically, at the beginning of a super-epoch i, nodes set their initial working probability to a pre-defined small value. After each phase, each node u increases its working probability p_u by a factor of $2^{\max\{0, \eta_u - 0.5\}/i}$, where η_u denotes the fraction of silent slots u observed within the phase. This mechanism provides two important advantages: (a) Eve has to keep jamming heavily to prevent p_u from reaching the ideal value; and (b) p_u and p_v might be different for two nodes u and v, but the difference is bounded. As for termination, the number of messages nodes heard could be a good metric. However, a simple threshold would not work. Instead, Gilbert et al. develop a two-stage termination mechanism: when a node u hears the message sufficiently many times, it becomes a `helper` and obtains an estimate of n; Later, when u is sure that all nodes have become `helper`, it will stop execution.

In MultiCastAdvAdp, we extend the above approach to the multi-channel setting. Specifically, we observe that the single-channel message dissemination scheme used in [15] is relatively slow in that it needs $\Theta(\lg n)$ phases to accomplish broadcast. By contrast, in MultiCastAdvAdp, the application of epidemic broadcast reduces this time period to a single weakly-jammed phase. This replacement is not a simple cut-and-paste. Instead, we also adjust the phase structure accordingly. In particular, each phase now contains two *steps*. This adjustment further demands us to change the way nodes' update their working probabilities after each phase: $p_u \leftarrow p_u \cdot 2^{\max\{0,\eta_u^{step1}+\eta_u^{step2}-1.5\}}$. In the end, MultiCastAdvAdp provides a slightly better resource competitive ratio than [15].

Handing adaptivity via coupling also becomes more challenging. In more detail, in each phase we need the number of silent slots u heard N_u to be close to its expectation for *any* jamming results vector (instead of, say, only when jamming is strong, as in the proof of Lemma 7). To acquire the desired results, we have to consider jamming results vectors at a much finer level (rather than a single category, as in the proof of Lemma 6 and Lemma 7), which in turn requires the failure probability for each category to be much lower (otherwise a union bound over the increased number of categories would not work). Allowing N_u to have larger deviation from its expectation solves the issue, but it further demands the initial working probability nodes used at the beginning of each epoch to be sufficiently high. Unfortunately, this increased initial working probability could result in nodes becoming `helper` with incorrect estimates of n, violating the correctness of the termination mechanism. We fix this problem by adding step three to each phase: observing the fraction of silent slots in step three allows nodes to determine the reliability of their estimates.

Algorithm description. MultiCastAdvAdp contains multiple super-epochs, and the first super-epoch number is $I_b = 2\lg C + 20$. In super-epoch i, there are bi phases numbered from 0 to $bi-1$, where b is some large constant. Each phase contains three steps. For any super-epoch i, the length of each step is always $R_i = a \cdot 2^i \cdot i^3$, where a is some large constant. Prior to execution, all nodes are in `init` status. Similar to MultiCastAdp, each node u maintains M_u to indicate whether it knows the message m or not.

We now describe nodes' behavior in each (i,j)-phase – i.e., phase j of super-epoch i – in detail. For each slot in an (i,j)-phase, each node will go to a channel chosen uniformly at random. Then, for each node u, it will broadcast or listen on the chosen channel, each with a certain probability. In step one and two, this probability is $p_u^{i,j}$; in step three, this probability is $p_{step3}^i = C^2/2^i$. We often call $p_u^{i,j}$ as the working probability of node u. Notice, at the beginning of an super-epoch i, the probability $p_u^{i,j}$, which is just $p_u^{i,0}$, is set to $C/2^i$. In a slot, if u chooses to send, then the broadcast content depends on the value of M_u: if M_u is *true* then u will broadcast m, otherwise u will broadcast a beacon message ±. On the other hand, if u chooses to listen in a slot, then it will record the channel feedback. One point worth noting is, a node u will only change M_u from $false$ to $true$ if it hears message m in step one. (The purpose of this somewhat strange behavior is to facilitate analysis.)

