A Resource-Competitive Jamming Defense[*]

Valerie King[1], Seth Pettie[2], Jared Saia[3], and Maxwell Young[4]

[1]Dept. of Computer Science, University of Victoria, BC, Canada
val@cs.uvic.ca
[2]Dept. of Electrical Engineering and Computer Science,University of Michigan, MI, USA
pettie@umich.edu
[3]Dept. of Computer Science, University of New Mexico, NM, USA
saia@cs.unm.edu
[4]Computer Science and Engineering Dept., Mississippi State University, MS, USA Email:
myoung@cse.msstate.edu

Abstract

Consider a scenario where Alice wishes to send a message m to Bob in a time-slotted wireless network. However, there exists an adversary, Carol, who aims to prevent the transmission of m by jamming the communication channel. There is a per-slot cost of 1 to send, receive or jam m on the channel, and we are interested in how much Alice and Bob need to spend relative to Carol in order to guarantee communication.

Our approach is to design an algorithm in the framework of ***resource-competitive analysis*** where the cost to correct network devices (i.e., Alice and Bob) is parameterized by the cost to faulty devices (i.e., Carol). We present an algorithm that guarantees the successful transmission of m and has the following property: if Carol incurs a cost of T to jam, then both Alice and Bob have a cost of $O(T^{\varphi-1}+1) = O(T^{.62}+1)$ in expectation, where $\varphi = (1+\sqrt{5})/2$ is the golden ratio. In other words, it possible for Alice and Bob to communicate while incurring asymptotically less cost than Carol. We generalize to the case where Alice wishes to send m to n receivers, and we achieve a similar result.

Our findings hold even if (1) T is *unknown* to either party; (2) Carol knows the algorithms of both parties, but not their random bits; (3) Carol can jam using knowledge of past actions of both parties; and (4) Carol can jam reactively, so long as there is sufficient network traffic in addition to m.

[*] This research was supported in part by an NSERC Discovery Grant, and NSF grants CCF-1217338, CNS-1318294, CCF-1514383, CCF-1637546, NSF CAREER Award 0644058, CCR-0313160, and CCF-1613772.

1 Introduction

The wireless communication medium is vulnerable to ***jamming*** whereby an attacker causes interference on the communication channel. Such attacks are prohibited by US federal law, however, they occur in practice (see [18, 20]) and will likely increase in frequency with the popularity of wireless devices.

Defending against jamming is particularly important to the security of wireless low-power networks (LPNs) such as wireless sensor networks (WSNs) and the Internet of Things (IoT), where these attacks are known to degrade performance [51,81]. Such networks are especially vulnerable given that their constituent devices are often battery-powered and, therefore, energy is a scarce resource that must be conserved.

With this setting in mind, we consider the following problem: Alice wishes to guarantee transmission of a message m to Bob over a wireless communication channel. However, there exists an adversary, Carol, who aims to prevent communication by jamming transmissions over the channel. Our solution to this problem makes use of a relatively new technique called ***resource competitiveness***. Informally, the idea behind resource-competitive analysis is to include the cost of the adversary, T, as another parameter by which to measure the algorithmic performance (details are provided in Section 1.1).

We assume a time-slotted model where it costs 1 to send, listen, or jam per slot on the communication channel; these operations typically dominate the operational costs of wireless devices in terms of energy usage (see Section 6 for discussion of this issue). Our results guarantee the successful transmission of m and has the following costs: if Carol incurs a cost of T to jam, then both Alice and Bob have a cost of $O(T^{\varphi-1}+1) = O(T^{.62}+1)$ slots in expectation, where $\varphi = (1+\sqrt{5})/2$ is the golden ratio. We are able to generalize to the setting where Alice wishes to send m to n receivers, and we obtain a similar result.

In both cases, the smaller asymptotic costs to the good devices implies that it is possible to "bankrupt" a jamming adversary. That is, in attempting to prevent communication through jamming, Carol will deplete her onboard power supply far more rapidly than either Alice or Bob (or multiple receivers) and, when this occurs, communication will succeed. Note that such a result is useful even in the face of continuous jamming; this is in contrast to most prior work where typically there are constraints on how the adversary is permitted to jam (Section 1.3 discusses previous work on jamming mitigation).

Our results holds even if (1) T is *unknown*; (2) Carol knows the algorithms used by the communicating parties, but not their random bits; and (3) Carol can attack using total knowledge of past actions of the communicating parties (i.e. Carol is an ***adaptive*** adversary). Under the assumption of sufficient network traffic (in addition to m), we also demonstrate that the result holds when Carol knows the channel is in use and then decides to jam (i.e. Carol is a ***reactive*** adversary).

1.1 Resource Competitiveness

Resource competitiveness is a new approach to designing robust algorithms for distributed systems [10, 36]. We now define what it means for a distributed algorithm $\mathcal{A}$ to be *resource competitive*.

Assume a system with a set G of n good nodes that obey actions prescribed by $\mathcal{A}$. There exists an adversary who incarnates a source of disruption in the system. For example, the adversary may represent (1) any number of malicious nodes that collude and deviate arbitrarily from $\mathcal{A}$, or (2) the effects of more benign failures due to software or hardware faults.

Let $Cost(\alpha, v)$ denote the resource expenditure (or cost) to a good node v over an execution α. A resource might be bandwidth, CPU cycles, energy, actual money, or another useful domain-specific measure. $Cost(\alpha, v)$ is the cost incurred by v for executing the actions prescribed by $\mathcal{A}$ in an execution α.

Let $T(\alpha)$ be the adversary's total cost; this is typically unknown to the good nodes. We simply refer to T and $Cost(v)$ as α is implicit. We define what it means for $\mathcal{A}$ to be resource competitive:

Definition 1. *Algorithm $\mathcal{A}$ is (ρ, τ)-resource competitive if* $\max_{v \in G} \{Cost(v)\} \leq \rho(T) + \tau$ *for* $\tau > 0$ *and any execution.*

In other words, $\mathcal{A}$ is resource competitive if the *maximum* cost incurred by any good node is less than some function of the adversary's total cost, $\rho(T)$, plus some additive term τ. The function ρ is called the *robustness function* and it is a function of T and possibly other parameters such as n. In designing $\mathcal{A}$, we desire ρ to be a slow-growing function. This implies that, as the amount of disruption increases, the cost incurred by $\mathcal{A}$ increases slowly. Indeed, in dealing with jamming attacks, we show that it is possible to achieve $\rho(T) = o(T)$.

Why do we require τ? Note that when $T = 0$ — that is, there is no disruption — the good nodes clearly cannot incur less than zero cost; τ represents the unavoidable cost to attain a goal even in the absence of disruption. Efficiency when the level of disruption is low, or non-existent, is critical to efficiency. It useful to make this separation explicit via τ, which we call the *efficiency function*, and it can be a function of parameters such as n, but it is *not* a function of T.

Functions other than the maximum cost over all nodes may be appropriate depending on the context. For instance, we may consider the average or median cost. However, Definition 1 has been applied in the existing literature. Finally, we typically report results using big-O notation of the form $O(\rho(T) + \tau)$.

1.2 Resource Competitiveness in Context

It is helpful to contrast the notion of resource competitive algorithms with a number of related concepts.

Notions of Competitiveness. Competitive analysis is a well-known technique that evaluates the worst-case performance of an online algorithm relative to an optimal offline algorithm $\mathcal{OPT}$ [72]. While the inputs to an online algorithm can be viewed as adversarially selected, there is no notion of cost to the adversary for selecting certain inputs over others. In contrast, resource competitiveness places the cost to the adversary directly in the performance metric (see Definition 1). Unlike online analysis where it is impossible for an online algorithm to outperform $\mathcal{OPT}$, a resource-competitive algorithm can actually outperform the adversary by achieving $\rho(T) = o(T)$.

Game theory provides another measure of competitiveness known as the "price of anarchy" which is the ratio of the worst-case Nash equilibrium to the global social optimum [62, 71]. In resource-competitive analysis, each node either obeys the protocol or it does not; in game theory, nodes seek to maximize their respective utility functions. It is possible to address malicious behavior in the context of game theory (see [1, 2, 19, 49, 75]). The incorporation of game theoretic concepts may be an interesting direction for *future* work on resource-competitive algorithms.

Notions of Inflicting Cost. The idea of inflicting cost on an opponent arises more explicitly in the domain of cryptography. A major differentiating aspect of cryptographic approaches is that the length of a private key is decided prior to encryption. This roughly determines (i) how much the adversary must spend in order to compromise the cryptosystem, and (ii) how much the good nodes must spend to achieve a particular level of security. In contrast, resource-competitive algorithms are *adaptive* in the following sense. When $T = 0$, there is a small upfront cost quantified by the efficiency function τ. Then, as T grows, the cost function $\rho(T)$ grows commensurately and will dominate the cost of the algorithm when T grows large enough; in this sense, resource-competitive algorithms are adaptive. This is very different from having a predetermined cost.

Additional Related Ideas. An early example of considering an attacker's resources is the cryptosystem by Merkle [58] where computational puzzles are used to inflict cost on an eavesdropper. However, the hardness of the puzzles must be set *a priori* and, therefore this scheme lacks the adaptivity of resource-competitive algorithms described above. Inflicting computational cost has been used to deter spam email [30]. Another example arises in settings where an attacker controls multiple identities. In such a network, one may issue a cost for joining via computational puzzles [48, 73] or monetary penalties [17]. Similar ideas arise in proposals to mitigate applicaton-level DDoS attacks. Typically, a client must spend bandwidth or computational resources prior to receiving service (see [55, 64, 76]). In contrast to these results, resource-competitive analysis goes beyond simply inflicting cost by quantifying a relationship between the cost to the adversary and the cost to the good nodes.

1.3 Related Work on Tolerating Jamming Attacks

Several works address applied security considerations with respect to jamming [4, 8, 52, 82] where the adversary deliberately disrupts the communication medium. Defenses include spread spectrum techniques (frequency or channel hopping), mapping with rerouting, and others (see [54, 60, 80, 81]).

Recent applied work by Ashraf *et al.* [5] investigates a similar line of reasoning by examining ways in which multi-block payloads (each block in a packet has its own cyclic redundancy code), so-called "look-alike packets", and randomized wakeup times for receivers can be used to force the adversary into expending more energy in order to jam transmissions. The authors' approach is interesting and the use of look-alike packets to prevent an adversary from being able to differentiate between different types of nework traffic is similar to our approach to foiling reactive jammers (see Section 4). However, on the whole, their approach is quite different from our own; moreover, analytical results are not provided in [5].

There are a number of theoretical results on jamming adversaries. Gilbert and Young [38] give a Monte Carlo 1-to-n partial broadcast algorithm that is resource competitive. However, the result critically depends on knowing n and still allows the adversary to prevent a small, but constant, fraction of the nodes from receiving the broadcast. Removing the reliance on n is challenging, but a Monte Carlo 1-to-n broadcast algorithm is given by Gilbert et al. [35] for this case.

Similarly, the problem of interactive communication on noisy channels has been examined from a resource-competitive perspective. Results in Dani [22] and Dani *et al.* [23, 24] address an adversary that can flip an unknown number of bits as information is transmitted across a channel. However, this work differs critically in that the goal is to minimize the latency of communication rather than energy expenditure

Recently, Bender *et al.* [9] demonstrated that contention resolution – how to coordinate access to a shared channel by multiple devices – can be accomplished despite a powerful jamming adversary. Informally, the result guarantees expected constant throughput in a setting where n requests are scheduled adversarially, and each client is shown to require at most an expected polylogarithmic number of channel-access attempts

Outside of resource-competitive results, there is large body of work on mitigating jamming attacks in wireless sensor networks (see [84] for a survey). Gilbert *et al.* [34] examines the duration for which communication between two parties can be disrupted in a model with collision detection in a time-slotted network against an adversary who interferes with an unknown number of transmissions. As we do here, the authors assume channel traffic is always detectable at the receiving end (i.e. silence cannot be "forged"). The authors employ the notion of *jamming gain* which is, roughly speaking, the ratio of the duration of the disruption to the adversary's cost for causing such disruption. This is an interesting metric which can gauge the efficiency of the adversary's attack; however, it does not incorporate the cost incurred by the correct parties in the system.

