A resource-competitive jamming defense

Valerie King1 · Seth Pettie2 · Jared Saia3 · Maxwell Young4

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 incurred by correct network devices (i.e., Alice and Bob) is parameterized by the cost incurred by 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 \( m \).

Keywords Resource-competitive algorithms · Jamming attacks · Wireless sensor networks · Internet of things · Security

1 Introduction

The wireless communication medium is vulnerable to jamming whereby an attacker causes interference on the communication channel. Such attacks are prohibited in some countries—for example, in the United States [32]—however, they do occur in practice (see [21,78]) 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 [53,87]. 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...
Let $\text{Cost}(\alpha, v)$ denote the resource expenditure (or cost) to a good node $v$ over an execution $\alpha$. A resource might be bandwidth, CPU cycles, energy, actual money, or another useful domain-specific measure. $\text{Cost}(\alpha, v)$ is the cost incurred by $v$ for executing the actions prescribed by $\mathcal{A}$ in an execution $\alpha$.

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

**Definition 1** Algorithm $\mathcal{A}$ is $(\rho, \tau)$-resource competitive if $\max_{v \in G} \{\text{Cost}(v)\} \leq \rho(T) + \tau$ 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 $\tau$. The function $\rho$ 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 $\rho$ 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 $\tau$? Note that when $T = 0$—that is, there is no disruption—the good nodes clearly cannot incur less than zero cost; $\tau$ represents the unavoidable cost to attain a goal even in the absence of disruption. We desire a small $\tau$ value since this implies that the algorithm incurs a low cost when the level of disruption is low or non-existent. It useful to make this separation explicit via $\tau$, 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 $OPT$ [76]. 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 $OPT$, a resource-
competitive algorithm can actually outperform the adversary by achieving \( \rho(T) = o(T) \).

In the context of competitive analysis, resource augmentation (see [43, 68]) allows for the online algorithm to possess additional resources relative to \( \text{OPT} \), such as a faster processor or a larger cache size. This approach shares common ground in that it incorporates the notion that resources are held by both the online algorithm and by \( \text{OPT} \); moreover, there is a disparity in that the online algorithm holds these additional resources. This issue of resource disparity has some relationship to the resource-competitive framework: if we consider an algorithm that achieves a cost of \( o(T) \), then this is similar to being able to tolerate a disparity in resources between algorithm and the adversary. However, outside of this observation, resource augmentation differs from resource competitiveness for the same reasons discussed above.

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 [64, 75]. 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, 51, 81]). 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 (1) how much the adversary must spend in order to compromise the cryptosystem, and (2) 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 \( \tau \). 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 [60] 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 [50, 79] or monetary penalties [18]. 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 [57, 66, 82]). 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

**Applied security** Several works address applied security considerations with respect to jamming [5, 9, 54, 88] where the adversary deliberately disrupts the communication medium. Defenses include spread spectrum techniques (frequency or channel hopping), mapping with rerouting, and others (see [56, 62, 86, 87]).

Applied work by Ashraf et al. [6] 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 network traffic is similar to our approach to foiling reactive jammers (see Sect. 4). However, on the whole, their approach is quite different from our own; moreover, analytical results are not provided in [6].

**Resource competitiveness** There are a handful of resource-competitive results that address the problem of communicating in the face of jamming. We discuss these results here in order to provide context for the results in this paper. Additionally, Table 1 summarizes the main attributes of the most-closely related prior work [36] in order to contrast them alongside the results in this paper.

Consider an adversary that is able to divide the set of nodes into \( \kappa \geq 1 \) subsets where (1) the members of each subset witness the same slots being jammed (assuming these nodes listen to the channel) and (2) any two different subsets may witness different slots being jammed. Such an adversary is referred to as \( \kappa \)-uniform in the literature [72]. Generally, the fact that not all nodes witness the same slots being jammed can introduce additional challenges.

In this work, we consider an \( n \)-uniform adversary. For example, consider two nodes: a single sender and a single receiver. During any slot, the receiver can be jammed without the sender knowing this, and vice versa (see Sect. 1.4); in other words, this is a 2-uniform adversary. For a general setting with \( n \) nodes, none of the \( n \) nodes may agree on which slots were jammed (in other words, they may each witness a different set of jammed slots). While our result in Sect. 5 addresses a communication problem in this context, we note that the solution assumes that \( \ln n \) is known.
In the closely-related subsequent work, Gilbert et al. [36] consider the less-powerful 1-uniform adversary, but \( n \) (and \( \ln n \)) is completely unknown. In this setting, the authors provide a Monte Carlo broadcast algorithm that allows all \( n \) nodes to receive a message with high probability in \( n \).

The authors derive a lower bound to show that their algorithm is asymptotically near-optimal. As an additional lower-bound result in [36], the Las Vegas algorithm considered in this current work is shown to be asymptotically-optimal for the case of a single sender and single receiver with a 2-uniform adversary.

There are three additional resource-competitive results that address interference on a communication channel; however, in each case, the model used is significantly different from that used in our current work.