At the end of each phase j, nodes will compute $p_u^{i,j+1}$ (i.e., the working probability of the next phase). Specifically, for each node u, define $\Delta_u^{step1} = \Delta_u^{step2} = R_i p_u^{i,j}/(1-p_u^{i,j}/C)$ and $\Delta_u^{step3} = R_i p_{step3}^i/(1-p_{step3}^i/C)$. Let $N_u^{step1,c}$, $N_u^{step2,c}$, and $N_u^{step3,c}$ denote the number of silent slots u observed in step one, step two, and step three in phase j, respectively. Then, $\eta_u^{i,j} = N_u^{step1,c}/\Delta_u^{step1} + N_u^{step2,c}/\Delta_u^{step2} + N_u^{step3,c}/\Delta_u^{step3}$, and $p_u^{i,j+1} = p_u^{i,j} \cdot 2^{\max\{0,\eta_u^{i,j}-2.5\}}$.

At the end of each phase j, nodes will also potentially change their status. Specifically, if a node u is in `init` status and finds: (a) $\eta_u^{i,j} \geq 2.4$; and (b) it has heard the message m at least ai^3 times during step two of phase j. Then, node u will become `helper` and compute an estimate of n as $n_u = C/((p_u^{i,j})^2 \cdot 2^i)$. On the other hand, if u is already a `helper` and finds $p_u^{i,j+1} \geq 64\sqrt{C/(2^i \cdot n_u)}$, then u will change its status to `halt` and stop execution. Complete pseudocode of MultiCastAdp is provided in the full version of the paper.

6 Analysis of MultiCastAdvAdp

Throughout the analysis, when considering an (i,j)-phase, we often omit the indices i and/or j if they are clear from the context. For any node u, we often use p_u to denote its working probability in a step. We always use V to denote active nodes, and M to denote active nodes with $M_u = \textit{true}$. Omitted proofs and auxiliary lemmas are provided in the full paper.

The "bounded difference" property. The main goal of this part is to show nodes' working probabilities can never differ too much. This "bounded difference" property is used extensively in remaining analysis, either explicitly or implicitly.

▶ **Lemma 8.** *Consider a super-epoch $i > \lg n$. With probability at least $1 - \exp(-\Theta(iC))$, we have $1/2 \leq p_u/p_v \leq 2$ for any two nodes u and v at any phase of the super-epoch.*

At a high level, the above lemma holds because the fraction of silent slots nodes observed during a phase cannot differ too much. To prove it formally, we show the following claim via a coupling argument. However, details of the coupling differ from the ones we saw in Section 4. Specifically, we divide jamming results vectors into $\binom{R+C}{C}$ categories.

▷ Claim 9. Consider a step of length R and two active nodes u and v. Let p_u (resp., p_v) be the sending/listening probabilities of u (resp., v); and let X_u (resp., X_v) be the number of silent slots u (resp., v) observed. Define $\Delta_u = Rp_u/(1-p_u/C)$ and $\Delta_v = Rp_v/(1-p_v/C)$. Define $\chi_u = \sqrt{giC/(Rp_u)}$ and $\chi_v = \sqrt{giC/(Rp_v)}$, where $g \leq a/20$ is a constant. Then:

1. $\Pr[X_u/\Delta_u > 1] \leq \exp(-\Theta(i^3C))$.
2. $\Pr[(X_u/\Delta_u > 0.2) \wedge (X_v/\Delta_v < 0.1)] \leq \exp(-\Theta(i^3C))$.
3. $\Pr[(|X_u/\Delta_u - X_v/\Delta_v| \geq \chi_u + \chi_v) \wedge (X_u/\Delta_u \geq 0.1) \wedge (X_v/\Delta_v \geq 0.1)] \leq \exp(-\Theta(iC))$.

Proof sketch. We begin with part (1). Define $\alpha = \prod_{w \in V}(1-p_w/C)$. To make X_u as large as possible, assume Eve does no jamming, thus whether u hears silence are independent among different slots. Notice that $\mathbb{E}[X_u] = p_u \cdot (\prod_{v \in V \setminus \{u\}}(1-p_v/C)) \cdot R = \alpha \cdot \Delta_u < \Delta_u$. Therefore, by a Chernoff bound, the probability that $X_u > \Delta_u$ is at most $\exp(-\Theta(\Delta_u)) = \exp(-\Omega(i^3C))$.

Proofs for part (2) and (3) both rely on coupling, and we only focus on part (2) here.

