Of Malicious Motes and Suspicious Sensors: On the Efficiency of Malicious Interference in Wireless Networks

Seth Gilbert[1], Rachid Guerraoui[2], and Calvin Newport[1]

[1] MIT CSAIL
{sethg, cnewport}@mit.edu
[2] EPFL IC
rachid.guerraoui@epfl.ch

Abstract. How efficiently can a malicious device disrupt communication in a wireless network? Imagine a basic game involving two honest players, Alice and Bob, who want to exchange information, and an adversary, Collin, who can disrupt communication using a limited budget of β broadcasts. How long can Collin delay Alice and Bob from communicating? In fact, the trials and tribulations of Alice and Bob capture the fundamental difficulty shared by several n-player problems, including reliable broadcast, leader election, static k-selection, and t-resilient consensus. We provide round complexity lower bounds—and (nearly) tight upper bounds—for each of those problems. These results imply bounds on adversarial efficiency, which we analyze in terms of *jamming gain* and *disruption-free complexity.*

1 Introduction

Ad hoc networks of wireless devices hold significant promise for the future of ubiquitous computing. Unfortunately, such networks are particularly vulnerable to adversarial interference due to their use of a shared, public communication medium and their deployment in unprotected environments. For example, a committed adversary can disrupt an ad hoc network by jamming the communication channel with noise. Continuous jamming, however, might be unwise for the adversary: it depletes the adversary's energy, allows the honest devices to detect its presence, and simplifies its localization—and subsequent destruction. The adversary, therefore, would rather be more efficient, disrupting the protocol using a minimal number of transmissions.

Jamming Gain. The efficiency of the adversary can be quantified, roughly speaking, by comparing the *duration* of the disruption to the adversary's *cost* for causing the disruption. In the systems literature, this metric has been informally referred to as *jamming gain* (e.g., [1]). In the context of round-based protocols (time-slotted wireless radio channels), the jamming gain can be defined as follows. Let $D_P(t)$ be the minimal number of broadcasts needed by the adversary to delay protocol P from terminating for t rounds, for some initial value. Then the *jamming gain* of protocol P is: $\mathsf{JG}(P) = \lim_{T\to\infty} \frac{T}{\max(D_P(T),1)}$. For example, if the adversary must broadcast in every round, the jamming gain is 1. By contrast, if the adversary need *never* broadcast to prevent termination, then the jamming gain is infinite.

Disruption-Free Complexity. A second metric, *disruption-free complexity*, measures how long the adversary can disrupt a protocol without performing *even one* broadcast. The uncertainty introduced by the possibility of adversarial broadcasts is sufficient to slow down many protocols. This is defined as: $\mathsf{DF}(P) = \max\{t : D_P(t) = 0\}$. If a protocol has large disruption-free complexity, then the adversary can significantly reduce the throughput of multiple consecutive executions, while avoiding the disadvantages of actually jamming.

This paper is the first theoretical examination of the efficiency of malicious disruption in a wireless ad hoc network. We begin by analyzing a 3-player game that captures many of the fundamental difficulties of wireless coordination. We then extend these results to several classical problems: reliable broadcast, leader election, static k-selection and consensus. For each problem, we present fundamental limits on the robustness to malicious interference, and we present algorithms that match these standards of robustness.

The 3-Player Game. The 3-player game consists of two honest players—Alice and Bob, and a third malicious player, Collin (the *Collider*). All three players share a time-slotted single-hop wireless radio channel. Alice and Bob each begin with a value to communicate. Colin is determined to prevent them from communicating, in either direction, for as long as possible. Collin can broadcast in any time slot (i.e., round), potentially destroying honest messages or overwhelming them with malicious data. In order to precisely measure the efficiency of a malicious adversary, we endow Collin with a budget of β messages, and analyze how long Alice and Bob can be disrupted. The size of β is not known *a priori* to Alice and Bob. (If it were, then Alice and Bob could communicate reliably by repeating each message $2\beta + 1$ times.)

3-Player Lower Bound. We show that Collin can delay Alice and Bob's communication for $2\beta + \lg|V|/2$ rounds, where V is the set of possible values that Alice and Bob may communicate. An immediate corollary is that no protocol for Alice and Bob can achieve a jamming gain better than 2. This result is surprising as it implies that every protocol has some semantic vulnerability that the adversary can exploit to gain extra efficiency. A second corollary is that the disruption-free complexity is $\Theta(\lg|V|)$. Therefore for large V, the passive presence of Collin can significantly reduce Alice and Bob's communication throughput. We prove these lower bounds (in Section 4) by exhibiting a strategy for Collin to delay Alice and Bob, exploiting the fact that they can never trust any message, since Collin could have *overwhelmed* it with a fake message.