First, Gilbert and Young [40] give a Monte Carlo partial broadcast algorithm that is resource-competitive and succeeds w.h.p. against an \( n \)-uniform adversary. However, their model differs substantially from ours in that the adversary is limited to jamming \( O(n^{1+1/k}) \) slots, for some integer \( k \geq 2 \). Similarly, each of the \( n \) nodes is allowed to send or listen on the channel for \( O(n^{1/k}) \) slots. Of the \( n \) good nodes, at least \( (1 - \epsilon) n \) nodes receive the message for an arbitrarily small constant \( \epsilon > 0 \). The expected cost is per node is \( \tilde{O}(T^{1+\epsilon}) \), where \( \tilde{O} \) denotes the existence of polylogarithmic terms, and this is tight w.h.p. Additionally, w.h.p. all nodes terminate in \( O(n^{1+\epsilon}) \) slots. While this result proves that such a powerful adversary can be tolerated, there are two shortcomings: the result (1) critically depends on knowing a constant-factor approximation to \( n \), and (2) still allows the adversary to prevent a small, but constant, fraction of the nodes from receiving the broadcast.

Second, Bender et al. [10] demonstrate that contention resolution—how to coordinate access to a shared channel by multiple devices—can be accomplished despite jamming. Here, the adversary is 1-uniform. 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.

An important difference in the model used in [10] is that there is no need to coordinate with a receiver. Each device attempts to transmit its message and is successful if it is the only sender in that slot and the slot is not jammed. This model is motivated by an “energy-rich” base station that can listen to the channel constantly.

Third, while the problem is not closely-related to jamming mitigation, it is worth mentioning the challenge of accomplishing interactive communication on noisy channels. This has been examined from a resource-competitive perspective, where results by Dani [22] and by Dani et al. [23, 24] address an adversary that can flip an unknown number of bits as information is transmitted across a channel. In addition to addressing bit flips rather than jammed communications, another major difference is that the goal is to minimize the latency of communication rather than energy expenditure. Additional theoretical results Outside of resource-competitive results, there is large body of work on mitigating jamming attacks in wireless sensor networks (see [90] for a survey).

Anantharamu et al. [4] examine packet delivery on a multiple access channel under dynamic arrivals subject to an injection rate and a jamming rate. In particular, a \((\rho, \lambda, b)\)-adversary can inject \( \rho |\tau| + b \) packets and jam at most \( \lambda |\tau| + b \) slots within in any contiguous segment \( \tau \) consisting of \( |\tau| \) slots, where \( \lambda < 1 \). Upper bounds on the worst-case latency are derived in this setting.

Gilbert et al. [35] examine 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.

Gilbert and Kowalski [38] examine the impact of trusted computing (i.e., a tamper-proof component) on resilience to jamming attacks when at most \( k \) devices wish to exchange information. The non-reactive adversary is permitted to jam only when the tamper-proof component on a malicious device permits activation of the onboard transceiver. In this setting, nearly-tight upper and lower bounds are derived for the communication latency, and these results provide evidence that trusted computing can have a significant impact on performance.

Pelec and Peleg [67] examine an adversary that randomly corrupts messages. Results by Awerbuch et al. [7] and Richa et al. [72–74] consider an adversary whose jamming is bounded within any sufficiently large time window. Under this adversarial model, Ogierman et al. [65] study medium access in the signal-to-interference-plus-noise ratio communication model. In the context of Sybil attacks [29], Gilbert and Zheng [41] devise methods for tolerating a jamming adversary using coordination amongst the communicating parties, and this is generalized to a completely decentralized scenario by Gilbert et al. [39].

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. [34] derive bounds on the time required for information exchange when a reactive adversary jams multiple channels. Meier et al. [59] 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. [58] 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 [12, 14–16, 45, 46, 48, 80]. Listening costs are accounted for by King et al. [45,46], 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. [17] 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. [13]; however, each node spends significant time in the costly listening state.

Finally, a preliminary version of our results appeared in [47]; 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 [47] (discussed in Sect. 2.4). Finally, we note that the lower bound originally provided in [47] is not provided here, since more recent work in [36] 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.

**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 (9 µs) 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 [69] are 38 and 35 mW, 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 [20].

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 Sect. 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.\(^2\) 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 Sect. 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.

### 1.5 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 \( \tau \) 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 \( \Phi \) 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 incurred by Alice and Bob is \( O(T^{0.5} + 1) \). In Case 2, the expected cost incurred by Alice and Bob is \( O(T^{0.62} + 1) \).
- Both parties terminate in an expected \( O(T^2) \) and \( O(T^\Phi) \) 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 Sect. 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 Equivalently, we can consider that Carol is Bob, or controls Bob completely. Conceptually, the only two parties are then Alice and Carol.
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 \( \ell \) 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 (\( \text{nack} \)) message. Else, Bob has received \( m \) and so he terminates the protocol; that is, he does not execute any further actions.

Alice listens in each odd-indexed slot and, if Alice receives a \( \text{nack} \), 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 \( \text{nack} \) but the 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 \textsc{Iteration-1} in Fig. 1. In the Send Phase, Alice and Bob send and listen each with probability \( 2/\sqrt{\ell} \). If Bob receives \( m \), he terminates the protocol at the end of the Send Phase; that is, he does not proceed to the subsequent \textsc{Nack Phase} or any other subsequent rounds.