We first setup the coupling. Assume the randomnesses of nodes come from C lists $(\boldsymbol{T}_0, \cdots, \boldsymbol{T}_C)$. Specifically, for each slot i in the step, if the jamming result is $Q_i \subseteq [C]$, then nodes' behavior in this slot is determined by $\Psi_{Q_i}\left(T_{|Q_i|}^{(\sum_{j \leq i} \mathbb{I}[|Q_j|=|Q_i|])}\right)$ using permutation π_{Q_i} and bijection Ψ_{Q_i}. Notice, π_{Q_i} and Ψ_{Q_i} are defined in Section 4 on page 7, and $T_{|Q_i|}^{\sum_{j \leq i} \mathbb{I}[|Q_j|=|Q_i|]}$ is the $(\sum_{j \leq i} \mathbb{I}[|Q_j| = |Q_i|])$-th chunk in list $\boldsymbol{T}_{|Q_i|}$. Let $X_{u,i}$ be an indicator random variable taking value 1 iff u hears silence in the i^{th} slot, define $X_u = \sum_{i=1}^{R} X_{u,i}$.

Define $\mathcal{Z} = \{\boldsymbol{z} = \langle z_1, z_2, \cdots, z_C \rangle \in \mathbb{N}^C : \sum_{l=1}^{C} z_l \leq R\}$, thus $|\mathcal{Z}| = \binom{R+C}{C} \leq (R+1)^C \leq (2R)^C$. (Intuitively, for every $l \in [C]$, z_l in $\boldsymbol{z}$ is the number of slots in which Eve leaves

l channels unjammed.) Denote the jamming results of this step as $\boldsymbol{Q} = (Q_1, \cdots, Q_R) \in \mathcal{Q} = (2^{[C]})^R$, and define $|\boldsymbol{Q}| = \sum_{i=1}^{R} |Q_i|$. Further define function $K : \mathcal{Q} \to \mathcal{Z}$ such that $K(\boldsymbol{Q}) = \langle K_1(\boldsymbol{Q}), \cdots, K_C(\boldsymbol{Q})\rangle$, where $K_l(\boldsymbol{Q}) = \sum_{i=1}^{R} \mathbb{I}[|Q_i| = l]$. (That is, $K_l(\boldsymbol{Q})$ counts the number of slots in which Eve leaves l channels unjammed.) Hence, given $K(\boldsymbol{Q})$, we can use a function $L : \mathcal{Z} \to \mathbb{N}$ to compute $|\boldsymbol{Q}|$. In particular, $L(\boldsymbol{z}) = \sum_{l=1}^{C} z_l \cdot l$ and $L(K(\boldsymbol{Q})) = |\boldsymbol{Q}|$.

Now, consider another execution, for any $j \geq 1$ and $l \in [C]$, let $Y_{u,l}^{(j)}$ be an indicator random variable taking value 1 iff u hears silence in a slot in which the jamming result is $[l]$ and the behavior of nodes is determined by the j^{th} chunk of $\boldsymbol{T}_l$ directly. Define $Y_u(\boldsymbol{z}) = \sum_{l=1}^{C} \sum_{j=1}^{z_l} Y_{u,l}^{(j)}$ for any $\boldsymbol{z} \in \mathcal{Z}$. By definition, it is easy to verify $X_u(\boldsymbol{Q}) = Y_u(K(\boldsymbol{Q}))$ for any $\boldsymbol{Q}$. That is, for any $\boldsymbol{Q}$, values of X_u and Y_u are identical. The significance of this observation is that it relates X_u – which counts the number of silent slots u heard – to Y_u, and Y_u can be interpreted as the sum of independent random variables once $\boldsymbol{z}$ is fixed.

Now we are ready to prove part (2). Notice $\mathbb{E}[X_u]/\Delta_u = \mathbb{E}[X_v]/\Delta_v = \alpha \cdot |\boldsymbol{Q}|/(RC)$. Also, it is easy to verify $\mathbb{E}[Y_u(\boldsymbol{z})]/\Delta_u = \mathbb{E}[Y_v(\boldsymbol{z})]/\Delta_v = \alpha \cdot L(\boldsymbol{z})/(RC)$. Let $\mathcal{Z}_1 = \{\boldsymbol{z} \in \mathcal{Z} : L(\boldsymbol{z}) \leq 0.15RC/\alpha\}$. Then for $\boldsymbol{z} \in \mathcal{Z}_1$, $\mathbb{E}[Y_u(\boldsymbol{z})] \leq 0.15\Delta_u$, further by a Chernoff bound, $\Pr[Y_u(\boldsymbol{z}) > 0.2\Delta_u] \leq \exp(-\Theta(i^3C))$. Similarly, for $\boldsymbol{z} \in \mathcal{Z} \setminus \mathcal{Z}_1$, $\Pr[Y_v(\boldsymbol{z}) < 0.1\Delta_v] \leq \exp(-\Theta(i^3C))$. Therefore, we can conclude $\Pr[X_u(\boldsymbol{Q}) > 0.2\Delta_u \wedge X_v(\boldsymbol{Q}) < 0.1\Delta_v] \leq \left(\sum_{\boldsymbol{z}\in\mathcal{Z}_1} \Pr[Y_u(\boldsymbol{z}) > 0.2\Delta_u]\right) + \left(\sum_{\boldsymbol{z}\in\mathcal{Z}\setminus\mathcal{Z}_1} \Pr[Y_v(\boldsymbol{z}) < 0.1\Delta_v]\right) \leq |\mathcal{Z}| \cdot \exp(-\Theta(i^3C)) = \exp(-\Theta(i^3C))$. ◀