Pelc and Peleg [65] examine an adversary that randomly corrupts messages. Results by Awerbuch et al. [6] and Richa et al. [68–70] consider an adversary whose jamming is bounded within any sufficiently large time window. Under this adversarial model, Ogierman et al. [63] study medium access in the signal-to-interference-plus-noise ratio communication model. In the context of Sybil attacks [29], Gilbert and Zheng [39] devise methods for tolerating a jamming adversary using coordination amongst the communicating parties, and this is generalized to a completely decentralized scenario by Gilbert, Newport and Zheng in [37].

In settings where devices have access to more than one channel, a number of theoretical results have been proposed. Dolev *et al.* [27] address a variant of the gossiping problem when multiple channels are jammed. Gilbert *et al.* [33] derive bounds on the time required for information exchange when a reactive adversary jams multiple channels. Meier *et al.* [57] examine the delay introduced by a jamming adversary for the problem of node discovery, again in a multi-channel setting. Dolev *et al.* [28] address secure communication using multiple channels with a non-reactive adversary. Recently, Dolev *et al.* [26] consider wireless synchronization in the presence of a jamming adversary. Emek and Wattenhofer [31] examine the effectiveness of frequency hopping against an adversary who jams a strict subset of the available channels.

There are also game-theoretic treatments for jamming attacks, and we refer the interested reader to the survey by Manshaei *et al.* [56] for a comprehensive treatment of this area.

The problem of robust communication has also been considered in the context of reliable broadcast where devices are laid out on a grid model [11, 13–15, 43, 44, 46, 74]. Listening costs are accounted for by King *et al.* [43, 44], but jamming adversaries are not considered. Alistarh *et al.* [3] assume collision detection and achieve authenticated reliable broadcast with the use of cryptography. With a reactive jamming adversary, Bhandhari *et al.* [16] give a reliable broadcast protocol when the amount of jamming is bounded and known *a priori*; however, correct nodes must expend considerably more energy than the adversary. Progress towards fewer broadcasts is made by Bertier *et al.* [12]; however, each node spends significant time in the costly listening state.

Finally, a preliminary version of our results appeared in [45]; however, we focus strictly on the single-hop case as this provides the most compelling results. This current version of our work contains additional exposition regarding the design of algorithm design, revised (and, we believe, clearer) versions of certain proofs, along with arguments and discussion that were previously omitted. Additionally, we consider an overlooked attack that was not addressed in [45] (discussed in Section 2.4). Finally, we note that the lower bound originally provided in [45] is not provided here, since more recent work in [35] gives a stronger bound that is asymptotically tight.

1.4 Our Model

We describe our network model and define the communication problem addressed in this work.

Las Vegas Property: Communication of m from Alice to Bob must be guaranteed with probability 1; that is, we require a Las Vegas algorithm. An obvious motivation for this Las Vegas property is a critical application, such as the dissemination of an important security update, where success is paramount.

The Las Vegas property has additional merit in multi-hop wireless networks where Monte Carlo algorithms may not be adequate. In particular, the failure probability in any single hop of a route may be small in the size of the broadcast neighborhood n. However, for large networks of total size N, it may be the case that $n \ll N$ and so a union bound over the path length may fail to offer a high-probability guarantee of correctness.

Channel Utilization: Sending or listening on the communication channel by Alice and Bob occurs in discrete units called *slots*. For example, under the common IEEE 802.11g, a slot may correspond to an actual time slot (9μs) in a time division multiple access (TDMA) type access control protocol. For simplicity, we assume that the message m fits within a single slot; otherwise, we can send m piecewise.

The cost for sending or listening is 1 per slot. This is a normalized cost meant to reflect the fact that sending and listening on the channel typically dominates the operational costs of LPN devices. For example, in the WSN setting, the send (at a transmit power of 0 dBm) and listen costs for the popular Telos motes [66] are 38mW and 35mW, respectively, and these far exceed the other operations costs for an active device. A similar relationship between the send and listen costs holds for the older, well-known MICA family of devices [21].

When Carol jams a slot, she disrupts the channel such that no communication is possible; jamming costs 1 per slot. T denotes the total amount Carol will spend over the course of the algorithm; this value is *unknown* to either Alice or Bob *a priori*.

In practice, disrupting communication within a slot may be less costly than sending a full message. However, so long as the relative costs for sending, receiving, and jamming are correct to within some (possibly large) constant factor, our asymptotic results hold. Further discussion of this issue, and other practical concerns, is provided in later in Section 6.

If two or more messages are sent within a single slot (a ***message collision***), or the slot is jammed, then the slot is said to be ***noisy***. If a slot is noisy, this is detectable by a party who is *listening* at the *receiving end* of the channel, but not by the originator of the transmission. For example, a transmission (jammed or otherwise) from Alice to Bob is detectable only by Bob; likewise, a transmission (jammed or otherwise) from Bob to Alice is detectable only by Alice. A party does not know *why* a slot is noisy; it may be due to a message collision or to jamming, but the party only learns that the channel is in use. Finally, a slot which is not noisy and does not contain a single message is said to be *clear*.

The Communicating Parties: If Alice is faulty, there is clearly no hope of communicating m; therefore, Alice is assumed to be correct. In other words, Alice can never be spoofed by the adversary Carol.

Regarding Bob, we define two cases. In Case 1, communications from Bob are always trustworthy. That is, Bob is never spoofed by Carol and we treat Carol as a separate third party. This is a benign case, and it corresponds to situations where communications sent by Bob can be trusted and jamming of m is the only obstacle.

In Case 2, Carol may spoof Bob.[1]. We emphasize that this can lead Alice to be uncertain about whether to trust Bob. This uncertainty corresponds to scenarios where a trusted dealer attempts to disseminate content to its neighbors, some of whom may be spoofed or have suffered a Byzantine fault and are used in an attempt attempt to consume resources by requesting numerous retransmissions.

The Adversary: Carol has full knowledge of past actions by Alice and Bob. This allows for *adaptive attacks* whereby Carol may alter her behaviour based on observations she has collected over time. Furthermore, under conditions discussed in Section 4, Carol can also be *reactive*: in any slot, she may detect activity on the channel and then jam; however, we assume that she cannot detect when a party is listening.

[1] Equivalently, we can consider that Carol *is* Bob, or controls Bob completely. Conceptually, the only two parties are then Alice and Carol

1.5 Design Goals for our Algorithm

In designing our resource-competitive algorithm, we have several goals. In terms of correctness, to reiterate, we want to guarantee (with probability 1) that Bob receives m.

It is also important that both Alice and Bob have termination conditions. For example, when Bob receives m, he should soon terminate such that he can either perform other network tasks, or power down to conserve energy. Similarly, when Alice is certain that Bob has received m, she should terminate; from an energy-aware perspective, it is no good to have Alice be unsure of Bob's state and, as a consequence, keep resending m in perpetuity.

We also desire the following three performance properties with regards to cost. First, we would like $\rho(T) = o(T)$ such that Alice and Bob both incur asymptotically less expected cost than Carol when T is large. Jamming attacks are effective because a correct device is often forced to incur a higher cost relative to an attacker. However, if the correct parties incur asymptotically less cost than Carol, then Alice and Bob enjoy the advantage, and Carol is faced with the problem of having her energy resources consumed disproportionately by her attempt to prevent communication.

Second, we want τ to be small. This property guarantees that the cost of running our resource-competitive algorithm is low when there is little to no attack.

Third, we want our results to be ***fair***: Alice and Bob should incur the same worst case asymptotic cost relative to Carol.

1.6 Our Results

Let $\varphi = (1 + \sqrt{5})/2$ denote the golden ratio. We also draw attention to the well-known relationship that $\Phi = \varphi - 1 = 1/\varphi = 0.618...$ where Φ is known as the *golden ratio conjugate*. Our main result is stated below.

Theorem 1. *Let Carol be an adaptive adversary that jams for T slots. There exists a resource-competitive algorithm that guarantees Bob receives m, both parties terminate, and has the following properties:*

- *In Case 1, the expected cost to Alice and Bob is $O(T^{0.5} + 1)$. In Case 2, the expected cost to Alice and Bob is $O(T^{\Phi} + 1) = O(T^{0.62} + 1)$.*
- *Both parties terminate in an expected $O(T^2)$ and $O(T^{\varphi})$ slots for Cases 1 and 2, respectively.*

In other words, we have $\rho(T) = O(T^{0.5})$ and $\rho(T) = O(T^{0.62})$ for Case 1 and Case 2, respectively, and $\tau = O(1)$ for both cases. Later, in Section 4, we demonstrate that Theorem 1 still holds when Carol is also reactive so long as there is sufficient network traffic in addition to the sending of m.

When Alice wants to send m to n receivers, a similar result is achievable. We consider the analogue to Case 2 where Carol can spoof any subset of the receivers and/or jam the communications of any/all parties. In this setting, we have the following theorem:

Theorem 2. *Let Carol be an adaptive adversary that jams for T slots. There exists a resource-competitive algorithm that guarantees all receivers obtain m, and all parties terminate, and has the following properties that hold with high probability in n:*[2]

- *The cost to Alice and Bob is $O(T^{\Phi} + \ln^{\varphi} n)$ and $O(T^{\Phi} + \ln n)$, respectively.*
- *All parties terminate within $O(T^{\varphi} + \ln^{\varphi} n)$ slots.*

Roadmap. The remainder of this paper is organized as follows. In Section 2, we proceed through the design process of our algorithm in order to motivate each step. In Section 3, we analyze our algorithm for Cases 1 and 2, and prove Theorem 1. In Section 4, we define the conditions under which we can tolerate a reactive adversary. We then demonstrate how our analysis can be amended to address a reactive adversary. In Section 5, we generalize the Alice and Bob setting to a general broadcast problem where Alice needs to transmit m to multiple receivers. In Section 6, we provide some additional motivation for our network model. Finally, we conclude with some open problems in Section 7.

2 Algorithm Design and Analysis

We incrementally build towards our resource-competitive algorithm. At each step, our design decisions are explained.

2.1 A Naive Attempt

As a first attempt, Alice and Bob can try to outspend the adversary. For example, let transmission of m be attempted over ℓ slots. In each even-indexed slot, Alice sends m while Bob listens. In each odd-indexed slot, if Bob has not received m, he sends a negative acknowledgement (`nack`) message; otherwise, Bob terminates. Alice listens in each odd-indexed slot and, if Alice receives a `nack`, she continues onto the next even-indexed slot and sends m. Similarly, if Alice detects a noisy odd-indexed slot, she interprets this as the situation where Bob sent `nack` but the

[2] With probability at least $1 - 1/n$. We will sometimes use ***w.h.p.*** as an abbreviation.

```
ITERATION-1 for any round
Send Phase: For each of $\ell$ slots do
    • Alice sends $m$ with probability $2/\sqrt{\ell}$
    • Bob listens with probability $2/\sqrt{\ell}$
If Bob received $m$, then he terminates
Nack Phase: For 1 slot do
    • Bob sends a nack message
    • Alice listens
If Alice listened to a clear slot, then she terminates
```

Figure 1: Pseudocode for ITERATION-1.

slot was jammed; therefore, she continues with the protocol. However, if Alice detects a clear odd-indexed slot, she knows Bob received m and terminated since the adversary cannot forge a clear slot; in this case, Alice safely terminates.

While Alice and Bob both finish correctly, note that if the adversary jams T consecutive even-indexed slots, then Alice and Bob each send and listen for $2T + 2$ slots. Therefore, Alice and Bob *each* spend more than twice what the adversary spends; that is, $\rho(T) > 2T$ and the adversary rapidly disables each party by depleting the respective energy supplies. This illustrates why jamming is often an effective attack.

2.2 Towards a Resource-Competitive Guarantee

An initial attempt at a resource-competitive approach is ITERATION-1 in Figure 1. In the Send Phase, Alice and Bob send and listen each with probability $2/\sqrt{\ell}$. If Bob ever receives m, he terminates the protocol. In the Nack Phase, if Bob has not terminated, he sends `nack` to Alice during the single slot asking her to enter into a new round. If Alice hears `nack`, or if the slot is jammed, she proceeds into the next round; otherwise, if the slot is clear, she terminates. Let a ***round*** refer to the execution of a Send Phase and the corresponding Nack Phase.