3-Player Upper Bound. For our upper bound (Section 5), we consider a (harder) setting where Alice needs to transmit a value to Bob, who does not broadcast any messages. We exhibit a protocol that allows Alice—using $\beta + \Delta$ broadcasts—to transmit her value to Bob in $2\beta + \max\{2\Delta 2^{\frac{\lg|V|}{\Delta}}, 4\lg|V|\}$ rounds. (Notice that if $\Delta < 1$, we show that Alice's task is impossible.) For $\Delta = \Omega(\lg|V|)$, the protocol matches our lower bound. For $\Delta < \lg|V|$, however, Collin can delay the communication more efficiently. For example, if $\Delta = 1$, the disruption-free complexity of our protocol increases to $|V|$. We show that a disruption-free complexity of $\max\{2\Delta 2^{\frac{\lg|V|}{\Delta}}, 4\lg|V|\}$ is unavoidable, highlighting an inherent tradeoff between Alice's message complexity and her throughput. Finally, we consider a variant of the 3-player game in which Alice and Bob do not start in the same round; Bob is activated asynchronously by the adversary. We present a protocol that solves this problem and still terminates within $2\beta + \Theta(\lg|V|)$ rounds (assuming Alice has an unrestricted message budget).

The n-Player Implications. The trials and tribulations of Alice and Bob capture something fundamental about how efficiently malicious devices can disrupt wireless coordination. In Section 6, we derive new lower bounds—via reduction to our 3-player game—for several classical n-player problems: reliable broadcast: $2\beta + \Theta(\lg|V|)$; leader election: $2\beta + \Omega(\log\frac{n}{k})$; static k-selection: $2\beta + \Omega(k\lg\frac{|V|}{k})$. For the latter two cases, k represents the number of participants contending to become leader and to transmit their initial value, respectively. As before, we draw immediate corollaries regarding the jamming gain and disruption-free complexity, resulting in a jamming gain of 2, and disruption-free complexity of $\Theta(\lg|V|)$, $\Omega(\log\frac{n}{k})$, and $\Omega(k\lg\frac{|V|}{k})$, respectively.

We next consider a more general framework that also includes crash failures: the malicious adversary can both broadcast β messages *and* also crash up to t honest devices. We study binary consensus as an archetypal problem in this framework, and derive a lower bound of $2\beta + \Theta(t)$ rounds. The $\Theta(t)$ factor is established by a technique that maintains the indistinguishability of two univalent configurations for t rounds. The 2β factor then follows from a (partial) reduction to consensus. This shows a jamming gain of 2, as before. By contrast, the disruption-free complexity, $\Theta(t)$, is significantly larger than for the crash-free models. (Notice that if the adversary is benign, then crash failures have no effect on the complexity.)

Finally, in Section 7, we present tight upper bounds for reliable broadcast and consensus and nearly tight bounds for leader election and static k-selection.

Assumptions and interpretations. Underlying our results on jamming gain and disruption-free complexity is an analysis of how long the adversary can disrupt communication given a limited broadcast budget. This interpretation is interesting in its own right: a limited broadcast budget models the (limited) energy available to a set of malicious devices.

Clearly, authentication—for example, using cryptographic keys—impacts our lower bounds. With authentication, the 3-player communication game completes in $\beta + 1$ rounds, resulting in a jamming gain and disruption-free complexity of 1. Intuitively, a jamming gain arises from semantic vulnerabilities in the protocol;

cryptographic techniques can eliminate this vulnerability. In general, however, deploying cryptographic solutions in wireless networks can be difficult. Public key authentication schemes are often expensive both in computation and, to some extent, communication. Symmetric key schemes (such as MACs) have been deployed in wireless networks (see, e.g., [2,3]), yet the focus has generally been link-level security, rather than authenticated broadcast, and there remain issues with key distribution. For example, if only a single key is used, the system is easily compromised by a single corrupted node; if multiple keys are used, then keys must be exchanged and communication is complicated. One interpretation of our bound is that authentication should be deployed only if its cost is less than the cost of waiting an additional $\beta + \Theta(\lg |V|)$ rounds.

2 Related Work

This paper explores the damage that can be caused by a genuinely malicious (Byzantine) device that can reliably disrupt communication in a wireless ad hoc network. Koo [4], Bhandari and Vaidya [5], as well as Pelc and Peleg [6], study "t-locally bounded" Byzantine failures in wireless networks, in which the number of Byzantine nodes in a region is bounded. In these papers, the Byzantine devices are required to follow a strict TDMA schedule, thus preventing them from interfering with honest communication. Others have considered models with probabilistic message corruption [7,8]. Wireless networks with crash failures (but not Byzantine failures) have also been studied extensively in both single hop (e.g., [9,10]) and multihop (e.g., [11,12]) contexts. By contrast, we consider a malicious adversary that can choose to send a message in any round, potentially destroying honest messages or overwhelming them with malicious data.

Simultaneous to this work, Koo, Bhandari, Katz, and Vaidya [13] have also considered a model where the adversary has a limited broadcast budget and can send a message in any round, overwhelming honest messages. A key difference, however, is that they assume that the adversary's budget is fixed *a priori* and known to all participants. By contrast, we do not assume that β is known in advance. (Thus it is no longer sufficient to repeat each message $2\beta + 1$ times.) Moreover, they focus primarily on feasibility, that is, determining the threshold density of dishonest players for which multihop broadcast is possible. By contrast, our paper focuses on the time complexity of the protocols and the efficiency of the adversary. Furthermore, we also consider the impact of combining crash failures with a malicious adversary.