In the \textsc{Nack Phase}, if Bob has not terminated (because he did not receive \( m \)), he sends \( \text{nack} \) to Alice during the single slot asking her to enter into a new round. If Alice hears \( \text{nack} \), or if the slot is jammed, she proceeds into the next round; otherwise, if the slot is clear, she terminates. Let a \textit{round} refer to the execution of a Send Phase and the corresponding \textsc{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 \textit{blocked Send Phase}—then Bob 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 \textsc{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 \( O(k\sqrt{\ell}) = O(T/\sqrt{T}) \) 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.
and so we opt for equality.

\[ \text{Fig. 2 Pseudocode for modified Iteration-2} \]

ITERATION-2 for rounds starting with \( i \geq 1 \)

\begin{align*}
\text{Send Phase:} & \text{ For each of the } 2^i \text{ slots do} \\
& - \text{Alice sends } m \text{ with probability } 2/2^{(c-a)i} \\
& - \text{Bob listens with probability } 2/2^{(c-b)i} \\
\end{align*}

If Bob received \( m \), then he terminates

\begin{align*}
\text{Nack Phase:} & \text{ For 1 slot do} \\
& - \text{Bob sends a nack message} \\
& - \text{Alice listens} \\
\end{align*}

If Alice listened to a clear slot, then she terminates

Figure 2 provides the pseudocode with this modification implemented; we name this intermediate algorithm Iteration-2. The length of the Send Phase is \( 2^i \) 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 costs incurred by Alice and Bob are \( O(2^a) \) and \( O(2^b) \), respectively, 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.

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 Sect. 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:

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

The probability that Bob fails to receive \( m \) over all \( 2^i = 2^i \) slots in the Send Phase is at most:

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

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 \( T = \Omega(2^i) \) 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 incurred by 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 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 \( 2^a \) 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, \( \ell \) 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 \( 2^a \) 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 \( 2^a \) exceeds \( \ell \)? Instead, as we show in Sect. 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 fewer than \( \ell \) whilst spoofing Bob, but Alice’s random strategy is adequate to thwart such behavior. The pseudocode for this modification is given in Fig. 3.
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 $2^a_{i+2}$ slots, she does nothing for the remainder of the phase. Similarly, if Bob has listened in $2^b_{i+2}$ slots, he does nothing 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 experienced by 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 does nothing for the remainder of the phase. The pseudocode for this final design decision is presented in Fig. 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 $4/(2^{ai}+2^{bi})$. 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 Sect. 3, we will show that this a low-probability event and does not impact the expected cost.
Nack failure messages or jams slots in order to trick Alice into thinking a valid Nack message 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 Sect. 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 $a < b$, then the expected cost of Alice is at least as much as the expected cost of 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 Sect. 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 Sect. 2.2, we will need that $a + b = c$ for the analysis to work.

Finally, as discussed in Sect. 2.2, we assume that $c < 2$. 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 prob-
ability 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 Sect. 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 Alice’s expected cost is $2^i$ and when Bob’s expected cost is $2^{i+1}$. Therefore, over epoch $i$, Alice’s expected cost by a factor of at least 2 is less than $e^{-2^{i}/3}$.

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

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

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 $s = 2^i$ be the number of slots in the Send Phase. Let $p_A$ be the probability that Alice sends in a particular slot. Let $p_B$ 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:

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

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:

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

Substituting for each conditional probability, we have:

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

where the last inequality follows from:

$$p_A p_B \sum_{j=1}^{s} q_j \geq (2/2^{(c-b)j})(2/2^{(c-b)j})(s/2)$$
\[
\sum_{i=1}^{\infty} e^{-2(i-1)} O(2^{ai}) + \sum_{i=1}^{\infty} \exp\left(-2^{\Theta(i-1)}\right) O(2^{ai}) = O(1) \sum_{i=1}^{\infty} \left(\frac{2^a}{e^2}\right)^i = O(1)
\]

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 Sect. 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 incurred by Alice is at most:

\[
\sum_{i=1}^{\infty} e^{-2(i-1)} O(2^{ai}) + \sum_{i=1}^{\infty} \exp\left(-2^{\Theta(i-1)}\right) O(2^{ai}) = O(1) \sum_{i=1}^{\infty} \left(\frac{2^a}{e^2}\right)^i = O(1)
\]

where the last line follows from the fact that \(a < 2\) and the sum of a geometric series. Similarly, the expected cost incurred by Bob is at most:

\[
\sum_{i=1}^{\infty} e^{-2(i-1)} O(2^{bi}) + \sum_{i=1}^{\infty} \exp\left(-2^{\Theta(i-1)}\right) O(2^{bi}) \leq O(1) \sum_{i=1}^{\infty} \left(\frac{2^b}{e^2}\right)^i = O(1)
\]

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, \(Pr[\text{Alice retransmits } m\text{ in round } i + 1] = Pr[X_1X_2 \cdots X_i = 1]\). 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, \(Pr[X_1X_2 \cdots X_s = 1] \leq e^{-p \sum_{j=1}^{s} q_j} \leq e^{-2}\).

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 incurred by Alice and Bob is \(O(T^{a/c})\) and \(O(T^{b/c})\), respectively. In Case 2, the expected cost incurred by 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 \(T = \Omega(2^{ci} + 2^{bi})\).