Using a birthday-paradox-like argument, there is likely to be a non-jammed slot where both Alice sends m and Bob listens; this is true even if a constant fraction, say $1/2$, of the slots are jammed. Therefore, communication likely succeeds unless the adversary jams more than half of the slots. In a Send Phase where more than half of the slots are jammed – referred to as a ***blocked Send Phase*** — then Bob

```
ITERATION-2 for round $i \geq 1$
Send Phase: For each of the $2^{ci}$ slots do
    • Alice sends $m$ with probability $2/2^{(c-a)i}$
    • Bob listens with probability $2/2^{(c-b)i}$
If Bob received $m$, then he terminates
Nack Phase: For $1$ slot do
    • Bob sends a nack message
    • Alice listens
If Alice listened to a clear slot, then she terminates.
```

Figure 2: Pseudocode for modified ITERATION-2.

may not receive m. But now $T = \Omega(\ell)$ while both Alice and Bob spend only $O(\sqrt{\ell}) = O(\sqrt{T})$ in expectation. Therefore, to prevent communication in the round, the adversary must incur a cost that is roughly quadratically larger. This is exactly the flavor of result that we seek.

Are we done? No, there are two shortcomings to ITERATION-1 and we describe the first here. Let us focus on the Send Phase and consider what happens if the adversary forces k consecutive rounds with blocked Send Phases. The adversary spends $T = \Omega(k\ell)$ while each party spends $O(k\sqrt{\ell}) = O(T/\sqrt{\ell})$ in expectation. Therefore, for $T > \ell$, we no longer have the quadratic advantage obtained above for a single blocked Send Phase.

This problem arises because each blocked Send Phase imposes equal asymptotic cost on the adversary. In contrast, imagine if the cost of the final blocked Send Phase equaled the total cost of all previous $O(k)$ blocked Send Phases. Then, the quadratic advantage would still hold by the same reasoning as above; simply apply the same analysis to the final blocked phase. We can achieve this property by increasing the length of the Send Phase as an exponential function of the number of rounds completed so far.

Figure 2 provides the pseudocode with this modification implemented; we name this intermediate algorithm ITERATION-2. The length of the Send Phase is 2^{ci} where the constant $c > 0$ is left unfixed (to be as general as possible) and will be determined later.

The probabilities of Alice sending and Bob listening are also modified such that the expected cost to Alice is $O(2^{ai})$ and to Bob is $O(2^{bi})$, where $a > 0$ and

$b > 0$ are constants that we leave unspecified for now. However, given that a birthday-paradox-like argument is used to show that Alice and Bob succeed in communication, then we require $a + b = c$ to make our analysis work.[3]

To illustrate what is meant by such an argument, we *sketch* some analysis for $c = 2$ and $a = b = 1$. For simplicity, assume a weaker adversary who jams each slot with probability p_j. Later, in Section 3, we provide a rigorous analysis that holds against our more powerful adversary.

First, consider the case of "light jamming", say $p_j \leq 1/2$. Since $a = b = 1$, Alice sends and Bob listens with probability $\Theta(1/2^i)$ per slot. The probability that Bob receives m in slot j is:

$$\begin{aligned} &= Pr(\text{slot } j \text{ not jammed}) \times Pr(\text{Alice sends in slot } j) \\ &\quad \times Pr(\text{Bob listens in slot } j) \\ &= (1 - p_j)\left(\frac{2}{2^i}\right)\left(\frac{2}{2^i}\right) \end{aligned}$$

The probability that Bob fails to receive m over all $2^{ci} = 2^{2i}$ slots in the Send Phase is at most:

$$\left(1 - \frac{4(1-p_j)}{2^{2i}}\right)^{2^{2i}} \leq e^{-2}$$

Therefore, when (roughly) at most half the slots are jammed, there is a constant probability of success, but this argument requires $a + b = c$.

Conversely, consider $p_j > 1/2$; for example, perhaps $p_j = 1$ and all slots are jammed in the Send Phase. Clearly, Alice will fail in sending m to Bob. However, we do "succeed" in making Carol incur significantly more cost. The expected cost to Carol is $T = \Omega(2^{ci})$ while Alice and Bob each spend $O(2^i) = O(\sqrt{T})$ in expectation. Therefore, we recover the quadratic result above for a blocked Send Phase.

Finally, we highlight the fact that $c > 2$ does not improve this advantage. For example, if we set $c = 4$, then we require $a + b = 4$ and the maximum expected cost to either party is minimized by setting $a = b = 2$ and this still yields an expected cost to Alice and Bob of $O(\sqrt{T})$.

2.3 Robustness to Attacks in the Nack Phase

The second shortcoming of ITERATION-1 (and also of ITERATION-2) involves the Nack Phase under Case 2. Recall that Bob can be spoofed and his communications can be jammed. Assume that Bob receives m and terminates. Then, in the very

[3] More generally, we need $a + b \geq c$, but we seek to minimize cost and so we opt for equality.

```
ITERATION-3 for round $i \geq 1$
Send Phase: For each of the $2^{ci}$ slots do
    • Alice sends $m$ with probability $2/2^{(c-a)i}$
    • Bob listens with probability $2/2^{(c-b)i}$
If Bob received $m$, then he terminates
Nack Phase: For each of the $\ell$ slots do
    • Bob sends a nack message
    • Alice listens with probability $4/\ell$
If Alice listened to a clear slot, then she terminates.
```

Figure 3: Pseudocode for ITERATION-3.

next Nack Phase, the adversary may spoof a `nack` message and force Alice to execute another round. In each subsequent Send Phase, the adversary will do nothing (in order to avoid any cost) and then send another `nack` in the next Nack Phase. Therefore, in each round i, Alice incurs a cost of roughly 2^{ai} in expectation while the adversary has a cost of 1. Through this attack, the adversary can force Alice to quickly deplete her energy supply.

Note that even if messages from Bob could be authenticated, the adversary may simply generate noise which will yield the same result. This is unavoidable since collision detection is used as a reliable negative acknowledgement.

This problem arises because it is "too cheap" for Bob to force Alice to proceed into the next round. Intuitively, a remedy is to increase the cost required for Bob to prevent Alice's termination. As with the Send Phase of ITERATION-1, this could be accomplished by setting the length of the Nack Phase to be, say, ℓ and requiring Bob to send a `nack` in each slot. Therefore, Bob is making a "down payment" and this should be large enough to deter the adversary from spoofing Bob in the Nack Phase.

Should Alice verify the down payment by listening to all $\ell \geq 4$ slots? She is already incurring an expected cost of 2^{ai} from the preceding Send Phase, so she could afford to spend the same in the Nack Phase without changing her asymptotic cost. But what happens if the algorithm runs sufficiently long such that 2^{ai} exceeds ℓ? Instead, as we show in Section 3, Alice can sample each slot with probability $4/\ell$ in the Nack Phase; this only contributes additional expected cost of 4. The adversary may attempt to spend less than ℓ whilst spoofing Bob, but Alice's random

```
ITERATION-4 for round $i \geq 2/b$
Send Phase: For each of the $2^{ci}$ slots do
    • Alice sends $m$ with probability $2/2^{(c-a)i}$
    • Bob listens with probability $2/2^{(c-b)i}$
If Bob received $m$, then he terminates
Nack Phase: For each of the $2^{bi}$ slots do
    • Bob sends a nack message
    • Alice listens with probability $4/2^{bi}$
If Alice listened to a clear slot, then she terminates.
```

Figure 4: Pseudocode for ITERATION-4.

strategy is adequate to thwart such behavior. The pseudocode for this modification is given in Figure 3.

Finally, we note that fixing the length of the Nack Phase to be ℓ in each round runs into a problem similar to that faced in the Send Phase of ITERATION-1: multiple consecutive blocked Nack Phases – where more than half the slots are jammed – will degrade the resource-competitive ratio. Therefore, we adopt the same solution by increasing the length of the Nack Phase as a function of the number of rounds completed so far. Since Bob spends an expected 2^{bi} in the Send Phase, the length of the Nack Phase can also be 2^{bi} without increasing his asymptotic cost.

The pseudocode for these modifications is presented as ITERATION-4 in Figure 4.

2.4 Handling Variance of Cost

Consider the following strategy by the adversary. All slots in each consecutive Send Phase are jammed until a round r is encountered where Alice sends 2^{cr} in the corresponding Send Phase. Note that Alice sends probabilistically, so there is a small, but non-zero, probability of this event occurring in each round, and the adversary's strategy means it will eventually occur with probability 1. At that point, $T = \Theta(2^{cr})$ and Alice's cost is $\Theta(T)$ which is clearly poor.

Why not modify the algorithm such that Alice randomly selects exactly 2^{ai+1} slots in which to send prior to executing the next Send Phase? This prevents any variance in cost and, therefore, prevents the above attack.

To see why this is a bad idea, note an advantage of having Alice decide randomly whether to send on a per-slot basis: *an adaptive adversary has no more power than a non-adaptive one*. This is because the action Alice takes in the current slot is independent of what she has done in the previous slots. In other words, knowledge of the players' past actions does not help Carol plan her jamming in the next slot. Therefore, this design choice provides a useful property for tolerating an adaptive adversary.

Selecting a fixed number of slots to send in *a priori* deprives us of this property. For example, if Alice does not send m in slot 1 of the Send Phase, an adaptive adversary learns of this prior to its decision on whether to jam in slot 2. The absence of sending in slot 1 implies that the probability that Alice sends in slot 2 has increased given that she must send in exactly 2^{ai+1} slots; that is, these events are dependent. For the same reason, the probability of a slot in which Alice sends *and* Bob listens changes as we proceed through the Send Phase. The adversary may use this information to improve its jamming and, at the very least, analyzing such an algorithm seems difficult.

Instead, we make the following modification. In the Send Phase, if Alice has sent in 2^{ai+2} slots, she remains silent for the remainder of the phase. Similarly, if Bob has listened in 2^{bi+2} slots, he performs no more listening for the remainder of the phase. This ***cutoff*** for either party bounds the cost to be at most a constant factor more than the expectation. In other words, the cost to Alice and Bob is $O(2^{ai})$ and $O(2^{bi})$, respectively (the additional constant factors are useful in applying a Chernoff bound later on in the analysis). The same modification is made to the Nack Phase: if Alice has listened in 2^{ai+2} slots, she performs no more listening for the remainder of the phase. The pseudocode for this final design decision is presented in Figure 5 as ROBUSTTALK.

This design maintains independence between events in each slot up until the cutoff point. Also, in each slot up until this cutoff point, the probability of a slot where Alice sends and Bob listens is always the same at $4/2^{(c-a)i+(c-b)i}$. Of course, once the cutoff point is reached, the adversary learns this and does not need to perform any more jamming; however, up to this point, we preserve the robustness to an adaptive adversary. In analyzing the impact of either party reaching the cutoff in a round, we will pessimistically assume failure and incorporate the cost of this round for each party. However, in Section 3, we will show that this a low-probability event and does not impact the expected cost.

2.5 Description of ROBUSTTALK

We summarize a round i of ROBUSTTALK:

- *Send Phase:* This phase consists of 2^{ci} slots. If Alice has sent in less than 2^{ai+2}

```
RobustTalk for round $i \geq 2/b$
Send Phase: For each of the $2^{ci}$ slots do
    • If Alice has sent in less than $2^{ai+2}$ slots in this phase, she sends $m$ with probability $2/2^{(c-a)i}$
    • If Bob has listened in less than $2^{bi+2}$ slots in this phase, he listens with probability $2/2^{(c-b)i}$
If Bob received $m$, then he terminates
Nack Phase: For each of the $2^{bi}$ slots do
    • Bob sends a nack message
    • If Alice has listened in less than $2^{ai+2}$ slots in this phase, she listens with probability $4/2^{bi}$
If Alice listened to a clear slot, then she terminates.
```

Figure 5: Pseudocode for RobustTalk.

slots, then she will send m in the current slot with probability $\frac{2}{2^{(c-a)i}}$. This yields an expected cost of $2^{(c-a)i+1}$ over the entire phase. If Bob has listened in less than 2^{bi+2} slots, he will listen in the current slot with probability $\frac{2}{2^{(c-b)i}}$. This yields an expected total cost of 2^{bi+1} over the entire phase.

• *Nack Phase:* This phase consists of 2^{bi} slots. If Bob has not received m, then he sends a `nack` in all 2^{bi} slots. If Alice has listened in less than 2^{ai+2} slots, then she will listen in the current with probability $4/2^{bi}$ (note that $i \geq 2/b$ is required) for an expected total cost of 4 over the phase.