Adversarial jamming of physical layer radio communication is a well studied problem in the electrical engineering community (see, e.g., [14]). In the context of wireless ad hoc networks, there has been recent interest in studying the jamming problem at the MAC layer and above. See, for example, [1,15,16,17], which analyze specific MAC and network layer protocols, highlighting semantic vulnerabilities that can be leveraged to gain increased jamming efficiency.

3 Preliminaries

We assume a synchronous round-based Multiple Access Channel (MAC) model with collision detection (as in, e.g. [18, 19, 20]). We consider n honest devices, *the players*, named from the set $[1, n]$, and one additional malicious device incarnating *the adversary.* In each round, each device can decide to broadcast a message or listen. If there are no broadcasts in a round, then none of the players receive a message. If exactly one message is broadcast, then all players receive the message. If two or more messages are broadcast, then each player can either: (1) receive exactly one of the broadcast messages; or (2) detect noise on the channel, i.e., a collision. (This channel behavior represents the unpredictability of real networks, for example, shadowing effects [21].) Without loss of generality, we assume that the adversary determines for each honest player whether option 1 or 2 occurs; in case of option 1 the adversary's message is systematically received.

Throughout this paper, we endow the adversary with a budget of β broadcast messages, where β is *a priori* unknown to the players. Also, we assume no message authentication capabilities. That is, a player cannot necessarily distinguish a message sent from the adversary from a message sent by a fellow honest player.

The basic game we consider involves two honest players, Alice and Bob, and an adversary, Collin. Alice is initialized with value $v_a \in V$ and Bob with $v_b \in V$, where $|V| > 1$ and V is known to all. The players can $\mathsf{output}(v)$ for any $v \in V$ such that the following are satisfied. **Safety:** Bob only outputs v_a and Alice only outputs v_b; and **Liveness:** Eventually, either Alice or Bob outputs a value.

4 Lower Bound for the 3-Player Game

We prove in this section a lower bound on the round complexity of the 3-player communication game. Our lower bound holds even if Alice and Bob have an unlimited budget of messages.

To prove our lower bound, we describe a strategy for Collin to frugally use his β messages to prevent communication. Two assumptions are key to this strategy: (1) Collin's budget of messages β is unknown to Alice and Bob; (2) Alice and Bob cannot distinguish a message sent by Collin from an honest message. A *silent* round, on the other hand, cannot be faked: if Bob (for example) receives no message and no collision notification, then he can be certain that Alice did not broadcast a message. Therefore, in order to prevent Alice and Bob from communicating, it is sufficient, roughly speaking, for Collin to disturb silence.

Theorem 1. *Any* 3*-player communication protocol requires at least* $2\beta + \lg|V|/2$ *rounds to terminate.*

Assume, for contradiction, a protocol, A, that defies this worst-case performance. Consider any value $v \in V$ and denote by $\gamma(v)$ the $\lg|V|/2 - 1$ round (good) execution of A where Alice and Bob begin with initial value v, and Collin performs no broadcasts. If Alice and Bob both broadcast in the same round, assume both messages are lost. We begin with the following lemma:

	Alice		**Bob**		**Collin**	
	$\alpha(v)$	$\alpha(w)$	$\alpha(v)$	$\alpha(w)$	$\alpha(v)$	$\alpha(w)$
1	m	-	-	-	-	m
2	-	m	-	-	m	-
3	-	-	m	-	-	m
4	-	-	-	m	m	-
5	m	-	m'	-	-	m'
6	-	m	-	m'	m'	-
7	m	-	-	m'	-	m
8	-	m	m'	-	m	-

(a) α Rules

	Alice		**Bob**		**Collin**
	$\alpha(v)$	$\rho(w,v)$	$\alpha(w)$	$\rho(w,v)$	$\rho(w,v)$
1	m	m'	-	-	m
2	-	-	m'	m	m'
3	m	-	-	-	m
4	-	-	m	-	m
5	m	-	m'	-	m
6	-	m	m'	-	m'
7	m	m'	m''	-	m
8	m	-	m''	m'	m''

(b) $\rho(w,v)$ Rules

Fig. 1. Collin's behavioral rules for α and $\rho(w,v)$ executions

Lemma 1. *There exist two values $v, w \in V$ $(v \neq w)$, such that Alice (resp. Bob) broadcasts in round r of $\gamma(v)$ if and only if Alice (resp. Bob) broadcasts in round r of $\gamma(w)$.*

Proof. In each round, there are four possibilities: (1) Alice broadcasts alone, (2) Bob broadcasts alone, (3) Alice and Bob both broadcast, and (4) neither Alice nor Bob broadcasts. Accordingly, for a sequence of c rounds, there are 4^c possible patterns of broadcast behavior. Thus, there are at most $4^{\lg|V|/2-1} = \frac{|V|}{4}$ possible broadcast patterns that result from the $|V|$ possible γ executions. It follows by the pigeonhole principle that at least two such executions have the same pattern.