We now sketch the proof of Lemma 8. Denote the working probabilities of the current phase and the next phase as p and p'. If $\eta_u \leq 2.5$ and $\eta_v \leq 2.5$, then $p'_u/p'_v = p_u/p_v$ and we are done. So assume $\eta_u > 2.5$. In such case, Claim 9 imply $|N_u^{c,step*}/\Delta_u^{step*} - N_v^{c,step*}/\Delta_v^{step*}| \leq \sqrt{giC/(Rp_u)} + \sqrt{giC/(Rp_v)}$ for any step $*$ in $\{1, 2\}$, and $|N_u^{c,step3}/\Delta_u^{step3} - N_v^{c,step3}/\Delta_v^{step3}| \leq 2\sqrt{giC/(Rp_{step3})}$. This further suggests $p'_u/p'_v \leq (p_u/p_v) \cdot 2^{1/bi}$, thus the lemma is proved.

Correctness. This part shows MultiCastAdvAdp enforces two nice properties. First, when some node halts, all nodes must have become `helper`. This property can be seen as a stronger version of Lemma 7, since a node must have heard the message m when becoming a `helper`. The second property, on the other hand, states that when a node becomes `helper`, it also obtains a good estimate of n. This property helps to ensure nodes can stop execution at the right time.

▶ **Lemma 10** ("halt-imply-helper" property). *The probability that some node has stopped execution while some other node has not become* ***`helper`*** *is at most* $n^{-\Omega(1)}$.

▶ **Lemma 11** ("good-estimate" property). *For each node u, the probability that u becomes* ***`helper`*** *with $n_u < n/256$ or $n_u > 4n$ is at most* $n^{-\Omega(1)}$.

The following lemma is helpful for proving both of the above two properties. Roughly speaking, this lemma states that if in an (i, j)-phase some node u has working probability $p_u = \Theta(\sqrt{C/(2^in)})$ and decides to raise p_u at the end of the phase, then all nodes must have heard the message many times in step two of the phase.

▶ **Lemma 12.** *Consider an (i, j)-phase where $i > \lg n$. Assume at the beginning of the phase: $(\sum_{u\in V} p_u)/C \leq 1/2$, all nodes are active and their working probabilities are within a factor of two, and the working probability of each node is at least $8\sqrt{C/(2^in)}$. Then, with probability at most $\exp(-\Theta(i^2))$, these two events both occur: (a) some node raises its working probability at the end of the phase; and (b) some node hears message m less than ai^3 times in step two.*

Proof sketch. Let $\mathcal{E}_R$ be the event that some node raises its working probability at the end of the phase, $\mathcal{E}_M$ be the event that some node hears m less than ai^3 times during step two, $\mathcal{E}_{un}$ be the event that some node is still uninformed by the end of step one. Moreover, let $\mathcal{E}_1$ (respectively, $\mathcal{E}_2$) be the event that $\mathcal{E}^{\geq 0.25}_{step1}(\geq 0.25)$ (respectively, $\mathcal{E}^{\geq 0.25}_{step2}(\geq 0.25)$) occurs during step one (respectively, step two) of the phase. We know:

$$\begin{aligned}\Pr(\mathcal{E}_M\mathcal{E}_R) \leq& \Pr(\mathcal{E}_M \wedge (\mathcal{E}_1 \wedge \mathcal{E}_2)) + \Pr(\mathcal{E}_R \wedge \overline{(\mathcal{E}_1 \wedge \mathcal{E}_2)})\\ \leq& \Pr(\mathcal{E}_{un}\mathcal{E}_1) + \Pr(\overline{\mathcal{E}_{un}}\mathcal{E}_M\mathcal{E}_2) + \Pr(\mathcal{E}_R \wedge (\overline{\mathcal{E}_1} \vee \overline{\mathcal{E}_2}))\end{aligned}$$

The reminder of the proof bounds the three probabilities in the last line.