Termination Conditions: As discussed in Section 1.5, termination conditions are important and we highlight these here. Bob terminates the protocol upon receiving m. Since Alice cannot be spoofed, as discussed in Section 1.4, this termination condition suffices.

Alice terminates if she listens to a clear slot (neither noisy nor containing a `nack` message) in the Nack Phase; since jammed slots are detectable by Alice while listening (Section 1.4), this condition suffices. That is, Alice continues into the next round if and only if (i) Alice listens to zero slots or (ii) all slots listened to by Alice in the Nack Phase contain a blocked slot or `nack` message. There are two situations where this occurs:

• *Send Failure:* Bob has not received m.

• *Nack Failure:* Bob has terminated and Carol either spoofs `nack` messages or jams slots in order to trick Alice into thinking a valid `nack` was sent but jammed.

Nack Failures and Cases 1 & 2: Note that an "acknowledgement" is indicated by having at least one clear slot in the Nack Phase. A ***Nack Failure*** refers to the situation in which, by the end of the Nack Phase, Alice believes Bob is alive and has failed to receive m.

In Case 2, as discussed in Section 2.3, a Nack Failure may occur via an attack in the Nack Phase after Bob has received m and terminated. Carol may spoof Bob in the Nack Phase by making it appear as if Bob did not receive m and is requesting that both he and Alice proceed into the next round. This behavior keeps Alice active and incurring a cost. Critically, this attack affects Alice only; if Bob has not terminated, a `nack` message will be issued anyway and the attack accomplishes nothing. Therefore, we need only consider this attack in the situation where Bob has terminated.

In Case 1, no jamming occurs in the Nack Phase and, therefore, no Nack Failure can occur. We note that, in Case 1, the Nack Phase can be shortened to a single slot – the algorithm ITERATION-2 will suffice – where Bob sends his `nack` message and Alice listens; however, this does not change our *asymptotic* cost for Case 1. Since our current presentation applies to both cases, we proceed with analyzing ROBUSTTALK.

3 Analysis of ROBUSTTALK

Throughout, assume ceilings on the number of slots indicated for use by Alice or Bob if it is not an integer. For any given round, we say it is a ***blocked Send Phase*** if Carol jams at least half of the slots in the Send Phase; otherwise, it is a ***non-blocked Send Phase***. Similarly, a ***blocked Nack Phase*** occurs if Carol jams or spoofs `nack` messages from Bob in at least half the slots in the Nack Phase; otherwise, it is a ***non-blocked Nack Phase***.

Bounds on constants a, b, c: We make a few important remarks about these constants. Note that if $a \geq b$, then the expected cost to Alice is at least as much as the expected cost to Bob. But recall that, in Case 2, the adversary can spoof Bob. Therefore, this allows Carol to cause a Nack Failure with at most the same cost as what Alice incurs in the preceding Send Phase (note that Carol will avoid listening in the Send Phase and incur zero cost there). As discussed in Section 2.3, we must avoid this since it admits an energy-draining attack against Alice. Therefore, in Case 2, we require $a < b$.

Additionally, our analysis will depend on a birthday-paradox-like argument (in Lemmas 2 and 4) to show that m or a `nack` message is successfully transmitted across the channel. As sketched in Section 2.2, we will need that $a + b = c$ for the analysis to work.

Finally, as discussed in Section 2.2, we assume that $c \leq 2$. Consequently, $a < 2$ and $b < 2$.

Our analysis is composed of two pieces. First, we consider "light jamming" where we have no blocked phases. We show that Alice succeeds in communicating m to Bob with probability at least $1 - e^{-2}$ in each non-blocked Send Phase. Given that Bob has terminated, a similar result is shown for the probability that Alice (correctly) terminates in a non-blocked Nack Phase.

Second, we consider "heavy jamming" where we address blocked phases. Given the exponentially increasing length of the phases, intuitively the expected cost is dominated by the cost to the parties in the last blocked round. We can then use this cost to compare against Carol's cost T.

In both situations, the issue of variance described in Section 2.4 adds a subtlety to our calculations. For light jamming, we only show constant probability of success in non-blocked phases so long as neither Alice nor Bob reach their respective cutoffs.

For heavy jamming, note that the adversary decides when the last blocked round occurs. For example, what if the adversary jams all slots until the (low-probability) event that Alice reaches her cutoff point in the Send Phase? This highlights the problem of conditioning on the adversary's final blocked round in order to calculate the expected cost.

Our modifications in Section 2.4 permit a clean analysis. We bound the number of rounds in which either party reaches its respective cutoff. We can show that this occurs with probability exponentially small in the round index i, and therefore the impact on the expected cost to each party is negligible. We make use of the following Chernoff bound:

Theorem 3. *([59]) Let* $X_1, \ldots, X_n$ *be binary random variables such that* $\Pr(X_i) = p$ *and let* $X = \sum_{i=1}^{n} X_i$*. For any* δ*, where* $0 < \delta < 1$:

$$\Pr(X > (1 + \delta)\, E[X]) \leq e^{-\delta^2\, E[X]/3}$$

Define a ***failed round*** to be a round in which either Alice or Bob reaches a cutoff in either the Send or Nack Phase; otherwise, it is ***non-failed round***. We now bound the probability of a failed round as a function of the round index i:

Lemma 1. *For any round* $i \geq 1$ *the probability of a failed round is at most* $\exp(-2^{ai}/3) + \exp(-2^{bi}/3)$.

Proof. In the Send Phase of round i, let the binary random variable $X_j = 1$ if Alice sends m in slot j; otherwise, $X_j = 0$. Letting $X = \sum_{j=1}^{2^{ci}} X_j$, then:

$$E[X] = \sum_{j=1}^{2^{ci}} E[X_j] = \sum_{j=1}^{2^{ci}} (2/2^{(c-a)i}) = 2^{ai+1}$$

by linearity of expectation. A similar calculation for the Nack Phase yields an expected cost of 4. Therefore, over epoch i, Alice's expected cost is $2^{ai+1} + 4$ for $i \geq 1$. By Theorem 3, using $\delta = 1$ yields that the probability of exceeding this expected cost by a factor of at least 2 is less than $e^{-2^{ai}/3}$.

In the Send Phase of round i, let the binary random variable $Y_j =$ if Bob listens in slot j; otherwise, $Y_j = 0$. Letting $Y = \sum_{j=1}^{2^{ci}} Y_j$, then:

$$E[Y] = \sum_{j=1}^{2^{ci}} E[Y_j] = \sum_{j=1}^{2^{ci}} (2/2^{(c-b)i}) = 2^{bi+1}$$

by linearity of expectation. There is an additional 2^{bi} cost associated with the Nack Phase. By Theorem 3, the probability of exceeding the expected cost by a factor of at least 2 is less than $e^{-2^{bi}/3}$. □ □

By Lemma 1, the probability of a failed round is very small as a function of the round index i. We will show that the contribution of failed rounds does not impact our asymptotic analysis.

We now present the light jamming portion of our analysis.

Lemma 2. *Consider a blocked Send Phase in a non-failed round. The probability that Bob does not receive the message from Alice is at most* e^{-2}.

Proof. Let $s = 2^{ci}$ be the number of slots in the Send Phase. Let p_A be the probability that Alice sends in a particular slot. Let p_B be the probability that Bob listens in a particular slot. Let $X_j = 1$ if the message is not delivered from Alice to Bob in the j^{th} slot. Then:

$$\begin{aligned} & Pr[m \text{ is not delivered in the Send Phase}] \\ = \; & Pr[X_1 X_2 \cdots X_s = 1] \\ = \; & Pr[X_s = 1 \mid X_1\, X_2 \cdots X_{s-1} = 1] \cdot \prod_{i=1}^{s-1} Pr[X_i = 1] \end{aligned}$$

Let $q_j = 1$ if Carol does not jam in slot j; otherwise, let $q_j = 0$. The value of q_j can be selected arbitrarily by Carol. Then:

$$Pr[X_i = 1 \mid X_1 X_2 \cdots X_{i-1} = 1] \quad = \quad 1 - p_A p_B q_j$$

Substituting for each conditional probability, we have:

$$\begin{aligned} Pr[X_1 X_2 \cdots X_s = 1] &= (1 - p_A p_B q_1) \cdots (1 - p_A p_B q_s) \\ &= \prod_{j=1}^{s} (1 - p_A p_B q_j) \\ &\leq e^{-p_A p_B \sum_{j=1}^{s} q_j} \\ &\leq e^{-2} \end{aligned}$$

where the last inequality follows from:

$$\begin{aligned} p_A p_B \sum_{j=1}^{s} q_j &\geq (2/2^{(c-a)i})(2/2^{(c-b)i})(s/2) \\ &= (2/2^{(c-a)i})(2/2^{(c-b)i})(2^{ci}/2) \\ &= 2 \end{aligned}$$

since $a + b = c$ and there is no blocked send phase which means Carol jams at most $s/2$ slots. □ □

Note that Lemma 2 handles adaptive (but not reactive) adversaries. A simple but critical feature of tolerating adaptive adversaries is that the probability that a party is active in one slot is independent from the probability that the party is active in another slot. Therefore, knowing that a party was active for k slots in the past conveys no information about future activity. For reactive adversaries, we need only modify Lemma 2 and we address this later (see Section 4).

Lemma 3. *Assume there are no blocked Send Phases and no blocked Nack Phases. The expected cost of each party is $O(1)$.*

Proof. We compute the expected cost over both non-failed and failed rounds for $i \geq 1$; note that this can only overestimate the expected cost since ROBUSTTALK specifies $i \geq 2/b$ and $b < 2$.

Using Lemmas 1 and 2, the expected cost to Alice is at most:

$$\begin{aligned}
& \sum_{i=1}^{\infty} e^{-2(i-1)}\, O(2^{ai}) + \sum_{i=1}^{\infty} \exp\left(-2^{\Theta(i-1)}\right) O(2^{ai}) \\
= \;& O(1) \sum_{i=1}^{\infty} \left(\frac{2^a}{e^2}\right)^i \\
= \;& O(1)
\end{aligned}$$

where the last line follows from the fact that $a < 2$ and the sum of a geometric series. Similarly, the expected cost to Bob is at most:

$$\begin{aligned}
& \sum_{i=1}^{\infty} e^{-2(i-1)} O(2^{bi}) + \sum_{i=1}^{\infty} \exp\left(-2^{\Theta(i-1)}\right) O(2^{bi}) \\
\leq \;& O(1) \sum_{i=1}^{\infty} \left(\frac{2^b}{e^2}\right)^i \\
= \;& O(1)
\end{aligned}$$

where the last line follows from the fact that $b < 2$ and the sum of a geometric series. □ □

Lemma 4. *Assume that Bob has received m by non-failed round i, and that round i has a non-blocked Nack Phase. Then, the probability of a Nack Failure is at most e^{-2}.*

Proof. Let $s = 2^{bi}$ denote the number of slots in the Nack Phase and let $p = 4/2^{bi}$ denote the probability that Alice listens in a slot. For slot j, define X_j such that $X_j = 1$ if Alice does not terminate. Then, $Pr[$ Alice retransmits m in round $i + 1] = Pr[X_1 X_2 \cdots X_s = 1]$. Let $q_j = 1$ if Carol does not jam in slot j; otherwise, let $q_j = 0$. The q_j values are determined arbitrarily by Carol. Since Alice terminates when she listens and hears a clear slot, then $Pr[X_j = 1] = (1 - pq_j)$. Therefore, $Pr[X_1 X_2 \cdots X_s = 1] \leq e^{-p \sum_{j=1}^{s} q_j} \leq e^{-2}$. □ □

We now consider the heavy jamming portion of our analysis.

Lemma 5. *Assume there is at least one blocked Send Phase. In Case 1, the expected cost to Alice and Bob is $O(T^{a/c})$ and $O(T^{b/c})$, respectively. In Case 2, the expected cost to Alice and Bob is $O(T^{a/c} + T^{a/b})$ and $O(T^{b/c})$, respectively.*

Proof. Let $i \geq 2/b$ be the last blocked Send Phase. Let $j \geq i$ be the last blocked Nack Phase; if no such blocked Nack Phase exists, then assume $j = 0$.

Carol may choose i and j depending on the history of the execution. With this notation in place, we can bound the cost to Carol, Alice, and Bob.