For the rest of this proof, we fix v and w to be the two values identified by Lemma 1. We define $\alpha(v)$ (resp. $\alpha(w)$) to be the execution of A in which Alice and Bob both begin with initial value v (resp. w) and Collin applies the α-rules described in Figure 1(a). In this table, "-" indicates silence and m and m' both represent a message broadcast. Each row matches a specific set of broadcast behaviors of Alice and Bob in two executions, with the corresponding broadcast behavior followed by Collin in these executions. Since Alice and Bob's algorithm is deterministic, Collin can predict their behavior in each round.

For example, Rule 1 from Figure 1(a) specifies that for any given round, if Alice broadcasts in exactly one α execution, and Bob is silent in both, then Collin replicates Alice's broadcast in the execution where she is silent. For any pattern of broadcast behavior not described in the table, assume that Collin performs no broadcasts. Also, assume that in any round where both Collin and Alice (resp. Bob) broadcast, only Collin's message is received by Bob (resp. Alice). We claim:

Lemma 2. *Neither Alice nor Bob can output during $\alpha(v)$ or $\alpha(w)$.*

Proof. We show that Bob cannot output in $\alpha(v)$ and Alice cannot output in $\alpha(w)$. The argument for Bob in $\alpha(w)$ and Alice in $\alpha(v)$ is symmetric.

We first define a third execution $\rho(w,v)$, of A, in which Alice starts with initial value w and Bob starts with initial value v. The behavior of Collin in execution ρ is defined by the rules in Figure 1(b). Notice that, in all three executions, we assume that Collin has an unlimited broadcast budget. This is without loss of generality because Alice and Bob do not know the value of β, and we will later concern ourselves only with the prefixes of the α executions in which Bob has not yet broadcast more then β times.

We show, by induction on the round number, r, that $\rho(w, v)$ is indistinguishable from $\alpha(v)$ with respect to Bob, and that $\rho(w, v)$ is indistinguishable from $\alpha(w)$ with respect to Alice. The lemma follows immediately from this indistinguishability and the safety requirement of the communication game.

Since Bob begins with value v in both $\rho(w, v)$ and $\alpha(v)$, and Alice begins with value w in both $\rho(w, v)$ and $\alpha(w)$, the base case ($r = 0$) is immediate. We now consider the possible behaviors of Alice and Bob during round $r + 1$.

– **Case 1:** *Alice broadcasts in* $\alpha(w)$. By induction, this implies Alice also broadcasts in $\rho(w, v)$, therefore these two executions remain indistinguishable with respect to Alice (as broadcasters cannot listen). We turn our attention to Bob, bypassing the sub-case of Bob broadcasting in $\alpha(v)$ as this is Case 2. Two sub-cases: (1) Alice is silent in $\alpha(v)$. If Bob is also silent in $\alpha(w)$, then, by α-Rule 2, Collin broadcasts Alice's $\alpha(w)$ (and $\rho(w, v)$) message in $\alpha(v)$. If Bob broadcasts in $\alpha(w)$, then, by α-rule 6, Collin broadcasts Bob's message in $\alpha(v)$ and, by ρ-rule 6, Collin also broadcasts Bob's message in $\rho(w, v)$. (2) Alice broadcasts in $\alpha(v)$. By ρ-Rule 1 or 7 (depending on whether Bob broadcasts in $\alpha(w)$) Collin replicates Alice's $\alpha(v)$ message in the ρ execution. In all cases, Bob receives the same message in $\alpha(v)$ and $\rho(w, v)$.
– **Case 2:** *Bob broadcasts in* $\alpha(v)$. This argument is symmetric to Case 1.
– **Case 3:** *Alice does not broadcast in* $\alpha(w)$ *and Bob does not broadcast in* $\alpha(v)$. There are four sub-cases. (1) Alice and Bob don't broadcast in $\alpha(v)$ and $\alpha(w)$, respectively. Collin does nothing and the executions are clearly indistinguishable. (2) Alice broadcasts in $\alpha(v)$ and Bob is silent in $\alpha(w)$. By α-rule 1, Collin broadcasts Alice's message in $\alpha(w)$. By ρ-rule 3, Collin broadcasts Alice's message in $\rho(w, v)$. Therefore, Bob receives Alice's message in $\alpha(v)$ and $\rho(w, v)$, and Alice receives her message (from Collin) in $\alpha(w)$ and $\rho(w, v)$. (3) Alice is silent in $\alpha(v)$ and Bob broadcasts in $\alpha(w)$. By α-rule 4, Collin broadcasts Bob's message in $\alpha(v)$. By ρ-rule 4, Collin broadcasts Bob's message in $\rho(w, v)$. Therefore, Alice receives Bob's message in $\alpha(w)$ and $\rho(w, v)$, and Bob receive his message (from Collin) in $\alpha(v)$ and $\rho(w, v)$. (4) Alice broadcasts in $\alpha(v)$ and Bob broadcasts in $\alpha(w)$. By α-rule 7, Collin broadcasts Alice's message in $\alpha(w)$. By ρ-rule 5, Collin broadcasts Alice's message in $\rho(w, v)$. Therefore, Alice receives her message (from Collin) in $\alpha(w)$ and $\rho(w, v)$, and Bob receives Alice's message in $\alpha(v)$ and $\rho(w, v)$.