Claim I: $\Pr(\mathcal{E}_{un}\mathcal{E}_1) \leq \Pr(\mathcal{E}_{un}|\mathcal{E}_1) \leq \exp(-\Theta(i^2))$. If $\mathcal{E}_1$ happens, then step one is not heavily jammed. Thus every node will be informed at the end of step one due to the effectiveness of the epidemic broadcast scheme, much like the proof of Lemma 5.

Claim II: $\Pr(\overline{\mathcal{E}_{un}}\mathcal{E}_M\mathcal{E}_2) \leq \Pr(\mathcal{E}_M\mathcal{E}_2|\overline{\mathcal{E}_{un}}) \leq \exp(-\Theta(i^3))$. Fix a node u, and assume all nodes know m at the beginning of step two. Similar to the proof of Lemma 6 (except that we focus on message slots and apply the coupling argument accordingly), the probability that u hears m less than ai^3 times during a step two in which $\mathcal{E}_2$ occurs is at most $\exp(-\Theta(i^3))$. Take a union over all nodes and the claim is proved.

Claim III: $\Pr(\mathcal{E}_R \wedge (\overline{\mathcal{E}_1} \vee \overline{\mathcal{E}_2})) \leq \exp(-\Theta(i^3C))$. Notice that $\Pr(\mathcal{E}_R \wedge (\overline{\mathcal{E}_1} \vee \overline{\mathcal{E}_2})) \leq \Pr(\mathcal{E}_R\overline{\mathcal{E}_1}) + \Pr(\mathcal{E}_R\overline{\mathcal{E}_2}) \leq \sum_{u\in V}\Pr(\mathcal{E}_{u,1}\overline{\mathcal{E}_1}) + \sum_{u\in V}\Pr(\mathcal{E}_{u,2}\overline{\mathcal{E}_2}) + 4\sum_{u\in V}\exp(-\Theta(i^3C))$. Here, $\mathcal{E}_{u,1}$ (respectively, $\mathcal{E}_{u,2}$) is the event that node u hears silence more than $\Delta_u^{step1}/2$ (respectively, $\Delta_u^{step2}/2$) times in step one (respectively, step two) of the phase, and the last inequality is due to part (1) of Claim 9. When $\overline{\mathcal{E}_1}$ occurs, the expected number of silent slots heard by u in step one is at most $7/16\Delta_u^{step1}$. Again via a coupling argument, we know $\Pr(\mathcal{E}_{u,1}\overline{\mathcal{E}_1}) \leq \exp(-\Theta(i^3))$, and bounding $\Pr(\mathcal{E}_{u,2}\overline{\mathcal{E}_2})$ is similar. ◀

At this point, to prove the "halt-imply-helper" property, we only need to combine the above lemma with the following two observations. First, nodes are unlikely to become `helper` in early super-epochs, as the sending probabilities in these super-epochs are too high and nodes cannot hear enough silent slots. Second, when nodes' working probabilities in step two are too small, they will also not become `helper` as the number of messages heard is not enough. Notice, this second observation also leads to an upper bound on the estimates of n. (Detailed proofs of the two observations can be found in the full paper.)

To prove the "good-estimate" property, what remains is to show a lower bound for n_u. To that end, we show if all nodes are alive and u's working probability is close to the ideal value $\Theta(\sqrt{C/(2^in)})$, then u must have become `helper` already. (Again, see the full paper for the proof.) By then, a lower bound of n_u can be derived as a simple corollary of this claim.

Termination. This part shows nodes will quickly become `helper` and then halt once jamming is weak. (In other words, Eve cannot delay nodes unless she spends a lot of energy.) We begin by classifying phases and super-epochs into *weakly jammed* ones and *strongly jammed* ones. Specifically, call a phase weakly jammed if $\mathcal{E}^{\geq 0.95}(\geq 0.95)$ occurs for all three steps of the phase. Otherwise, if $\mathcal{E}^{>0.05}(< 0.95)$ occurs for any of the three steps, then the phase is strongly jammed. Call a super-epoch weakly jammed if at least half of the phases in the super-epoch are weakly jammed, otherwise the super-epoch is strongly jammed.