Carol: In Case 1, only blocked Send Phases occur and so $T = \Omega(2^{ci})$. In Case 2, since there can be a blocked Nack Phase, the total cost to Carol is $T = \Omega(2^{ci}+2^{bj})$.

Alice: We begin by calculating the expected cost to Alice prior to successfully transmitting m. Using Lemmas 1 and 2, the expected cost to Alice prior to m being delivered is at most:

$$\begin{aligned} & O(2^{ai}) + \sum_{k=1}^{\infty} e^{-2(k-1)}\, O(2^{a(i+k)}) \\ & \qquad + \sum_{k=1}^{\infty} \exp\left(-2^{\Theta(k-1)}\right) O(2^{a(i+k)}) \\ = & \ O(2^{ai}) + O(2^{ai}) \sum_{k=1}^{\infty} \left(\frac{2^a}{e^2}\right)^k + O(2^{ai}) \\ = & \ O(2^{ai}) \end{aligned}$$

where the last line follows $a < 2$ and the sum of a geometric series.

Next, using Lemmas 1 and 4, we calculate the expected cost to Alice after delivery of m; this addresses blocked Nack Phases possible only in Case 2. By assumption, the last blocked Nack Phase occurs in round j and therefore Alice's expected cost is at most:

$$\begin{aligned} & O(2^{aj}) + \sum_{k=1}^{\infty} e^{-2(k-1)} O(2^{a(j+k)}) \\ & \qquad + \sum_{k=1}^{\infty} \exp\left(-2^{\Theta(k-1)}\right) O(2^{a(j+k)}) \\ = & \ O(2^{aj}) + O(2^{aj}) \sum_{k=1}^{\infty} \left(\frac{2^a}{e^2}\right)^k + O(2^{aj}) \\ = & \ O(2^{aj}) \end{aligned}$$

where the last line follows $a < 2$ and the sum of a geometric series.

Therefore, in Case 2, the total expected cost to Alice is $O(2^{ai} + 2^{aj})$. Since $T = \Omega(2^{ci} + 2^{bj})$, this cost as a function of T is $O(T^{a/c} + T^{a/b})$. For Case 1, there are no blocked Nack Phases and so Alice's cost is simply $O(T^{a/c})$.

Bob: We analyze the expected cost to Bob up until m is received. Note that we assume Carol does not have any cost for blocked Nack Phases up until this point since, as discussed in Section 2.5, there is no benefit to causing a Nack Failure via jamming.

Using Lemma 2, Bob's expected cost prior to receiving m is at most:

$$\begin{aligned} & O(2^{bi}) + \sum_{k=1}^{\infty} e^{-2(k-1)} O(2^{b(i+k)}) \\ & \quad + \sum_{k=1}^{\infty} \exp\left(-2^{\Theta(k-1)}\right) O(2^{b(i+k)}) \\ \leq \;& O(2^{bi}) + O(2^{bi}) \sum_{k=1}^{\infty} \left(\frac{2^b}{e^2}\right)^k + O(2^{bi}) \\ = \;& O(2^{bi}) \end{aligned}$$

where the last line follows $b < 2$ and the sum of a geometric series. Therefore, the expected cost for Bob as a function of $T = \Omega(2^{ci})$ is $O(T^{b/c})$. □ □

We now prove Theorem 1 stated in Section 1.6:
Proof of Theorem 1: For both Case 1 and 2, when there are no blocked Send Phases and no blocked Nack Phases, the expected cost to each party is $O(1)$ by Lemma 3. Therefore, we have $\tau = O(1)$.

For both Case 1 and Case 2, if there is at least one blocked Send Phase, Lemma 5 gives the expected cost for Alice and Bob as $O(T^{a/c})$ and $O(T^{b/c})$, respectively. Recall from Section 1.5 that we desire the worst-case relative costs for Alice and Bob be equal. Therefore, setting $a/c = b/c$ implies that $a = b = 1$ and $c = 2$. This yields $\rho(T) = O(T^{0.5})$ and therefore the expected cost to each party is $O(T^{0.5} + 1)$.

In Case 2, if there is at least one blocked Send Phase or Nack Phase, the expected cost to Alice and Bob is $O(T^{a/c} + T^{a/b})$ and $O(T^{b/c})$, respectively, by Lemma 5. The exponents of interest are now a/c, a/b, and b/c. Notice that a/c and a/b both correspond to Alice's expected cost, but that $a/c < a/b$. Therefore, we need consider only a/b and b/c and we wish $a/b = b/c$ to obtain fairness. Also recall that we insisted on $a + b = c$.

We now have three parameters and two constraints. However, this is sufficient for us to derive useful values for a, b, and c. By dividing the second constraint by the first, we have:

$$\begin{aligned} & \frac{(a+b)b}{a} = \frac{c^2}{b} \\ \Rightarrow\quad & \left(1+\frac{b}{a}\right) = \left(\frac{c}{b}\right)^2 \\ \Rightarrow\quad & \left(1+\frac{c}{b}\right) = \left(\frac{c}{b}\right)^2 \quad \text{since } \frac{a}{b} = \frac{b}{c} \\ \Rightarrow\quad & \left(\frac{c}{b}\right)^2 - \left(\frac{c}{b}\right) - 1 = 0 \end{aligned}$$

Solving for the roots of this quadratic polynomial yields $c/b = (1+\sqrt{5})/2 = \varphi$ which is the golden ratio. Therefore, so long as the constants a, b, c are set such that $c/b = b/a = \varphi$, RobustTalk has $\rho(T) = O(T^{\Phi}) = O(T^{\varphi-1}) = O(T^{0.62})$ where $\Phi = 1/\varphi = \varphi - 1$.

How many slots in expectation occur prior to termination by both Alice and Bob? We focus on Alice since she terminates after Bob. Assume pessimistically that a blocked Send or Nack Phase prevents her from terminating. Carol can stretch her budget the farthest by blocking the Nack Phase only as this incurs a cost of $2^{bi}/2$ rather than a cost of $2^{ci}/2$ for blocking the Send Phase.

For how many rounds can Carol do this? Solving for s in $\sum_{i=2/b}^{s} 2^{bi}/2 \geq T$ implies that $s \leq \lg(T)/b + O(1)$. To this, we add the number of non-blocked phases Alice endures before terminating; let X denote the random variable for this number of rounds. Then, $E[X] \leq (1-e^{-2}) + 2\,e^{-2}(1-e^{-2}) + 3\,e^{-4}(1-e^{-2}) + ... = \sum_{i=1}^{\infty} i\,e^{2(1-i)}(1-e^{-2}) = (1-e^{-2})e^2 \sum_{i=1}^{\infty} i(e^{-2})^i$ by Lemmas 2 and 4. Therefore, $E[X] \leq 1/(1-e^{-2}) = O(1)$ which means Carol can delay the termination by an additional $O(1)$ rounds.

For each of the $\lg(T)/b + O(1)$ rounds, Alice experiences a delay of $2^{ci} + 2^{bi} = O(2^{ci})$ slots. This means that Alice terminates within an expected $O(2^{(c/b)\lg(T)+O(c)}) = O(T^{c/b})$ slots. This translates to $O(T^2)$ and $O(T^{\varphi})$ for Cases 1 and 2, respectively. □

Figure 6 provides an updated RobustTalk with a setting of the constants $a = \varphi - 1, b = 1$, and $c = \varphi$ such that we obtain the guarantees of Theorem 1 for Case 2.

4 Tolerating a Reactive Adversary

Consider a reactive adversary Carol who can detect channel activity for free — that is, distinguish between when the channel is clear and when it is busy — and then jam. This ability to reactively jam is possible in WSNs [79].

```
RobustTalk for round $i \geq 2$
Send Phase: For each of the $2^{\varphi i}$ slots do
    • If Alice has sent in less than $2^{(\varphi-1)i+2}$ slots in this phase, she sends $m$ with probability $2/2^i$
    • If Bob has listened in less than $2^{i+2}$ slots in this phase, he listens with probability $2/2^{(\varphi-1)i}$
If Bob received $m$, then he terminates
Nack Phase: For each of the $2^i$ slots do
    • Bob sends a nack message
    • If Alice has listened in less than $2^{\varphi i+2}$ slots in this phase, she listens with probability $4/2^i$
If Alice listened to a clear slot, then she terminates.
```

Figure 6: Pseudocode for RobustTalk with the constants $a = \varphi - 1$, $b = 1$, $c = \varphi$.

Carol can now detect that m is being sent in the Send Phase and jam it without fail. To address this powerful adversary, we consider the case where non-critical data, m', is sent over the channel by other participants in addition to Alice and Bob. Carol can detect the traffic; however, she cannot discern whether it is m or m' without listening to a portion of the communication. For example, in practice, she may need to listen to a part of the packet header in order to make such a determination.

In a slot where channel activity is detected, if Carol listens for a portion of the message, we assume she incurs a cost. Therefore, the cost to Carol is proportional to (1) the number of messages to which she listens, and (2) the number of slots in which she jams. Importantly, in the presence of m', Carol's ability to detect is unhelpful since m' provides "camouflage" for m.

As an example, assume that *all* slots in the Send Phase are used either by Alice to send m (as per our algorithm) or another party, Dave, whose transmissions of m' do not interest Carol. Assume that Dave's transmissions are in the majority.

In this situation, detecting channel activity does not help Carol decide on whether to jam; all slots are used. Regardless of how she decides to act, Carol can do no better than picking slots independent of whether she detects channel activity. In other words, channel activity is no longer useful in informing Carol's decisions about whether to jam.

But assuming *all* slots are active is problematic. How is this guaranteed or coordinated? We relax our assumption that all slots are used. Instead, we assume that other network traffic occurs such that Carol will always detect traffic on at least a *small* constant fraction of slots in the Send Phase. This traffic is assumed to be random and independent of Alice sending m.

Upon detecting traffic, Carol may listen to a portion of the message to discover if it is m or m' and then decide on whether to jam, all at a cost of 1. However, this is roughly as expensive — it is the same order of magnitude — as simply jamming outright (also a cost of 1 in our model).

In practice, such situations can arise where communication occurs between many participants, or via several distributed applications, or internally between respective co-located networks. An attacker may wish to selectively target only one component of the system in order to conserve energy and reduce the chances of its malicious activity being detected.

As with any two messages, if m and m' are sent over the channel in the same slot, the two messages collide and Bob receives neither. Define a slot as ***active*** if either m or m' is sent in that slot. For this result only, redefine a ***blocked Send Phase*** as one where Carol listens to or jams more than a $1/3$-fraction of the *active* slots in the Send Phase; otherwise, it is ***non-blocked***. We provide a result analogous to Lemma 2.

Lemma 6. *Let Carol be an adaptive and reactive adversary. Then, in a non-blocked Send Phase, the probability that Bob does not receive* m *from Alice is at most* e^{-2}.

Proof. Let $x = 2^{\varphi i}$ be the number of slots in the Send Phase. Consider the set of slots used by all participants (such as Dave) other than Alice. We assume these participants pick their slots at random to send, so that for any slot the probability is 2/3 that the slot is chosen by at least one of them. Let p_A and p_B be the probabilities of sending and listening by Alice and Bob in the Send Phase, respectively.

Since we assume these messages m' are sent independently at random, then Chernoff bounds imply that, with high probability (i.e., $1-1/x^{c'}$ for constants c', ϵ and sufficiently large x) the number of slots y during which m' is sent is greater than $(2x/3)(1-\epsilon)$ where x is the total number of slots in a phase.

In the same way, assume the number of slots in which Alice sends is at least $a' = (1-\delta)xp_A = (1-\delta)2^{(\varphi-1)i+1}$ with probability $1 - 1/x^{c''}$ for a constant δ, c'' and sufficiently large x. The number of active slots is clearly at least y.

By definition of a non-blocked Send Phase, Carol listens to or blocks at most $x/3$ active slots in the Send Phase. As Carol has no information about the source of a message sent in an active slot until she listens to it, her choice is independent of the source of the message.