We now show that one of these two indistinguishable α executions requires only β broadcasts by Collin during the first $2\beta + \lg |V|/2 - 1$ rounds.

Proof (Theorem 1). By Lemma 2, Alice and Bob do not produce an output in either $\alpha(v)$ or $\alpha(w)$ as long as Collin continues to follow the α rules. It suffices to show that, in at least one of the two executions $\alpha(v)$ and $\alpha(w)$, Collin broadcasts in no more than β of the first $2\beta + \lg |V|/2 - 1$ rounds.

Algorithm 1: Bit Broadcast Sub-Protocol

```
bcast-Alice($b$)
  $active \leftarrow$ true
  while ($active$) do
    if ($b$=1) then
      bcast($vote$)    ▷ Data round broadcast
    $m \leftarrow$ recv()    ▷ Data round receive
    if ($b$=0) and ($m \neq \perp$) then
      $bcast(veto)$    ▷ Veto round broadcast
    $m \leftarrow$ recv()    ▷ Veto round receive
    if ($m = \perp$) then
      $active \leftarrow$ false
  return
```

```
recv-Bob()
  $active \leftarrow$ true
  while ($active$) do
    $votes \leftarrow$ recv()    ▷ Data round receive
    $vetos \leftarrow$ recv()    ▷ Veto round receive
    if ($vetos = \perp$) then
      $active \leftarrow$ false
  if ($votes = \perp$) then
    return 0
  else
    return 1
```

Algorithm 2: Sequence Broadcast Protocol

```
SEQ-Alice($s \in \{0,1\}^k$,$k$)
  $count \leftarrow 1$
  while ($count \leq k$) do
    bcast-Alice($s[count]$)
    $count \leftarrow count + 1$
```

```
SEQ-Bob($k$)
  $count \leftarrow 1$
  while ($count \leq k$) do
    $s[count] \leftarrow$ recv-Bob()
    $count \leftarrow count + 1$
  output($s$)
```

We first consider rounds 1 through $\lg|V|/2 - 1$ of $\alpha(v)$ and $\alpha(w)$. We know by Lemma 1 that Alice and Bob broadcast on the same schedule for these initial rounds when they start both with v or both with w. Notice, however, that Collin broadcasts (according to the α rules) only in situations of asymmetric silence, i.e. when Alice and Bob are *not* on the same schedule. It follows that Collin does not broadcast in either $\alpha(v)$ or $\alpha(w)$ for the first $\lg|V|/2 - 1$ rounds.

Now we turn our attention to the 2β rounds that follow. For a given round, Collin only broadcasts in $\alpha(v)$ or $\alpha(w)$, but not both, since he only fills in asymmetric silent rounds. Therefore, by a simple counting argument, it is impossible for Collin to broadcast in more than half of the rounds in both executions. We therefore choose the execution in which Collin broadcasts in no more than half of the following 2β rounds. This delays both Alice and Bob from outputting for $2\beta + \lg|V|/2 - 1$ rounds. □

We conclude with an immediate corollary of Theorem 1:

Corollary 1. *Any 3-player communication protocol has a jamming gain of at least 2, and a disruption-free complexity of $\Omega(\lg|V|)$.* □

5 Upper Bounds for the 3-Player Game

We prove in this section that our (round complexity) lower bound is tight, by showing that there is a protocol that matches it. To strengthen our upper bound result, we consider the seemingly harder problem of Alice transmitting her value to Bob in a setting where Bob does not broadcast. Specifically, we give a protocol that, assuming Alice has a budget of $\beta + \Delta$ messages, transmits Alice's input value to Bob in $2\beta + \max\{2\Delta 2^{\frac{\lg|V|}{\Delta}}, 4\lg|V|\}$ rounds. For $\Delta = \Omega(\lg|V|)$, this protocol matches our lower bound. For $\Delta = o(\lg|V|)$ the round complexity grows. We show this to be unavoidable.

Our protocol broadcasts a sequence of bits (Algorithm 2), using a sub-protocol for each bit. Alice to Bob (Algorithm 1). The basic idea of Algorithm 1 is to alternate data rounds and veto rounds. In a data round, Alice transmits a message if $b = 1$ and remains silent otherwise. If Collin interferes with the data round (i.e. by broadcasting in the case where $b = 0$), Alice indicates this interference by broadcasting in the veto round. At this point, Alice and Bob try again with a new pair of rounds. Of course, Collin can also interfere by broadcasting in a veto round. This too causes Alice and Bob to try again with a new pair of rounds. The sub-protocol continues until the first silent veto round.