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

\[
O(2^{ai}) + \sum_{k=1}^{\infty} e^{-2(k-1)} O(2^{a(i+k)}) + \sum_{k=1}^{\infty} \exp\left(-2^{\Theta(k-1)}\right) O(2^{a(i+k)}) = O(2^{ai}) + O(2^{ai}) \sum_{k=1}^{\infty} \left(\frac{2^a}{e^2}\right)^k + O(2^{ai}) = O(2^{ai})
\]

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 of 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:

\[
O(2^{aj}) + \sum_{k=1}^{\infty} e^{-2(k-1)} O(2^{a(j+k)}) + \sum_{k=1}^{\infty} \exp\left(-2^{\Theta(k-1)}\right) O(2^{a(j+k)})
\]
= O(2^{aj}) + O(2^{aj}) ∞ \sum_{k=1}^{\infty} \left( \frac{2^{a}}{e^2} \right)^k + O(2^{aj})

= O(2^{aj})

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

Therefore, in Case 2, the total expected cost incurred by Alice is \( O(2^{aj}) \). Since \( T = O(2^{ci} + 2^{bj}) \), this cost as a function of \( T \) is \( O(2^{aj/c} + T^{a/b}) \). For Case 1, there are no blocked Nack Phases and so Alice’s cost is simply \( O(2^{aj}) \). Bob We analyze the expected cost incurred by 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 Sect. 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:

\[ O(2^{bj}) + \sum_{k=1}^{\infty} e^{-2(k-1)} O(2^{(b(i+k)}) \]

\[ + \sum_{k=1}^{\infty} \exp \left( -2\Theta(k-1) \right) O(2^{b(i+k)}) \]

\[ \leq O(2^{bj}) + O(2^{bj}) \sum_{k=1}^{\infty} \left( \frac{2^{b} e}{e^2} \right)^k + O(2^{bj}) \]

\[ = O(2^{bj}) \]

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

We now prove Theorem 1 stated in Sect. 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 Sect. 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 experienced by 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:

\[ \frac{(a + b)b}{a} = \frac{c^2}{b} \]

\[ \Rightarrow \left( 1 + \frac{b}{a} \right) = \left( \frac{c}{b} \right)^2 \]

\[ \Rightarrow \left( 1 + \frac{c}{b} \right) = \left( \frac{c}{b} \right)^2 \text{ since } \frac{a}{b} = \frac{b}{c} \]

\[ \Rightarrow \left( \frac{c}{b} \right)^2 - \left( \frac{c}{b} \right) - 1 = 0 \]

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 \( \sum_{i=2}^{\infty} 2^{bi}/2 \geq T \) 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, \( E[X] \leq (1 - e^{-2}) + 2 e^{-2}(1 - e^{-2}) + 3 e^{-4}(1 - e^{-2}) + \ldots = \sum_{i=1}^{\infty} i e^{2(i-1)}(1 - e^{-2}) = (1 - e^{-2})e^{2} \sum_{i=1}^{\infty} i(e^{-2})^{i} \) 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. \( \square \)

Fig. 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 [85].

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 themessage, 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 \( c' \), \( \epsilon \) 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 \( \delta \), \( c'' \) 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:

\[
\left(1 - \frac{x}{3y}\right) \left(\frac{1}{3}\right) p_B = \left(1 - \frac{1}{2(1-\epsilon)}\right) \left(\frac{1}{3}\right) p_B \geq \left(\frac{1}{3} - \frac{(1 + \delta)}{6}\right) p_B
\]
for sufficiently large $x$ (that reduces the size of $\delta$) when $y > (1 - \varepsilon)(2x/3)$.

The probability that all messages that Alice sends fail to be delivered is at most $(1 - p_B/12)^y + 2/x^c$ where the last term is the probability of the bad event that $y$ or $a'$ is small and $c'' > 0$ is a constant. Redefine $p_B = 24/((1 - \delta)2(\log-1)^i)$ 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 Sect. 3 proceeds almost identically. Therefore, we then have $(1 - p_B/12)^y + 2/x^c' \leq e^{-2}$.

\begin{proof}

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 Sect. 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 [88].

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 $ln$ 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 Fig. 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 $m$. As mentioned in Sect. 1.3, such an adversary is referred to as $n$-uniform [72]. The cost for jamming is still 1 per slot regardless of how many receivers are jammed. We maintain the definition in Sect. 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 a non-failed round. In the multi-receiver case, we can bound the probability of a failed round as a function of $n$.

\begin{lemma}

For any round $i \geq \log(4 \ln n)$ the probability of a failed round is at most $1/n$.
\end{lemma}

\begin{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 $X = \sum_{j=1}^{2^{\psi}} X_j$, then:

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

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 \(2^{(q-1)i} + 4 \ln n\). By Theorem 3, the probability that she reaches her cutoff is at most:

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

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

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

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:

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

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 \(s = 2^i\) 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:

\[
Pr[m \text{ is not delivered in the Send Phase}]
\]

\[
= Pr[X_1X_2 \cdots X_s = 1]
\]

\[
= Pr[X_s = 1 | X_1X_2 \cdots X_{s-1} = 1] \cdot \prod_{i=1}^{s-1} Pr[X_i = 1]
\]