We first show, if a node's working probability has not reached the ideal value, then this probability will increase by some constant factor in a weakly jammed phase.

▶ **Lemma 13.** *Fix an (i,j)-phase where $i \geq \lg(nC)+6$, and fix an active node u satisfying $p_u^{i,j} < C/(128n)$. By the end of the phase, the following two events happen simultaneously with probability $\exp(-\Omega(iC))$: (a) the phase is weakly jammed; and (b) $p_u^{i,j+1} < p_u^{i,j} \cdot 2^{(1/10)}$.*

Building upon Lemma 13, we can prove nodes' working probabilities will reach $\tilde{p_i} = 1024\sqrt{C/(2^i n)}$ in a weakly jammed super-epoch, as there are enough weakly jammed phases.

▶ **Lemma 14.** *Fix a super-epoch* $i \geq 34 + \lg(nC)$ *and a node* u *that is active at the beginning of the super-epoch. The following two events happen simultaneously with probability* $\exp(-\Omega(iC))$*: (a) the super-epoch is weakly jammed; and (b) by the end of the super-epoch* u *is still alive with a working probability less than* $\tilde{p_i}$*.*

Lastly, we show that when a node's working probability reaches $\tilde{p_i}$, it will halt.

▶ **Lemma 15.** *Fix a super-epoch* $i \geq \lg(nC) - 7$ *and a node* u*. Assume the "halt-imply-helper" property and the "good-estimate" property both hold. Then, the probability that* u *is active at the end of super-epoch* i *with a working probability exceeding* $\tilde{p_i}$ *is at most* $\exp(-\Theta(i))$*.*

Main theorem. In this last part we sketch the proof of Theorem 2.

Fix an arbitrary node u. The first step is to analyze how long u remains active. Since super-epoch length increases geometrically, we only need to focus on the last super-epoch in which u is active. Specifically, let $\hat{I} = 34 + \lg C + \max\{\lg C, \lg n\}$, let r_i be the number of slots in super-epoch i, and let $sr_i = \sum_{k=\hat{I}+1}^{i} r_k$ be the total number of slots from super-epoch $\hat{I}+1$ to super-epoch i. It is easy to verify, for $i \geq \hat{I}+1$, $sr_i \leq 5r_{i-1}$. Define constant $\beta = 2400$, and let random variable L denote node u's actual runtime starting from super-epoch $\hat{I}+1$. Combine Lemma 10, 11, 14, 15, along with the fact that Eve spends less than $r_i C/\beta = bi/2 \cdot 0.05^2 R_i C$ energy in super-epoch i implies super-epoch i is weakly jammed, we can prove $L \leq 5\beta T/C$ holds w.h.p. Take a union bound over all nodes, we know every node will terminate within $(\sum_{k=I_b}^{\hat{I}} bk \cdot 3R_k) + 5\beta T/C = \tilde{O}(T/C + nC + C^2)$ slots, w.h.p.

Next, we analyze the cost of node u. Let $F_{step1,2}$ (resp., F_{step3}) be node u's total actual cost during step one and step two (resp., step three) in all phases starting from super-epoch $\hat{I}+1$. By an analysis similar to above, we show $F_{step1,2} \leq \Theta(\sqrt{T/n} \cdot \lg^2 T)$ and $F_{step3} \leq \Theta(C^2 \cdot (\hat{I} + \lg T)^5)$, w.h.p. As a result, we can conclude w.h.p. the energy cost of each node is bounded by $F_{step1,2} + F_{step3} + \sum_{k=I_b}^{\hat{I}}(bk \cdot 3R_k) = \tilde{O}(\sqrt{T/n} + nC + C^2)$.

Finally, notice the algorithm itself ensures a node must be informed when it halts.

7 Lower Bounds

In this section, we show our algorithms achieve (near) optimal time and energy complexity simultaneously against an adaptive adversary with budget T. The time complexity part is obvious: Eve can jam all channels during the first T/C slots, so the $O(T/C)$ term in the runtime of MultiCastAdp and MultiCastAdvAdp is asymptotically optimal.