Alice and Bob both know that Alice has a broadcast budget of $\beta+\Delta$. Typically, Alice would broadcast a binary encoding of her value, which might require $\lg|V|$ broadcasts. If $\Delta < \lg|V|$, we encode the value as bit strings of length k containing at most Δ 1's. We choose k to be the minimum value such that $\binom{k}{\Delta} \geq |V|$, that is, the smallest value that allows us to express all V values. Alice then transmits this encoding as described above. This is summarized in the following theorem, whose proof is in the full version:

Theorem 2. *There exists a protocol through which Alice transmits her initial value to Bob, within* $2\beta+\max\{2\Delta 2^{\frac{\lg|V|}{\Delta}}, 4\lg|V|\}$ *rounds, using a budget of* $\beta+\Delta$ *messages. This protocol thus has a jamming gain of 2, and a disruption-free complexity of* $\max\{2\Delta 2^{\frac{\lg|V|}{\Delta}}, 4\lg|V|\}$

Notice that Theorem 2 assumes Alice has a message budget that is strictly larger than Collin's budget (as indicated by the constraint $\Delta > 0$). This is in fact necessary, and it is impossible to communicate a value from Alice to Bob if Alice's budget is $\leq \beta$ since, in this case, Collin can successfully simulate Alice's behavior (see the full version). Notice that the round complexity grows significantly as Δ decreases below $\lg|V|$. We show this trade-off to be inherent:

Theorem 3. *Let* $k = \max\{\frac{\Delta 2^{\lg(|V|)/\Delta}}{e} - \Delta, \frac{lg|V|}{2}\}$*. If Alice has a budget of size* $\beta + \Delta$ *(*$\Delta > 0$*), then there exists no protocol through which Alice can transmit her initial value to Bob in less than* $2\beta + k$ *rounds. Thus every such protocol has a disruption-free running time of* $\Omega(k)$*.*

The Wake-Up Case. We have assumed that Alice and Bob begin in the same round. Consider the case where Bob is activated at an unknown point in the execution. Thus, Bob no longer has round numbers synchronized with Alice. This models the situation where Alice represents a base station that needs to transmit a value to intermittently awake tiny devices (i.e., Bob). There is (in the full version) a variant of our protocol that solves the problem in $2\beta+\Theta(\lg|V|)$ rounds after Bob awakes, asymptotically matching our lower bound from Section 4, despite the extra synchronization challenges. This variant requires Alice to never terminate, which is inevitable given that she can never distinguish between Bob and Collin pretending to be Bob (while Bob is still sleeping).

6 Lower Bounds for n-Player Problems

We generalize here our results to n-player coordination problems. We then consider the impact of combining malicious behavior with crash failures.

6.1 n-Player Reductions

We show here how Alice and Bob can together simulate an arbitrary n-player protocol. We then use this simulation to derive lower bounds, via reduction from the 3-player communication game, for several n-player problems. None of our round-complexity lower bounds restricts the message budget of honest players.

A simulation by Alice and Bob is defined by a 5-tuple: $\{A, n, S_A, S_B, I\}$, where: (1) A is the n-player protocol being simulated; (2) S_A and S_B partition the n players into two non-empty and non-overlapping sets; (3) I is a mapping of players to their respective initial values.

Alice simulates the players in S_A, initializing them according to I. (Alice is provided only the initial values for nodes in S_A, i.e., $I|S_A$.) In each round, if any of the players in S_A choose to broadcast, Alice arbitrarily chooses one of their messages to broadcast. She then delivers to each simulated player any messages or collision notifications from that round. Bob simulates the players in S_B in an equivalent manner. The following can be proved by straightforward induction:

Theorem 4. *Consider simulation $\{A, n, S_A, S_B, I\}$. For all r-round executions of the simulation, there exists an r-round execution α of A, initialized according to I, where the outputs of Alice and Bob are equivalent to the outputs in α, and Collin broadcasts the same number of messages in the simulation and α.*

Reliable Broadcast. In reliable broadcast, one player—the *source*—is provided with an input value $v_0 \in V$. Each player must receive this initial value. *Safety* requires that each player output only v_0, i.e., perform $\mathsf{output}(v)$ only if $v = v_0$. *Liveness* requires that all players eventually perform an output.

Theorem 5. *Any reliable broadcast protocol requires at least* $2\beta + \lg |V|/2$ *rounds to terminate.*

Proof. Assume by contradiction that A is a reliable broadcast protocol that terminates in $R < 2\beta + \lg |V|/2$ rounds for all initial values. We reduce 3-player communication, for value domain V, to A. Alice and Bob simulate A for n players, where: (1) S_A contains the source, S_B contains all other players, and (2) I maps the source to v_a, Alice's initial value. Bob outputs the first value output by a simulated player. By Theorem 4, Bob always outputs $v_0 = v_a$ by round R, contradicting Theorem 1.

Leader Election. In leader election, $k \leq n$ *participants* contend to become the leader. All n players should learn the leader, i.e., perform $\mathsf{output}(\ell)$, for some ℓ. *Safety* requires that the leader be a participant, and that there be only one leader. *Liveness* requires every player to perform an output.

Theorem 6. *Any leader election protocol requires at least* $2\beta + \lg \lfloor \frac{n-1}{k} \rfloor / 2$ *rounds to terminate.*

Proof. Assume by contradiction that A is a leader election protocol that terminates in $R < 2\beta + \lg \lfloor \frac{n-1}{k} \rfloor / 2$ rounds for all choices of k participants. We reduce to leader election, the 3-player game defined over the value space V, where V contains every integer between 1 and $\lfloor \frac{n-1}{k} \rfloor$, to A.