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:

\[
Pr[X_i = 1 | X_1X_2 \cdots X_{i-1} = 1] = 1 - p_A p_R q_j
\]

Substituting for each conditional probability, we have:

\[
Pr[X_1X_2 \cdots X_s = 1] = (1 - p_A p_R q_1) \cdots (1 - p_A p_R q_s)
\]

\[
= \prod_{j=1}^{s} (1 - p_A p_R q_j)
\]

\[
\leq e^{-p_A p_R \sum_{j=1}^{s} q_j}
\]

\[
\leq e^{-(4 \ln n/2^i)(1/2)}
\]

\[
= e^{-2 \ln n}
\]

\[
= n^{-2}
\]

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 incurred by Alice and each receiver is \(O(\ln^8 n)\) 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 incurred by Alice is at most:

\[
\sum_{i=\sigma}^{\infty} (1/n)^{(i-\sigma)} O(2^{(q-1)i} \ln n) = O(\ln^8 n)
\]

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, \(Pr[A\text{ Alice retransmits } m \text{ in round } i + 1] = Pr[X_1X_2 \cdots X_s = 1]\). 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:

\[
Pr[X_1X_2 \cdots X_s = 1] \leq e^{-p \sum_{j=1}^{s} q_j}
\]

\[
\leq e^{-2 \ln n}
\]

\[
\leq n^{-2}
\]

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 \(O(2^{(p-1)i} \ln n)\) 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 \(X = \sum_{j=1}^{2^i} X_j\), we have:

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

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 \(Y = \sum_{j=1}^{2^i} Y_j\), 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, for high probability, Alice’s cost is \(O(2^{(p-1)i} \ln n) + O(\ln n) = O(2^{(p-1)i} \ln n)\).

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

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

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 incurred by Alice and each receiver is \(O(T^{p-1})\).

**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 \(T = \Omega(2^{qi} + 2i)\).

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

\[
O(2^{(p-1)i} \ln n) + \sum_{k=1}^{\infty} (2/n)^{k-1} O(2^{(p-1)(i+k)} \ln n)
= O(2^{(p-1)i} \ln n).
\]

Next, using Lemmas 7, 10, and 11, we calculate the cost incurred by 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:

\[
O(2^{(p-1)j} \ln n) + \sum_{k=1}^{\infty} (2/n)^{k-1} O(2^{(p-1)(j+k)} \ln n)
= O(2^{(p-1)j} \ln n).
\]

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

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 Sect. 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:

\[
O(2^i) + \sum_{k=1}^{\infty} (2/n)^{k-1} O(2^{i+k})
= O(2^i + \ln n)
= O(2^i).
\]

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

Lemma 13 With high probability, Alice and all correct receivers terminate ROBUSTBROADCAST in \(O(T^{p} + \ln^p n)\) 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 \(\sum_{i=1}^{\lg(4 \ln n)} 2^i / 2 \geq T\) 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 \(O(2^{\lg(T+O(1))} + 2^{\lg(4 \ln n)}) = O(T^{p} + \ln^p n)\).

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 $O(T^{p-1} + \ln n)$, each receiver has a cost of $O(T^{p-1} + \ln n)$, and all parties terminate within $O(T^{p} + \ln^{p} n)$ 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 38 and 35 mW, respectively and the sleep state cost is 15 μW [69]; 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.

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. The discretization of time into slots is a common feature among wireless communication standards. For instance, the IEEE 802.15.4 standard employs slots in the beacon (or slotted) mode [70]. Two further examples are the popular IEEE 802.11 family of specifications, and the well-known LEACH [42]. For simplicity, a global broadcast schedule is assumed; however, this is likely avoidable if nodes maintain multiple schedules as with S-MAC [89]. Even then, global scheduling has been demonstrated by experimental work in [52] and secure synchronization has been shown [33].

Clear channel assessment (CCA), which subsumes carrier sensing, is a common feature on devices for detecting activity on the channel [71]; this is considered practical under IEEE 802.11 [25]. Collisions are usually only detectable by the receiver [86]. 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 [63] 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 [3,7,17,35,72]).

On reactive adversaries CCA is performed via the radio chip using the received signal strength indicator (RSSI) [77]. If the RSSI value is below a clear channel threshold, then the channel is assumed to be clear [8]. 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 incurs an energy consumption on the order of milliwatts, and our argument from Sect. 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 [44,49,55,83,84].

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 [83,86]. 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 [36] demonstrates that our result for Alice and Bob is asymptotically tight in that $\Omega(T^{p-1})$ 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.

Acknowledgements We thank Martin Karsten, Srinivasan Keshav, and James Horey for their valuable comments.