Obtaining an energy complexity lower bound is much more involved. To do so, the first step is a simulation argument. Specifically, given any fair multi-channel broadcast algorithm $\mathcal{A}_n$, we can devise a multi-channel 1-to-1 communication algorithm $\mathcal{A}_2$ (in which the goal is to let one node called Alice to send a message m to another node Bob) that simulates $\mathcal{A}_n$ internally. To make the simulation feasible, we allow Alice and Bob to have multiple transceivers, so that in each slot they can operate on multiple channels, as well as send and listen simultaneously. In more detail, Alice in $\mathcal{A}_2$ mimics the source node in $\mathcal{A}_n$. As for Bob, he simulates the $n-1$ non-source nodes in $\mathcal{A}_n$. Particularly, in each slot, for each channel, if at least one non-source node listens, then Bob uses a transceiver to listen; if exactly one non-source node broadcasts, then Bob uses a transceiver to broadcast the unique message; and if at least two non-source nodes broadcast, then Bob uses a transceiver to broadcast noise. (Notice Bob can simultaneously listen and broadcast on a channel: he uses two transceivers and incurs two units of cost.) On the other hand, Eve's strategy for disrupting $\mathcal{A}_n$ and $\mathcal{A}_2$ is called $\mathcal{S}$: in each slot, for each channel, Eve jams it iff the probability that the source node (respectively, Alice) successfully transmits m to some non-source node (respectively, Bob) over this channel exceeds $1/T$.

Clearly, the above simulation is "perfect": an execution of $\mathcal{A}_2$ is identical to an execution of $\mathcal{A}_n$, assuming nodes use identical random bits in the two executions. To simplify presentation, we further assume $\mathcal{A}_n$ automatically stops once all nodes are informed, and $\mathcal{A}_2$ automatically stops once Bob is informed. This modification will not increase nodes' energy cost, thus will not affect the correctness of our lower bound. Now, observe that the success of $\mathcal{A}_2$ is a necessary condition for the success of $\mathcal{A}_n$, and Bob's energy cost will not exceed the sum of all non-source nodes' cost, hence the following lemma is immediate.

▶ **Lemma 16.** *For any fair multi-channel broadcast algorithm $\mathcal{A}_n$, there exists a multi-channel 1-to-1 communication algorithm $\mathcal{A}_2$. If in $\mathcal{A}_n$ each node incurs an expected cost of $f(T)$ and $\mathcal{A}_n$ succeeds with probability p, then: (a) in $\mathcal{A}_2$ Alice and Bob incur an expected cost of at most $f(T)$ and $n \cdot f(T)$, respectively; (b) $\mathcal{A}_2$ succeeds with probability at least p.*

What remains is an energy complexity lower bound for $\mathcal{A}_2$: with such a result, Theorem 3 is immediate via simple reduction. Indeed, we are able to prove Theorem 17, an energy complexity lower bound for 1-to-1 communication in the multi-channel setting. This result could be of independent interest, and at a high-level its proof is organized in the following way. First, we note that in a rough sense, any multi-channel 1-to-1 communication algorithm $\mathcal{A}$ can be viewed as a decision tree, and each path from the root to a leaf in the tree corresponds to an oblivious algorithm. Then, we argue that $\mathcal{A}$ can be used to generate another algorithm $\mathcal{A}'$ which is a "convex combination" (or, a distribution) of all such oblivious algorithms, without changing the success probability or the product of Alice's and Bob's expected cost. Moreover, an important observation is that among all the oblivious algorithm used in the "convex combination", at least one – say $\mathcal{A}_{\hat{w}}$ – is (roughly) as good as $\mathcal{A}'$ in terms of both success probability and energy efficiency. Finally, depending on whether Eve uses all her budget during execution, we consider two potential scenarios for $\mathcal{A}_{\hat{w}}$, and for both we show $\mathbb{E}_{\mathcal{A}_{\hat{w}}}[A] \cdot \mathbb{E}_{\mathcal{A}_{\hat{w}}}[B] \in \Omega(T)$, which in turn implies $\mathbb{E}_{\mathcal{A}}[A] \cdot \mathbb{E}_{\mathcal{A}}[B] \in \Omega(T)$. Complete proof of Theorem 17 is provided in the full paper.

▶ **Theorem 17.** *Consider any multi-channel 1-to-1 communication algorithm that succeeds with constant probability against an adaptive adversary Eve with budget T. Let A and B denote Alice's and Bob's expected cost respectively, then Eve can force $\mathbb{E}[A] \cdot \mathbb{E}[B] \in \Omega(T)$.*

References

1 John Augustine, Valerie King, Anisur Molla, Gopal Pandurangan, and Jared Saia. Scalable and secure computation among strangers: Message-competitive byzantine protocols. In *International Symposium on Distributed Computing*, DISC '20. Springer, 2020.