Alice and Bob simulate A for n players where: (1) S_A contains players 1 through $n-1$, S_B contains player n, and (2) I activates player $i \in S_A$ if and only if $(i \mod \lfloor \frac{n-1}{k} \rfloor) + 1 = v_a$ and fewer than k nodes have been activated so far in I_A. Let i be the leader output by Bob's simulated player. Bob outputs $v_a = (i \mod \lfloor \frac{n-1}{k} \rfloor) + 1$, as required. By Theorem 4 Bob always outputs v_a within R rounds, contradicting Theorem 1, since $2\beta + \lg V/2 = 2\beta + \lg \lfloor \frac{n-1}{k} \rfloor /2$.

Static k-Selection. In static k-Selection, k *participants* are provided with values $v_i \in V$. Each player must receive all values. *Safety* requires that the first k outputs of a player equal the k values. *Liveness* requires that all players eventually perform at least k output actions. The protocol *terminates* when all players have performed at least k output actions. (The selection problem is well-studied in radio networks, e.g., [22, 23].)[1]

Theorem 7. *Any static k-selection protocol requires at least* $2\beta + \Omega(k \lg \frac{|V|}{k})$ *rounds to terminate.*

Proof. Assume by contradiction that A is a protocol that terminates in $R < 2\beta + o(k \lg |V|/k)$ rounds, for all initial values and choices of participants. We reduce to k-selection, the 3-player game for the value space V', where V' contains one entry for every multiset of k values drawn from V, to A.

Alice and Bob simulate A for n players where: (1) S_A contains players 1 through k, S_B contains the remaining players, and (2) I activates players 1 through k, and provides each a different value from the multiset described by $v_a \in V'$. Given k simulated outputs, Bob can reconstruct and output the unique multiset described by these values. By Theorem 4 Bob will always output v_a in R rounds, contradicting Theorem 1, since $2\beta + \lg |V'|/2 = 2\beta + \lg \frac{|V|^k}{k!}/2 = 2\beta + \Theta(k \lg \frac{|V|}{k})$ rounds.

Corollary 2. *Any protocol for reliable broadcast, leader election or static k-selection has a jamming gain of at least 2 and a disruption-free running time of* $\Omega(\log |V|)$, $\Omega(\lg \frac{n-1}{k})$, *and* $\Omega(k \lg \frac{|V|}{k})$, *respectively.* □

6.2 Combining Malicious and Crash Behavior

We now study the impact of combining malicious behavior with crash failures. We assume that the adversary, in addition to having a budget of β messages, can also crash up to t players. We consider the problem of *binary consensus.* The n honest players each propose a value. *Liveness* requires that all non-crashed players eventually decide a value. *Agreement* requires all players that decide to choose the same value. *Validity* requires that if all non-crashed players propose the same value, then all deciding players choose that value.

[1] Often k-selection is oblivious to initial values. We allow a dependence on the initial values, strengthening the lower bound.

By a simple indistinguishability argument, it is easy to see that consensus is impossible if $n \leq 2t$: it is impossible to distinguish a correct player from a crashed player that is simulated by the adversary; thus no player can decide in an execution in which t players propose '0' and t propose '1'.

We therefore assume that $n = 2t+1$, and establish a lower bound of $2\beta + \Theta(t)$ on the round complexity of consensus. Our bound reveals the interesting fact that the possibility of crashed honest devices increases the power of the malicious adversary. This is perhaps surprising as, if there is no malicious adversary, crash-failures have no effect on termination (in a synchronous broadcast network).

As before, we use a simulation by Alice and Bob of the (t-resilient) n-player consensus protocol. The simulation, however, is more challenging than those used for the n-player problems studied previously as we must compensate for the crash failures. We do not start the simulation from the initial configuration, but instead from one of two univalent configurations arising after t rounds. These configurations are constructed in Lemma 3, which is interesting in its own right as it exhibits executions in which information (about initial values) is transmitted at most one bit per round. By combining it with valency arguments, we show how the 3-player game can aid the construction of involved lower bounds.

Theorem 8. *Any t-resilient binary consensus protocol requires at least $2\beta + t$ rounds to terminate.*

We fix the environment such that if multiple messages are sent in a round, and the adversary does not broadcast, then the message sent by the player with the smallest id is received by everyone. An execution (or prefix) is *failure-free* if it includes no crashes or broadcasts by the adversary.

Given these assumptions, it is clear that each initial configuration results in a deterministic failure-free execution. We represent all of these possible failure-free executions as a tree T. Every execution begins at the root, and a node at depth r represents the execution at the beginning of round r. Each node at depth r contains one outgoing edge for every possible message m that may be received in round r, and one outgoing edge for a silent round (labeled $\perp$). Thus, every failure-free execution of A is represented by a single path in T. Accordingly, for each initial configuration c, we say that a node $x \in T$ is *reachable* from c—with respect to A—if the path associated with c's failure-free execution includes node x. We define the tree $T(A)$ to be T pruned to contain only reachable nodes. That is, if $x \in T(A)$, then there exists some initial configuration c for which x is reachable. Notice that if a depth r node x is reachable for two initial configurations c and c', and some player i has the same initial value in c and c', then at the beginning of round r, player i cannot distinguish c from c'. If c is 0-valent, and c' is 1-valent, then i cannot decide prior to round r.