References

1. Abraham, I., Alvisi, L., Halpern, J.Y.: Distributed computing meets game theory: combining insights from two fields. SIGACT News 42(2), 69–76 (2011)
2. Aiyer, A.S., Alvisi, L., Clement, A., Dahlin, M., Martin, J.-P., Porth, C.: BAR fault tolerance for cooperative services. In: Proceedings of the 20th ACM Symposium on Operating Systems Principles, pp. 45–58 (2005)
A resource-competitive jamming defense 437

3. Alistarh, D., Gilbert, S., Guerraoui, R., Milosevic, Z., Newport, C.: Securing your every bit: reliable broadcast in byzantine wireless networks. In: Proceedings of the Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 50–59 (2010)

4. Anantharamu, L., Chlebus, B.S., Kowalski, D.R., Rokicki, M.A.: Medium access control for adversarial channels with jamming. In: Proceedings of the 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO), pp. 89–101 (2011)

5. Aschenbruck, N., Gerhrads-Padilla, E., Martini, P.: Simulative evaluation of adaptive jamming detection in wireless multi-hop networks. In: Proceedings of the 30th International Conference on Distributed Computing Systems Workshops, pp. 213–220 (2010)

6. Ashraf, F., Hu, Y.-C., Kravets, R.: Demo: bankrupting the jammer. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys) (2011)

7. Awerbuch, B., Richa, A., Scheidele, C.: A jamming-resistant MAC protocol for single-hop wireless networks. In: Proceedings of the 27th ACM Symposium on Principles of Distributed Computing (PODC), pp. 45–54 (2008)

8. Bardwell, J.: Converting Signal Strength Percentage to dBm Values (2002)

9. Bayraktaroglu, E., King, C., Liu, X., Noubir, G., Rajaraman, R., Thapa, B.: On the performance of IEEE 802.11 under jamming. In: Proceedings of the International Conference on Computer Communications (INFOCOM), pp. 1265–1273 (2008)

10. Bender, M.A., Fineman, J.T., Gilbert, S., Young, M.: How to scale exponential backoff: constant throughput, polylog access attempts, and robustness. In: Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 636–654 (2016)

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

12. Bertier, M., Kermarrec, A.-M., Tan, G.: Brief announcement: reliable broadcast tolerating byzantine faults in a message-bounded radio network. In: Proceedings of the 22nd International Symposium on Distributed Computing (DISC), pp. 516–517 (2008)

13. Bertier, M., Kermarrec, A.-M., Tan, G.: Message-efficient byzantine fault-tolerant broadcast in a multi-hop wireless sensor network. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), pp. 408–417 (2010)

14. Bhandari, V., Vaidya, N.H.: On reliable broadcast in a radio network. In: Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), pp. 138–147 (2005)

15. Bhandari, V., Vaidya, N.H.: On reliable broadcast in a radio network: a simplified characterization. Technical Report. CSL, UIUC (2005)

16. Bhandari, V., Vaidya, N.H.: Reliable broadcast in wireless networks with probabilistic failures. In: Proceedings of the International Conference on Computer Communications (INFOCOM), pp. 715–723 (2007)

17. Bhandari, V., Katz, J., Koo, C.-Y., Vaidya, N.: Reliable broadcast in radio networks: the bounded collision case. In: Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), pp. 258–264 (2006)

18. Castro, M., Druschel, P., Ganesh, A., Rowstron, A., Wallach, D.S.: Secure routing for structured peer-to-peer overlay networks. In: 5th Usenix Symposium on Operating Systems Design and Implementation (OSDI), pp. 299–314 (2002)

19. Clement, A., Napper, J., Li, H., Martin, J.-P., Alvisi, L., Dahlin, M.: Theory of BAR games. In: Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Computing, pp. 358–359 (2007)

20. Crossbow: MICAz Wireless Measurement System. http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

21. David Goldman CNN: FCC to Marriott: Never Try to Block WiFi Again (2015). http://money.cnn.com/2015/01/27/technology/fcc-wifi-hotel/

22. Dani, V.: Resource-competitive error correction. In: Proceedings of the 10th ACM International Workshop on Foundations of Mobile Computing, pp. 53–58 (2014)

23. Dani, V., Hayes, T., Movahedi, M., Saia, J., Young, M.: Interactive communication with unknown noise rate. Inf. Comput. (2017) (In press)

24. Dani, V., Movahedi, M., Saia, J., Young, M.: Interactive communication with unknown noise rate. In: Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP), pp. 575–587 (2015)

25. Deng, J., Varshney, P.K., Haas, Z.J.: A New Backoff Algorithm for the IEEE 802.11 Distributed Coordination Function. http://surface.syr.edu/eeecs/85/

26. Dolev, S., Gilbert, S., Guerraoui, R., Kuhn, F., Newport, C.: The wireless synchronization problem. In: Proceedings of the 28th ACM symposium on Principles of distributed computing. Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), pp. 190–199 (2009)

27. Dolev, S., Gilbert, S., Guerraoui, R., Newport, C.: 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), pp. 208–222 (2007)

28. Dolev, S., Gilbert, S., Guerraoui, R., Newport, C.: Secure communication over radio channels. In: Proceedings of the Symposium on Principles of Distributed Computing (PODC), pp. 105–114 (2008)

29. Doucet, J.: The sybil attack. In: Proceedings of the Second International Peer-to-Peer Symposium (IPTPS), pp. 251–260 (2002)

30. Dwork, C., Naor, M.: Pricing via processing or combatting junk mail. In: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, pp. 139–147 (1993)

31. Emek, Y., Wattenhofer, R.: Frequency hopping against a powerful adversary. In: Proceedings of the 27th International Symposium Distributed Computing (DISC), pp. 329–343 (2013)