2 Baruch Awerbuch, Andrea Richa, and Christian Scheideler. A jamming-resistant mac protocol for single-hop wireless networks. In *Proceedings of the 27th ACM Symposium on Principles of Distributed Computing*, PODC '08, pages 45–54. ACM, 2008.

3 Reuven Bar-Yehuda, Oded Goldreich, and Alon Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *Journal of Computer and System Sciences*, 45(1):104–126, 1992.

4 M. Bender, J. Fineman, M. Movahedi, J. Saia, V. Dani, S. Gilbert, S. Pettie, and M. Young. Resource-competitive algorithms. *SIGACT News*, 46(3):57–71, 2015.

5 Yi-Jun Chang, Varsha Dani, Thomas Hayes, Qizheng He, Wenzheng Li, and Seth Pettie. The energy complexity of broadcast. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, PODC '18, pages 95–104. ACM, 2018.

6 Yi-Jun Chang, Varsha Dani, Thomas P. Hayes, and Seth Pettie. The energy complexity of bfs in radio networks. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, PODC '20, pages 27–282. ACM, 2020.

7 Yi-Jun Chang, Tsvi Kopelowitz, Seth Pettie, Ruosong Wang, and Wei Zhan. Exponential separations in the energy complexity of leader election. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC '17, pages 771–783. ACM, 2017.

8 Haimin Chen and Chaodong Zheng. Fast and resource competitive broadcast in multi-channel radio networks. In *Proceedings of the 31st ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '19, pages 179–189. ACM, 2019.

9 Haimin Chen and Chaodong Zheng. Broadcasting competitively against adaptive adversary in multi-channel radio networks. arXiv, 2020. URL: `https://arxiv.org/abs/2001.03936`.

10 Artur Czumaj and Peter Davies. Exploiting spontaneous transmissions for broadcasting and leader election in radio networks. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '17, pages 3–12. ACM, 2017.

11 Shlomi Dolev, Seth Gilbert, Rachid Guerraoui, and Calvin Newport. Gossiping in a multichannel radio network. In *International Symposium on Distributed Computing*, DISC '07, pages 208–222. Springer Berlin Heidelberg, 2007.

12 Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

13 Leszek Gasieniec, Erez Kantor, Dariusz R. Kowalski, David Peleg, and Chang Su. Energy and time efficient broadcasting in known topology radio networks. In *International Symposium on Distributed Computing*, DISC '07, pages 253–267. Springer Berlin Heidelberg, 2007.

14 Mohsen Ghaffari, Bernhard Haeupler, and Majid Khabbazian. Randomized broadcast in radio networks with collision detection. *Distributed Computing*, 28(6):407–422, 2015.

15 Seth Gilbert, Valerie King, Seth Pettie, Ely Porat, Jared Saia, and Maxwell Young. (near) optimal resource-competitive broadcast with jamming. In *Proceedings of the ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '14, pages 257–266. ACM, 2014.

16 Ramakrishna Gummadi, David Wetherall, Ben Greenstein, and Srinivasan Seshan. Understanding and mitigating the impact of rf interference on 802.11 networks. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, pages 385–396. ACM, 2007.

17 Marcin Kardas, Marek Klonowski, and Dominik Pajak. Energy-efficient leader election protocols for single-hop radio networks. In *2013 42nd International Conference on Parallel Processing*, ICPP '13, pages 399–408. IEEE, 2013.

18 Valerie King, Seth Pettie, Jared Saia, and Maxwell Young. A resource-competitive jamming defense. *Distributed Computing*, 31(6):419–439, 2018.

19 Valerie King, Jared Saia, and Maxwell Young. Conflict on a communication channel. In *Proceedings of the 30th ACM Symposium on Principles of Distributed Computing*, PODC '11, pages 277–286. ACM, 2011.

20 Dominic Meier, Yvonne Anne Pignolet, Stefan Schmid, and Roger Wattenhofer. Speed dating despite jammers. In *International Conference on Distributed Computing in Sensor Systems*, DCOSS '09, pages 1–14. Springer Berlin Heidelberg, 2009.

21 Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, pages 364–369. IEEE, 2005.

22 Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. A jamming-resistant mac protocol for multi-hop wireless networks. In *International Symposium on Distributed Computing*, DISC '10, pages 179–193. Springer Berlin Heidelberg, 2010.