Given a slot in which Alice sends, there is at least a $1 - (x/3)/y$ chance it will not be listened to or jammed by Carol. The probability that this slot will not be used by another participant is $1/3$ and the probability that Bob will listen to the slot is p_B. Hence the probability of a successful transmission from Alice to Bob in a slot for which Alice sends is at least:

$$\begin{aligned}\left(1 - \frac{x}{3y}\right)\left(\frac{1}{3}\right)p_B &= \left(1 - \frac{1}{2(1-\epsilon)}\right)\left(\frac{1}{3}\right)p_B \\ &\geq \left(\frac{1}{3} - \frac{(1+\delta)}{6}\right)p_B \\ &\geq \left(\frac{1}{12}\right)p_B\end{aligned}$$

for sufficiently large x (that reduces the size of δ) when $y > (1-\epsilon)(2x/3)$.

The probability that all messages that Alice sends fail to be delivered is at most $(1 - p_B/12)^{a'} + 2/x^{c''}$ where the last term is the probability of the bad event that y or a' is small and $c'' > 0$ is a constant. Redefine $p_B = 24/((1-\delta)2^{(\varphi-1)i})$ where the value 24 is set to off-set the additive $2/x^{c''}$ term. Note that this constant factor increase in the listening probability does not change our asymptotic results and our analysis in Section 3 proceeds almost identically. Therefore, we then have $(1 - p_B/12)^{a'} + 2/x^{c''} \leq e^{-2}$. □ □

The ROBUSTTALK can be modified so that the initial value of i is large enough to render the error arising from the use of Chernoff bounds sufficiently small; we omit these details.

The required level of channel traffic detected by Carol is flexible. Different values can be accommodated if the parties' probabilities for sending and listening are modified appropriately in ROBUSTTALK; our results hold asymptotically. We emphasize that further revision of the arguments presented in Section 3 are minor. Lemmas 3,4, and 5 do not require modification. Carol cannot decide to block only when Alice is listening since detecting when a node is listening is assumed to be impossible under our model. Alternately, Carol cannot silence a `nack` through (reactive) jamming since this is still interpreted as a retransmission request. Using Lemma 6, Theorem 1 follows as before with the same asymptotic guarantees.

Finally, we note that the conclusion of our argument aligns with claims put forth in empirical results on reactive jamming; that is, such behavior does not necessarily result in a more energy-efficient attack because the adversary must still be listening to the channel for broadcasts prior to committing itself to their disruption [82].

5 Resource-Competitive Local Broadcast

We present a resource-competitive algorithm, ROBUSTBROADCAST, that allows Alice to send a message m to n neighboring ***receivers*** within her transmission range; that is, our algorithm allows Alice to perform a local broadcast. We assume that the quantity $\ln n$ is known to Alice; however, our results likely still hold if a constant-factor approximation is known.

Throughout this section, we only consider the equivalent of Case 2 in this section; that is, where Carol may spoof any receiver(s) and can jam their transmissions. However, equivalent results trivially hold for the simpler Case 1. ROBUSTBROADCAST preserves much of the same guarantees as ROBUSTTALK, and is fair to all parties up to a polylogarithmic factor.

Our pseudocode is given in Figure 7. The probabilities for sending and listening are modified from ROBUSTTALK. Note that `nack` messages from multiple receivers can (and likely will) collide in the Nack Phase. This is fine since such a collision is due to either jamming or multiple receivers requesting a retransmission; in either case, Alice will correctly resend.

If Carol jams, then we assume she chooses the subset of the receivers (which can be the entire set of receivers) whose members do not receive m. In the literature, such an adversary is referred to as n-uniform [68]. The cost for jamming is still 1 per slot regardless of how many receivers are jammed. We maintain the definition in Section 2.2 of a blocked Send Phase (or Nack Phase): more than half the slots are jammed.

In this n-party setting, the arguments regarding the Send Phase are in need of minor revision, and we repeat these for completeness. We use nearly-identical definition of a failed round: any round in which either Alice or any receiver exceeds her/his respective expected cost by a factor of 2 or more in either the Send or Nack phase; otherwise, it is ***non-failed round***. In the multi-receiver case, we can bound the probability of a failed round as a function of n.

Lemma 7. *For any round $i \geq \lg(4\ln n)$ the probability of a failed round is at most $1/n$.*

Proof. In the Send Phase of round i, let the binary random variable $X_j = 1$ if Alice sends m in slot j; otherwise, $X_j = 0$. Letting $X = \sum_{j=1}^{2^{\varphi i}} X_j$, then:

$$E[X] = \sum_{j=1}^{2^{\varphi i}} E[X_j] = \sum_{j=1}^{2^{\varphi i}} (2\ln n/2^i) = 2^{(\varphi-1)i+1}\ln n$$

by linearity of expectation. A similar, calculation for the Nack Phase yields an expected cost of $4\ln n$. Therefore, over epoch i, Alice's expected cost is $2^{(\varphi-1)i} +$

$4 \ln n$. By Theorem 3, the probability that she reaches her cutoff is at most:

$$\exp\left(-2^{(\varphi-1)i+1} \ln n/3\right) \leq \exp\left(-\Theta(\ln^{\varphi} n)\right)$$

since $i \geq \lg(4 \ln n)$. In the Send Phase of round i, let the binary random variable $Y_j =$ if receiver R listens in slot j; otherwise, $Y_j = 0$. Letting $Y = \sum_{j=1}^{2^{\varphi i}} Y_j$, then:

$$E[Y] = \sum_{j=1}^{2^{\varphi i}} E[Y_j] = \sum_{j=1}^{2^{\varphi i}} (2/2^{(\varphi-1)i}) = 2^i$$

by linearity of expectation. There is an additional 2^i cost associated with the Nack Phase. By Theorem 3, the probability that R reaches its cutoff is at most:

$$\exp\left(-2^{i+1}/3\right) \leq \exp\left(-8/3\right) \ln n \leq n^{-8/3}$$

Taking a union bound implies that the probability that any receiver reaches its respective cutoff is at most $n^{-5/3}$. Finally, adding Alice's error from the above calculations yields the result. □ □

Next, we prove the analogue to Lemmas 2 and 3 in the multi-receiver case.

Lemma 8. *Consider a blocked Send Phase in a non-failed round. The probability that any receiver does not receive the message from Alice is at most* $1/n$.

Proof. Let $s = 2^{\varphi i}$ be the number of slots in the Send Phase. Let p_A be the probability that Alice sends in a particular slot. Let p_R be the probability that a fixed receiver listens in a particular slot. Let $X_j = 1$ if the message is not delivered from Alice to the receiver in the j^{th} slot. Then:

$$\begin{aligned} & Pr[m \text{ is not delivered in the Send Phase}] \\ = \; & Pr[X_1 X_2 \cdots X_s = 1] \\ = \; & Pr[X_s = 1 \mid X_1 \; X_2 \cdots X_{s-1} = 1] \cdot \prod_{i=1}^{s-1} Pr[X_i = 1] \end{aligned}$$

Let $q_j = 1$ if Carol does not jam in slot j; otherwise, let $q_j = 0$. The value of q_j can be selected arbitrarily by Carol. Then:

$$Pr[X_i = 1 \mid X_1 X_2 \cdots X_{i-1} = 1] \quad = \quad 1 - p_A p_R p_R q_j$$

Substituting for each conditional probability, we have:

$$\begin{aligned}
Pr[X_1X_2\cdots X_s = 1] &= (1-p_Ap_Rq_1)\cdots(1-p_Ap_Rq_s)\\
&= \prod_{j=1}^{s}(1-p_Ap_Rq_j)\\
&\leq e^{-p_Ap_R\sum_{j=1}^{s}q_j}\\
&\leq e^{-(4\ln n/2^{\varphi i})(s/2)}\\
&= e^{-2\ln n}\\
&= n^{-2}
\end{aligned}$$

Taking a union bound over all n receivers yields the result. □ □

Lemma 9. *Assume there are no blocked Send Phases and no blocked Nack Phases. With high probability, the cost to Alice and each receiver is $O(\ln^{\varphi} n)$ and $O(\ln n)$, respectively.*

Proof. For ease of exposition, let $\sigma = \lg(4\ln n)$. We compute the expected cost for both non-failed and failed rounds for $i \geq \sigma$. Using Lemmas 7 and 8, the expected cost to Alice is at most:

$$\sum_{i=\sigma}^{\infty}(1/n)^{(i-\sigma)}\,O(2^{(\varphi-1)i}\ln n) = O(\ln^{\varphi} n)$$

Similarly, the expected cost to each receiver R is at most:

$$\sum_{i=\sigma}^{\infty}(1/n)^{(i-\sigma)}O(2^{i}) = O(\ln n)$$

which completes the proof. Using a Chernoff bound (Theorem 5) provides the guarantee with high probability. □ □

Lemma 10. *Assume that all receivers have received m by non-failed round i and that round i is a non-blocked Nack Phase. Then the probability of a Nack Failure is less than $1/n^2$.*

Proof. Let $s = 2^i$ be the number of slots in the Nack Phase and let $p = 4\ln n/2^i$ be the probability that Alice listens in a slot. For slot j, define X_j such that $X_j = 1$ if Alice does not terminate. Then, $Pr[$ Alice retransmits m in round $i+1] = Pr[X_1X_2\cdots X_s = 1]$. Let $q_j = 1$ if Carol does not jam in slot j; otherwise, let

$q_j = 0$. The q_j values are determined arbitrarily by Carol. Since Alice terminates when she listens and hears a clear slot, then $Pr[X_j = 1] = (1 - pq_j)$. Therefore:

$$\begin{aligned} Pr[X_1 X_2 \cdots X_s = 1] &\leq e^{-p\sum_{j=1}^{s} q_j} \\ &\leq e^{-2\ln n} \\ &\leq n^{-2} \end{aligned}$$

which completes the proof. □ □

The analogue to Lemma 5 is mostly unchanged, although we first establish the following result:

Lemma 11. *With high probability, the cost of a round i is $O(2^{(\varphi-1)i}\ln n)$ and $O(2^i)$ for Alice and each receiver, respectively.*

Proof. For the Send Phase, define a binary random variable $X_j = 1$ if Alice sends in slot j; otherwise, $X_j = 0$. Setting $X = \sum_{j=1}^{2^{\varphi i}} X_j$, we have:

$$E[X] = \sum_{j=1}^{2^{\varphi i}} E[X_j] = \sum_{j=1}^{2^{\varphi i}} 2\ln n/2^i = O(2^{(\varphi-1)i}\ln n).$$

By Theorem 3, the cost is within a small constant factor with high probability. Similarly, in the Nack Phase, let the indicator random variable $Y_j = 1$ if Alice listens in slot j. Setting $Y = \sum_{j=1}^{2^{\varphi i}} Y_j$, we have:

$$E[Y] = \sum_{j=1}^{2^{i}} E[Y_j] = \sum_{j=1}^{2^{i}} 4\ln n/2^i = 4\ln n.$$

By Theorem 3, we can bound this to within a small constant factor with high probability. Therefore, with high probability, Alice's cost is $O(2^{(\varphi-1)i}\ln n) + O(\ln n) = O(2^{(\varphi-1)i}\ln n)$.

Consider a receiver R. Let $X'_j = 1$ if R listens in slot j of the Send Phase; otherwise, $X'_j = 0$. Setting $X' = \sum_{j=1}^{2^{\varphi i}} X'_j$, we have:

$$E[X'] = \sum_{j=1}^{2^{\varphi i}} E[X'_j] = \sum_{j=1}^{2^{\varphi i}} 2/2^{(\varphi-1)i} = O(2^i) = O(\ln n)$$

where the last equality follows from $i \geq \lg(4\ln n)$. By Theorem 3, we can bound this to within a small constant factor with high probability. Taking a union bound over all n receivers establishes the result. □ □

Lemma 12. *Assume there is at least one blocked Send Phase. With high probability, the cost to Alice and each receiver is $O(T^{\varphi-1})$.*

Proof. Let $i \geq \lg(4\ln n)$ be the last blocked Send Phase. Let $j \geq i$ be the last blocked Nack Phase; if no such blocked Nack Phase exists, then assume $j = 0$. Carol may choose i and j depending on the history of the execution. With this notation in place, we can bound the cost to Carol, Alice, and the receivers.

Carol: The total cost to Carol is $T = \Omega(2^{\varphi i} + 2^j)$.