32. Federal Communications Commission (FCC): Jammer Enforcement (2017). https://www.fcc.gov/general/jammer-enforcement

33. Ganeswaran, S., Popper, C., Capkun, S., Srivastava, M.B.: Secure time synchronization in sensor networks. ACM Trans. Inf. Syst. Secur. 11, 23 (2008)

34. Gilbert, S., Guerraoui, R., Kowalski, D., Newport, C.: Interference-resilient information exchange. In: Proceedings of the International Conference on Computer Communications (INFOCOM), pp. 2249–2257 (2009)

35. Gilbert, S., Guerraoui, R., Newport, C.C.: Of malicious motes and suspicious sensors: on the efficiency of malicious interference in wireless networks. In: International Conference On Principles Of Distributed Systems (OPODIS), pp. 215–229 (2006)

36. Gilbert, S., King, V., Pettie, S., Porat, E., Saia, J., Young, M.: (Near) optimal resource-competitive broadcast with jamming. In: Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA’14, pp. 257–266 (2014)

37. Gilbert, Seth, King, Valerie, Saia, Jared, Young, Maxwell: Resource-Competitive Analysis: A New Perspective on Attack-Resistant Distributed Computing. In: Proceedings of the 8th ACM International Workshop on Foundations of Mobile Computing, (2012)

38. Gilbert, S., Kowalski, D.R.: Trusted computing for fault-prone wireless networks. In: Proceedings of the 24th International Symposium, Distributed Computing (DISC), pp. 359–373 (2010)

39. Gilbert, S., Newport, C., Zheng, C.: Who are you? Secure identities in ad hoc networks. In: Proceedings of the 28th International Symposium on Distributed Computing (DISC), pp. 227–242 (2014)