Alice: Using Lemmas 7, 8, and 11, with high probability the cost to Alice prior to m being delivered is at most:

$$\begin{aligned} & O(2^{(\varphi-1)i}\ln n) + \sum_{k=1}^{\infty}(2/n)^{k-1}\, O(2^{(\varphi-1)(i+k)}\ln n) \\ = \; & O(2^{(\varphi-1)i}\ln n). \end{aligned}$$

Next, using Lemmas 7, 10, and 11, we calculate the cost to Alice after delivery of m; this addresses blocked Nack Phases. By assumption, the last blocked Nack Phase occurs in round j and, therefore, with high probability Alice's expected cost is at most:

$$\begin{aligned} & O(2^{(\varphi-1)j}\ln n) + \sum_{k=1}^{\infty}(2/n)^{k-1} O(2^{(\varphi-1)(j+k)}\ln n) \\ = \; & O(2^{(\varphi-1)j}\ln n). \end{aligned}$$

Therefore, with high probability, the cost to Alice is $O(T^{\varphi-1})$.

Receivers: We analyze the cost to each receiver up until m is received. Note that we assume Carol does not have any cost for blocked Nack Phases up until this point since, as discussed in Section 2.5, there is no benefit to causing a Nack Failure via jamming.

Using Lemmas 7, 8, and 11, prior to receiving m, with high probability each receiver has cost at most:

$$\begin{aligned} & O(2^i) + \sum_{k=1}^{\infty}(2/n)^{k-1} O(2^{i+k}) \\ = \; & O(2^i + \ln n) \\ = \; & O(2^i). \end{aligned}$$

Thus, with high probability, the cost to each receiver is $O(T^{1/\varphi}) = O(T^{\varphi-1})$.
□ □

Lemma 13. *With high probability, Alice and all correct receivers terminate* ROBUSTBROADCAST *in $O(T^{\varphi} + \ln^{\varphi} n)$ slots.*

Proof. We analyze the time it takes for Alice to terminate since she does so only after all receivers terminate. As with the analysis of ROBUSTTALK, we pessimistically assume a blocked phase prevents Alice from terminating in the current round. Furthermore, Carol will only perform her jamming to block Nack Phases since this maximizes the rounds that she keeps Alice alive.

Solving $\sum_{i=\lg(4\ln n)}^{s} 2^i/2 \geq T$ yields that the number of rounds that Carol can block is at most $s \leq \lg(T) + O(1)$. By the end of the very next round, all receivers are guaranteed with high probability to receive the message by Lemma 8, and Alice is guaranteed with high probability to terminate by Lemma 10. Therefore, with high probability, Alice terminates after $O(2^{\varphi(\lg T + O(1))} + 2^{\varphi \lg(4\ln n)}) = O(T^{\varphi} + \ln^{\varphi} n)$. □

The following theorem on the resource-competitive properties of ROBUSTBROADCAST follows directly from Lemmas 9, 12, and 13

Theorem 4. *Let Carol be an adaptive adversary that jams for T slots.* ROBUSTBROADCAST *guarantees m is received by all receivers and all parties terminate. With high probability, Alice has a cost of $O(T^{\varphi-1} + \ln^{\varphi} n)$, each receiver has a cost of $O(T^{\varphi-1} + \ln n)$, and all parties terminate within $O(T^{\varphi} + \ln^{\varphi} n)$ slots.*

6 Discussion

In this section, we provide some follow-up discussion regarding the practical aspects of wireless LPNs and how our abstract network model is motivated.

Sending and Listening Costs in Practice: Wireless network cards typically offer states such as *sleep, receive (or listen)* and *transmit (or send)*. While the sleep state requires negligible power, the cost of the send and listen states are roughly equivalent and dominate the operating cost of a device. For example, the send and listen costs for the popular Telos motes are 38mW and 35mW, respectively and the sleep state cost is 15μW [66]; therefore, the cost of the send/listen state is more than a factor of 2000 greater and the sleep state cost is negligible. In the context of our work, when a party is not active, we can assume it is in the energy-efficient sleep state.

Disruption may not require jamming an entire slot, and so jamming a slot can be less costly than sending/listening. However, we can assume a small m (or m is broken into small packets) such that jamming and sending costs are within a constant factor of each other.

```
RobustBroadcast for $i \geq \lg(4 \ln n)$
Send Phase: For each of the $2^{\varphi i}$ slots do
    • If Alice has sent in less than $2^{\varphi i+2} \ln n$ slots in this phase, she sends $m$ with probability $\frac{2 \ln n}{2^i}$.
    • If this receiver has listened in less than $2^{i+2}$ slots in this phase, it listens with probability $\frac{2}{2^{(\varphi-1)i}}$.
Any receiver that obtains $m$ terminates.
Nack Phase: For each of the $2^i$ slots do
    • Each receiver that has not terminated sends nack.
    • If Alice has listened in less than $2^{\varphi i+2} \ln n$ slots in this phase, she listens with probability $\frac{4 \ln n}{2^i}$.
If Alice listened to a clear slot, then she terminates.
```

Figure 7: Pseudocode for RobustBroadcast.

Slots: A time-slotted network corresponds to a time division multiple access (TDMA)-like medium access control (MAC) protocol; that is, a time-slotted network. Two examples are the popular IEEE 802.11 family of specifications, and the well-known LEACH [40]. For simplicity, a *global* broadcast schedule is assumed; however, this is likely avoidable if nodes maintain multiple schedules as with S-MAC [83]. Even then, global scheduling has been demonstrated by experimental work in [50] and secure synchronization has been shown [32].

Clear channel assessment (CCA), which subsumes carrier sensing, is a common feature on devices for detecting activity on the channel [67]; this is considered practical under IEEE 802.11 [25]. Collisions are usually only detectable by the receiver [80]. When a collision occurs, a correct node discards any received data. We assume that the absence of channel activity cannot be forged by the adversary; this aligns with the empirical work by Niculescu [61] who shows that channel interference increases linearly with the combined rate of the sources. Finally, we also note that several theoretical models feature collision detection (see [3,6,16,34,68]).

On Reactive Adversaries: CCA is performed via the radio chip using the *received signal strength indicator* (RSSI) [41]. If the RSSI value is below a clear channel threshold, then the channel is assumed to be clear [7]. Such detection consumes on the order of 10^{-6} W which is three orders of magnitude smaller than the send/listen costs; therefore, a reactive Carol can detect activity (but not message content) at essentially zero cost. However, listening to even a small portion of a message costs on the order of milliwatts and our argument from Section 4 now applies.

Cryptographic Authentication: We assume that the message m from Alice can be authenticated. Therefore, Carol cannot spoof Alice. Several results show how light-weight cryptographic authentication can be implemented in sensor networks [42, 47, 53, 77, 78].

What about having a shared secret for Alice and Bob that helps coordinate their communication? This is outside the scope of this work, but we remark that the adversary may capture a limited number of parties (such as Bob). These parties are said to suffer a Byzantine fault and are controlled by the adversary [77, 80]. Given this attack, we emphasize that, while we assume authentication on Alice's side, attempts to share a secret send/listen schedule between Alice and Bob allows Carol to manipulate parties in ways that appear difficult to overcome.

7 Conclusion and Future Work

In this paper, we provided resource-competitive algorithms for mitigating jamming attacks in wireless LPNs. We see that the golden ratio arises naturally from our analysis, and its appearance in this adversarial setting is interesting. Notably, a later result in [35] demonstrates that our result for Alice and Bob is asymptotically tight in that $\Omega(T^{\varphi-1})$ expected cost is necessary.

Future work includes pursuing resource-competitive algorithms to mitigate jamming in multihop networks. Additionally, the application of these results to problems of consensus and leader election may prove fruitful. Finally, investigations into more sophisticated communication models, such as the signal-to-interference-plus-noise (SINR) model, may be of interest.

References

[1] Ittai Abraham, Lorenzo Alvisi, and Joseph Y. Halpern. Distributed Computing Meets Game Theory: Combining Insights from Two Fields. *SIGACT News*, 42(2):69–76, June 2011.

[2] Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Mike Dahlin, Jean-Philippe Martin, and Carl Porth. BAR Fault Tolerance for Cooperative Services. In *Proceedings of the* 20^{th} *ACM Symposium on Operating systems Principles*, pages 45–58, 2005.

[3] Dan Alistarh, Seth Gilbert, Rachid Guerraoui, Zarko Milosevic, and Calvin Newport. Securing Your Every Bit: Reliable Broadcast in Byzantine Wireless

Networks. In *Proceedings of the Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 50–59, 2010.

[4] Nils Aschenbruck, Elmar Gerhards-Padilla, and Peter Martini. Simulative Evaluation of Adaptive Jamming Detection in Wireless Multi-hop Networks. In *Proceedings of the* 30^{th} *International Conference on Distributed Computing Systems Workshops*, pages 213–220, 2010.

[5] Farhana Ashraf, Yih-Chun Hu, and Robin Kravets. Demo: Bankrupting the Jammer. In *Proceedings of the* 9^{th} *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2011.

[6] 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)*, pages 45–54, 2008.

[7] J. Bardwell. Converting Signal Strength Percentage to dBm Values, 2002.

[8] Emrah Bayraktaroglu, Christopher King, Xin Liu, Guevara Noubir, Rajmohan Rajaraman, and Bishal Thapa. On the Performance of IEEE 802.11 Under Jamming. In *Proceedings of the International Conference on Computer Communications (INFOCOM)*, pages 1265–1273, 2008.

[9] Michael A. Bender, Jeremy T. Fineman, Seth Gilbert, and Maxwell Young. How to Scale Exponential Backoff: Constant Throughput, Polylog Access Attempts, and Robustness. In *Proceedings of the* 27^{th} *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 636–654, 2016.

[10] Michael A. Bender, Jeremy T. Fineman, Mahnush Movahedi, Jared Saia, Varsha Dani, Seth Gilbert, Seth Pettie, and Maxwell Young. Resource-Competitive Algorithms. *SIGACT News*, 46(3):57–71, September 2015.

[11] Marin Bertier, Anne-Marie Kermarrec, and Guang Tan. Brief Announcement: Reliable Broadcast Tolerating Byzantine Faults in a Message-Bounded Radio Network. In *Proceedings of the* 22^{nd} *International Symposium on Distributed Computing (DISC)*, pages 516–517, 2008.

[12] Marin Bertier, Anne-Marie Kermarrec, and Guang Tan. Message-Efficient Byzantine Fault-Tolerant Broadcast in a Multi-Hop Wireless Sensor Network. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, pages 408–417, 2010.

[13] Vartika Bhandari and Nitin H. Vaidya. On Reliable Broadcast in a Radio Network. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 138–147, 2005.

[14] Vartika Bhandari and Nitin H. Vaidya. On Reliable Broadcast in a Radio Network: A Simplified Characterization. Technical report, CSL, UIUC, May 2005.

[15] Vartika Bhandari and Nitin H. Vaidya. Reliable Broadcast in Wireless Networks with Probabilistic Failures. In *Proceedings of the International Conference on Computer Communications (INFOCOM)*, pages 715–723, 2007.

[16] Vartika Bhandhari, Jonathan Katz, Chiu-Yuen Koo, and Nitin Vaidya. Reliable Broadcast in Radio Networks: The Bounded Collision Case. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 258 – 264, 2006.

[17] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, and Dan S. Wallach. Secure Routing for Structured Peer-to-Peer Overlay Networks. In 5^{th} *Usenix Symposium on Operating Systems Design and Implementation (OSDI)*, pages 299–314, 2002.

[18] Steve Schmadeke Chicago Tribune. Man Accused of Jamming Calls on Red Line 'disturbed by people talking around him', 2016. `http://www.chicagotribune.com/news/local/breaking/ct-cell-phone-jamming`

[19] Allen Clement, Jeff Napper, Harry Li, Jean-Philipe Martin, Lorenzo Alvisi, and Michael Dahlin. Theory of BAR Games. In *Proceedings of the* 26^{th} *Annual ACM Symposium on Principles of Distributed Computing*, pages 358–359, 2007.

[20] David Goldman CNN. FCC to Marriott: Never Try to Block Wi-Fi Again, 2015. `http://money.cnn.com/2015/01/27/technology/fcc-wifi-hotel/`.

[21] Crossbow. MICAz Wireless Measurement System. `http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf`.

[22] Varsha Dani. Resource-competitive error correction. In *Proceedings of the 10th ACM International Workshop on Foundations of Mobile Computing*, pages 53–58, 2014.