40. Gilbert, S., Young, M.: Making evildoers pay: resource-competitive broadcast in sensor networks. In: Proceedings of the 31th Sympo-
sium on Principles of Distributed Computing (PODC), pp. 145–154 (2012)
41. Gilbert, S., Zheng, C.: SybilCast: broadcast on the open airwaves. In: Proceedings of the 25th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 130–139 (2013)
42. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS), pp. 3005–3014 (2000)
43. Kalyanasundaram, B., Pruhs, K.: Speed is as powerful as clairvoyance. J. ACM 47(4), 617–643 (2000)
44. Karlof, C., Sastry, N., Wagner, D.: TinySec: a link layer security architecture for wireless sensor networks. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys), pp. 162–175 (2004)
45. King, V., Phillips, C., Saia, J., Young, M.: Sleeping on the job: energy-efficient and robust broadcast for radio networks. In: Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), pp. 243–252 (2008)
46. King, V., Phillips, C.A., Saia, J., Young, M.: Sleeping on the job: energy-efficient and robust broadcast for radio networks. Algorithmica 61(3), 518–554 (2011)
47. King, V., Saia, J., Young, M.: Conflict on a communication channel. In: Proceedings of the 30th Symposium on Principles of Distributed Computing (PODC), pp. 277–286 (2011)
48. Koo, C.-Y.: Broadcast in radio networks tolerating byzantine adversarial behavior. In: Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), pp. 275–282 (2004)
49. Law, Y.W., Doumen, J., Hartel, P.: Survey and Benchmark of Block Ciphers for Wireless Sensor Networks. ACM Trans. Sens. Netw. 2(1), 65–93 (2006)
50. Li, F., Mittal, P., Caesar, M., Borisov, N.: SybilControl: practical sybil defense with computational puzzles. In: Proceedings of the Seventh ACM Workshop on Scalable Trusted Computing, STC’12, pp. 67–78 (2012)
51. Li, H.C., Clement, A., Wong, E.L., Napper, J., Roy, I., Alvisi, L., Dahlin, M.: BAR gossip. In: Proceedings of the Seventh Symposium on Operating Systems Design and Implementation, pp. 191–204 (2006)
52. Li, Y., Ye, W., Heidemann, J.: Energy and latency control in low duty cycle MAC protocols. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), pp. 676–682 (2005)
53. Lichtman, M., Reed, J.H., Clancy, T.C., Norton, M.: Vulnerability of LTE to hostile interference. In: IEEE Global Conference on Signal and Information Processing, pp. 285–288 (2013)
54. Lin, G., Noubir, G.: On link layer denial of service in data wireless LANs. Wirel. Commun. Mob. Comput. 5(3), 273–284 (2005)
55. Liu, D., Ning, P.: Multi-level μTESLA: broadcast authentication for distributed sensor networks. ACM Trans. Embed. Comput. Syst. 3, 800–836 (2004)
56. Liu, X., Noubir, G., Sundaram, R., Tan, S.: SPREAD: foiling smart jammers using multi-layer agility. In: Proceedings of the International Conference on Computer Communications (INFOCOM), pp. 2536–2540 (2007)
57. Liu, X., Yang, X., Xia, Y.: NetFence: preventing internet denial of service from inside out. In: Proceedings of the ACM SIGCOMM 2010 Conference, pp. 255–266 (2010)
58. Manshaei, M.H., Zhu, Q., Tansu, A., Bačar, T., Hubaux, J.-P.: Game theory meets network security and privacy. ACM Comput. Surv. 45(3), 25:1–25:39 (2013)
59. Meier, D., Pignolet, Y.A., Schmid, S., Wattenhofer, R.: Speed dating despite jammers. In: Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 1–14 (2009)
60. Merkle, R.C.: Secure communications over insecure channels. Commun. ACM 21(4), 294–299 (1978)
61. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press, Cambridge (1995)
62. Navda, V., Bohra, A., Gargaly, S., Rubenstein, D.: Using channel hopping to increase 802.11 resilience to jamming attacks. In: Proceedings of the International Conference on Computer Communications (INFOCOM), pp. 2526–2530 (2007)
63. Niculescu, D.: Interference map for 802.11 networks. In: Internet Measurement Conference (IMC), pp. 339–350 (2007)
64. Nisan, N., Roughgarden, T., Tardos, É., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, Cambridge (2007)
65. Ogierman, A., Richa, A., Scheideler, C., Schmid, S., Zhang, J.: Competitive MAC under adversarial SINR. In: Proceedings of the International Conference on Computer Communications (INFOCOM), pp. 2751–2759 (2014)
66. Paro, B., Wendlandt, D., Shi, E., Perrig, A., Maggs, B., Hu, Y.-C.: Portcullis: protecting connection setup from denial-of-capability attacks. In: Proceedings of the ACM SIGCOMM 2007 Conference, pp. 289–300 (2007)
67. Pelc, A., Peleg, D.: Feasibility and complexity of broadcasting with random transmission failures. In: Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), pp. 334–341 (2005)
68. Phillips, C.A., Stein, C., Torng, E., Wein, J.: Optimal time-critical scheduling via resource augmentation. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC), pp. 140–149 (1997)
69. Polastre, J., Szewczyk, R., Culler, D.: Telos: enabling ultra-low power wireless research. In: IPSN (2005)
70. Ramachandran, I., Das, A.K., Roy, S.: Analysis of the contention access period of IEEE 802.15.4 MAC. ACM Trans. Sens. Netw. 3(1), 4 (2007)
71. Ramachandran, I., Roy, S.: Clear channel assessment in energy-constrained wideband wireless networks. IEEE Wirel. Commun. 14(3), 70–78 (2007)
72. Richa, A., Scheideler, C., Schmid, S., Zhang, J.: A jamming-resistant mac protocol for multi-hop wireless networks. In: Proceedings of the International Symposium on Distributed Computing (DISC), pp. 179–193 (2010)
73. Richa, A., Scheideler, C., Schmid, S., Zhang, J.: Competitive and fair medium access despite reactive jamming. In: Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS), pp. 507–516 (2011)
74. Richa, A., Scheideler, C., Schmid, S., Zhang, J.: Competitive and fair throughput for co-existing networks under adversarial interference. In: Proceedings of the 31st ACM Symposium on Principles of Distributed Computing (PODC), pp. 291–300 (2012)
75. Robinson, S.: The price of anarchy. SIAM News 37(5), 1–4 (2004)
76. Sleator, D., Tarjan, R.: Amortized efficiency of list update and paging rules. Commun. ACM 28(2), 202–208 (1985)
77. Srinivasan, K., Levis, P.: RSSI is under appreciated. In: EmNets (2005)
78. 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/cell-phone-jamming-red-line-20160609-story.html
79. Tsugui, F., Fu, X.: SybilConf: computational puzzles for confining sybil attacks. In: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM), pp. 1–2 (2010)
80. Vaiukonutav, V.: 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)
81. Vilaça, X., Denysyuk, O., Rodrigues, L.: Asynchrony and collusion in the n-party BAR transfer problem. In: Proceedings of the
82. Walfish, M., Vutukuru, M., Balakrishnan, H., Karger, D., Shenker, S.: DDoS Defense by Offense. In: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pp. 303–314 (2006)

83. Walters, J.P., Liang, Z., Shi, W., Chaudhary, V.: Security in Distributed, Grid, Mobile, and Pervasive Computing. Chapter 16: Wireless Sensor Network Security: A Survey. Auerbach Publications, Boca Raton (2007)

84. Watro, R., Kong, D., Cuti, S., Gariner, C., Lynn, C., Kruus, P.: TinyPK: securing sensor networks with public key technology. In: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN), pp. 59–64 (2004)

85. Wilhelm, M., Martinovic, I., Schmitt, J.B., Lenders, V.: Short paper: reactive jamming in wireless networks: how realistic is the threat? In: Proceedings of the Fourth ACM Conference on Wireless Network Security, pp. 47–52. WiSec (2011)

86. Wood, A.D., Stankovic, J.A.: Denial of service in sensor networks. Computer 35(10), 54–62 (2002)

87. Xu, W., Ma, K., Trappe, W., Zhang, Y.: Jamming sensor networks: attack and defense strategies. IEEE Netw. 20(3), 41–47 (2006)

88. Xu, W., Trappe, W., Zhang, Y., Wood, T.: 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), pp. 46–57 (2005)

89. Ye, W., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In: Proceedings of the International Conference on Computer Communications (INFOCOM), pp. 1567–1576 (2002)

90. Young, M., Boutaba, R.: Overcoming adversaries in sensor networks: a survey of theoretical models and algorithmic approaches for tolerating malicious interference. IEEE Commun. Surv. Tutor. 13(4), 617–641 (2011)