[23] Varsha Dani, Tom Hayes, Mahnush Movahedi, Jared Saia, and Maxwell Young. Interactive Communication with Unknown Noise Rate. *Information and computation*, 2017. In press.

[24] Varsha Dani, Mahnush Movahedi, Jared Saia, and Maxwell Young. Interactive Communication with Unknown Noise Rate. In *Proceedings of the Colloquium on Automata, Languages, and Programming (ICALP)*, 2015.

[25] Jing Deng, Pramod K. Varshney, and Zygmunt J. Haas. A New Backoff Algorithm for the IEEE 802.11 Distributed Coordination Function. `http://surface.syr.edu/eecs/85/`.

[26] Shlomi Dolev, Seth Gilbert, Rachid Guerraoui, Fabian Kuhn, and Calvin Newport. The Wireless Synchronization Problem. In *Proceedings of the* 28^{th} *ACM symposium on Principles of distributed computing*, Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), pages 190–199, 2009.

[27] Shlomi Dolev, Seth Gilbert, Rachid Guerraoui, and Calvin Newport. Gossiping in a Multi-channel Radio Network: An Oblivious Approach to Coping with Malicious Interference. In *Proceedings of the International Symposium on Distributed Computing (DISC)*, pages 208–222, 2007.

[28] Shlomi Dolev, Seth Gilbert, Rachid Guerraoui, and Calvin Newport. Secure Communication over Radio Channels. In *Proceedings of the Symposium on Principles of Distributed Computing (PODC)*, pages 105–114, 2008.

[29] John Douceur. The Sybil Attack. In *Proceedings of the Second International Peer-to-Peer Symposium (IPTPS)*, pages 251–260, 2002.

[30] Cynthia Dwork and Moni Naor. Pricing via Processing or Combatting Junk Mail. In *Proceedings of the* 12^{th} *Annual International Cryptology Conference on Advances in Cryptology*, pages 139–147, 1993.

[31] Yuval Emek and Roger Wattenhofer. Frequency Hopping against a Powerful Adversary. In *Proceedings of the* 27^{th} *International Symposium Distributed Computing (DISC)*, pages 329–343, 2013.

[32] Saurabh Ganeriwal, Christina Pöpper, Srdjan Čapkun, and Mani B. Srivastava. Secure Time Synchronization in Sensor Networks. *ACM Transactions on Information and System Security*, 11(23), 2008.

[33] Seth Gilbert, Rachid Guerraoui, Dariusz Kowalski, and Calvin Newport. Interference-Resilient Information Exchange. In *Proceedings of the International Conference on Computer Communications (INFOCOM)*, pages 2249–2257, 2009.

[34] Seth Gilbert, Rachid Guerraoui, and Calvin C. Newport. Of Malicious Motes and Suspicious Sensors: On the Efficiency of Malicious Interference in Wireless Networks. In *International Conference On Principles Of Distributed Systems (OPODIS)*, pages 215–229, 2006.

[35] Seth Gilbert, Valerie King, Seth Pettie, Ely Porat, Jared Saia, and Maxwell Young. (Near) Optimal Resource-Competitive Broadcast with Jamming. In *Proceedings of the 26^{th} ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '14, pages 257–266, 2014.

[36] Seth Gilbert, Valerie King, Jared Saia, and Maxwell Young. Resource-Competitive Analysis: A New Perspective on Attack-Resistant Distributed Computing. In *Proceedings of the 8^{th} ACM International Workshop on Foundations of Mobile Computing*, 2012.

[37] Seth Gilbert, Calvin Newport, and Chaodong Zheng. Who Are You? Secure Identities in Ad Hoc Networks. In *Proceedings of the 28^{th} International Symposium on Distributed Computing (DISC)*, pages 227–242, 2014.

[38] Seth Gilbert and Maxwell Young. Making Evildoers Pay: Resource-Competitive Broadcast in Sensor Networks. In *Proceedings of the 31^{th} Symposium on Principles of Distributed Computing (PODC)*, pages 145–154, 2012.

[39] Seth Gilbert and Chaodong Zheng. SybilCast: Broadcast on the Open Airwaves. In *Proceedings of the 25^{th} Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 130–139, 2013.

[40] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proceedings of the 33^{rd} Hawaii International Conference on System Sciences (HICSS)*, pages 3005–3014, 2000.

[41] Kannan Srinivasan and P. Levis. RSSI is Under Appreciated. In*EmNets*, 2006.

[42] Chris Karlof, Naveen Sastry, and David Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. In *Proceedings of the 2^{nd} International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 162–175, 2004.

[43] Valerie King, Cynthia Phillips, Jared Saia, and Maxwell Young. Sleeping on the Job: Energy-Efficient and Robust Broadcast for Radio Networks. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 243–252, 2008.

[44] Valerie King, Cynthia A. Phillips, Jared Saia, and Maxwell Young. Sleeping on the Job: Energy-Efficient and Robust Broadcast for Radio Networks. *Algorithmica*, 61(3):518–554, 2011.

[45] Valerie King, Jared Saia, and Maxwell Young. Conflict on a Communication Channel. In *Proceedings of the* 30^{th} *Symposium on Principles of Distributed Computing (PODC)*, pages 277–286, 2011.

[46] Chiu-Yuen Koo. Broadcast in Radio Networks Tolerating Byzantine Adversarial Behavior. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 275–282, 2004.

[47] Y. W. Law, J. Doumen, and P. Hartel. Survey and Benchmark of Block Ciphers for Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 2(1):65–93, 2006.

[48] Frank Li, Prateek Mittal, Matthew Caesar, and Nikita Borisov. SybilControl: Practical Sybil Defense with Computational Puzzles. In *Proceedings of the Seventh ACM Workshop on Scalable Trusted Computing*, STC '12, pages 67–78, 2012.

[49] Harry C. Li, Allen Clement, Edmund L. Wong, Jeff Napper, Indrajit Roy, Lorenzo Alvisi, and Michael Dahlin. BAR gossip. In *Proceedings of the Seventh Symposium on Operating systems Design and Implementation*, pages 191–204, 2006.

[50] Yuan Li, Wei Ye, and John Heidemann. Energy and Latency Control in Low Duty Cycle MAC Protocols. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 676–682, 2005.

[51] Marc Lichtman, Jeffrey H. Reed, T. Charles Clancy, and Mark Norton. Vulnerability of LTE to hostile interference. In *IEEE Global Conference on Signal and Information Processing*, pages 285–288, 2013.

[52] Guolong Lin and Guevara Noubir. On Link Layer Denial of Service in Data Wireless LANs. *Wireless Communications & Mobile Computing*, 5(3):273–284, 2005.

[53] Donggang Liu and Peng Ning. Multi-Level μTESLA: Broadcast Authentication for Distributed Sensor Networks. *ACM Transactions in Embedded Computing Systems*, 3:800–836, 2004.

[54] Xin Liu, Guevara Noubir, Ravi Sundaram, and San Tan. SPREAD: Foiling Smart Jammers using Multi-layer Agility. In *Proceedings of the International Conference on Computer Communications (INFOCOM)*, pages 2536–2540, 2007.

[55] Xin Liu, Xiaowei Yang, and Yong Xia. NetFence: Preventing Internet Denial of Service From Inside Out. In *Proceedings of the ACM SIGCOMM 2010 Conference*, pages 255–266, 2010.

[56] Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Bacşar, and Jean-Pierre Hubaux. Game Theory Meets Network Security and Privacy. *ACM Comput. Surv.*, 45(3):25:1–25:39, July 2013.

[57] Dominic Meier, Yvonne Anne Pignolet, Stefan Schmid, and Roger Wattenhofer. Speed Dating Despite Jammers. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 1–14, 2009.

[58] Ralph C. Merkle. Secure Communications over Insecure Channels. *Communications of the ACM*, 21(4):294–299, April 1978.

[59] Rajeev Motwani and Prbhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[60] Vishnu Navda, Aniruddha Bohra, Samrat Ganguly, and Dan Rubenstein. Using Channel Hopping to Increase 802.11 Resilience to Jamming Attacks. In *Proceedings of the International Conference on Computer Communications (INFOCOM)*, pages 2526–2530, 2007.

[61] Dragoş Niculescu. Interference Map for 802.11 Networks. In *Internet Measurement Conference (IMC)*, pages 339–350, 2007.

[62] Noam Nisan, Time Roughgarden, Éva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[63] Adrian Ogierman, Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Competitive MAC under Adversarial SINR. In *Proceedings of the International Conference on Computer Communications (INFOCOM)*, pages 2751–2759, 2014.

[64] Bryan Parno, Dan Wendlandt, Elaine Shi, Adrian Perrig, Bruce Maggs, and Yih-Chun Hu. Portcullis: Protecting Connection Setup from Denial-of-Capability Attacks. In *Proceedings of the ACM SIGCOMM 2007 Conference*, pages 289–300, 2007.

[65] Andrzej Pelc and David Peleg. Feasibility and Complexity of Broadcasting with Random Transmission Failures. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, pages 334–341, 2005.

[66] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: Enabling Ultra-Low Power Wireless Research. In *IPSN*, 2005.

[67] Iyappan Ramachandran and Sumt Roy. Clear Channel Assessment in Energy-Constrained Wideband Wireless Networks. *IEEE Wireless Communications*, 14(3):70–78, 2007.

[68] Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. A Jamming-Resistant MAC Protocol for Multi-Hop Wireless Networks. In *Proceedings of the International Symposium on Distributed Computing (DISC)*, pages 179–193, 2010.

[69] Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Competitive and Fair Medium Access Despite Reactive Jamming. In *Proceedings of the* 31^{st} *International Conference on Distributed Computing Systems (ICDCS)*, pages 507–516, 2011.

[70] Andrea Richa, Christian Scheideler, Stefan Schmid, and Jin Zhang. Competitive and Fair Throughput for Co-Existing Networks Under Adversarial Interference. In *Proceedings of the* 31^{st} *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 291–300, 2012.

[71] Sara Robinson. The Price of Anarchy. *SIAM News*, 37(5):1–4, 2004.

[72] D. Sleator and R. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Communications of the ACM*, 28(2):202–208, 1985.

[73] Florian Tegeler and Xiaoming Fu. SybilConf: Computational Puzzles for Confining Sybil Attacks. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM)*, pages 1–2, 2010.

[74] Vinod Vaikuntanathan. Brief announcement: Broadcast in Radio Networks in the Presence of Byzantine Adversaries. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, 2005.

[75] Xavier Vilaça, Oksana Denysyuk, and Luís Rodrigues. Asynchrony and Collusion in the n-Party BAR Transfer Problem. In *Proceedings of the* 19^{th} *International Conference on Structural Information and Communication Complexity (SIROCCO)*, pages 183–194, 2012.

[76] Michael Walfish, Mythili Vutukuru, Hari Balakrishnan, David Karger, and Scott Shenker. DDoS Defense by Offense. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pages 303–314, 2006.

[77] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary. *Security in Distributed, Grid, Mobile, and Pervasive Computing. Chapter 17: Wireless Sensor Network Security: A Survey*. Auerbach Publications, 2007.

[78] R. Watro, D. Kong, S. Cuti, C. Gariner, C. Lynn, and P. Kruus. TinyPK: Securing Sensor Networks with Public Key Technology . In *Proceedings of the* 2^{nd} *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 59–64, 2004.

[79] Matthias Wilhelm, Ivan Martinovic, Jens B. Schmitt, and Vincent Lenders. Short Paper: Reactive Jamming in Wireless Networks: How Realistic is the Threat? In *Proceedings of the Fourth ACM Conference on Wireless Network Security*, WiSec, pages 47–52, 2011.

[80] Anthony D. Wood and John A. Stankovic. Denial of Service in Sensor Networks. *Computer*, 35(10):54–62, 2002.

[81] Wenyuan Xu, Ke Ma, Wade Trappe, and Yanyong Zhang. Jamming Sensor Networks: Attack and Defense Strategies. IEEE *Networks*, 20(3):41–47, 2006.

[82] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 46–57, 2005.

[83] Wei Ye, John Heidemann, and Deborah Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the International Conference on Computer Communications (INFOCOM)*, pages 1567–1576, 2002.

[84] Maxwell Young and Raouf Boutaba. Overcoming Adversaries in Sensor Networks: A Survey of Theoretical Models and Algorithmic Approaches for Tol-

erating Malicious Interference. *IEEE Communications Surveys & Tutorials*, 13(4):617–641, 2011.