Lemma 3. *There exists a path of length t in $T(A)$, ending at node R_t, where R_t is reachable from two initial configurations, c_0 and c_1, such that some player p_t has the same initial value in c_0 and c_1, and every crash-free extension of c_0 is 0-valent and every crash-free extension of c_1 is 1-valent, with respect to A.*

Proof. Starting at the root of $T(A)$, given an initial configuration c_0, construct a path of length t by applying the following: (1) If there exists ≥ 1 outgoing message edges, choose the message from the player with the smallest id. (2) Otherwise, follow the $\perp$ edge. Let R_t be the node reached after t iterations.

Configuration c_0 contains either a majority of '0's or a majority of '1's. Notice that a majority contains at least $t+1$ players, since $n = 2t+1$. Assume without loss of generality that a majority of players (i.e., at least $t+1$) propose '0' in c_0. This implies that any crash-free extension of c_0 must decide '0', since any such execution is indistinguishable from one in which all players propose '0', and those $\leq t$ players proposing '1' are crashed nodes emulated by the adversary—in which case a decision of '1' violates validity.

We now construct an initial configuration c_1. Denote by P the set of players that broadcast messages which were received along the path to R_t. Note that P contains $\leq t$ players. Choose c_1 such that the players in P propose the same initial value as in c_0, and the remaining players (at least $t+1$) all propose '1'. Choose some $p_t \in P$. (If $|P| = 0$, then arbitrarily choose one player p_t to have the same initial value in c_0 and c_1.) It is clear, by the same reasoning applied to c_0, that all crash-free extensions of c_1 must decide '1'. It follows that R_t is reachable from c_1, by a straightforward induction argument.

Proof (Theorem 8). Let α_0 (resp. α_1) denote the failure-free execution prefix starting from c_0 (resp. c_1) and ending at R_t. Executions α_0 and α_1 are indistinguishable with respect to p_t; hence p_t has not decided prior to round t. To this point, the adversary has used zero broadcasts. To achieve a further 2β delay, we defer to Alice and Bob, who can solve the binary communication game by performing a *crash-free* simulation of the n-player protocol, in which Alice begins in the final state of α_0 or α_1, and Bob simulates p_t. This simulation cannot terminate in fewer than 2β round, implying the desired bound. □

Corollary 3. *Any t-resilient binary consensus protocol has a jamming gain of at least 2 and a disruption-free complexity of $\Omega(t)$.* □

7 Upper Bounds for the n-Player Problems

We now present protocols for reliable broadcast, leader election, static k-selection, and binary consensus. Our reliable broadcast and consensus protocols match the lower bounds. Those for leader election and k-selection leave a gap.

Reliable Broadcast. An algorithm for reliable broadcast follows from the algorithm in Section 5. The source runs Alice's protocol, and all other players run Bob's protocol, resulting in a running time of $2\beta + O(\lg |V|)$, matching the lower bound. This protocol requires the source to have a budget of $\beta + \lg |V|$.

Binary Consensus. Assuming t crashes, consensus can be achieved using reliable broadcast: each of $2t+1$ players transmits their initial value sequentially. (Notice that a crashed player, if there is no malicious interference, transmits a '0', according to the protocol.) Everyone decides the majority value. The running time is $2\beta + \Theta(t)$. Each player needs a budget of $\beta + 1$ broadcasts.

Leader Election. In order to elect a leader, we use a *tournament tree*, a binary tree with n leaves, each labeled with a player's id. Assume $c \geq 1$ is an integer parameter. The protocol begins at the root, and at each step descends to a child or ascends to the parent. At each step, the protocol determines whether there are any participants in the left or right subtrees. First, each participant in the left subtree broadcasts up to c times. If all of these rounds are non-silent, the protocol descends to the left subtree. Otherwise, the first time a silent round occurs, it skips the remaining rounds and checks the right subtree: each participant in the right subtree broadcasts up to c times. If all of these rounds are non-silent, the protocol descends to the right subtree. Otherwise, on the first silent round, the protocol ascends to the parent. On reaching a leaf, the identified node uses reliable broadcast to transmit a '1' if it is participating and a '0' otherwise. In the latter case, the protocol ascends to the parent and continues. Each participant needs a budget of $2c \lg n + \beta + 1$ broadcasts.

Theorem 9. *The leader election protocol terminates after* $2\beta\frac{c+1}{c} + 2c\lg n + 2$ *rounds, for all* $c \geq 1$. □

k-Selection. A protocol for static k-selection can be obtained by repeating the leader election protocol k times, each time using reliable broadcast to transmit the initial value. The protocol completes when leader election finds no further contenders. Each participant needs a budget of $2c\lg n + \beta + \log|V|$ broadcasts.

Theorem 10. *The* k*-selection protocol terminates in* $2\beta\frac{c+1}{c} + 2kc\lg n + k\lg V + 2k + 2$*, which equals* $2\beta\frac{c+1}{c} + O(ck\lg|V|)$ *if* $\lg n = O(\lg|V|)$*, for all* $c \geq 1$. □

8 Concluding Remarks

Interestingly, our lower bounds hold for weaker games. Lemmas 1 and 2 imply that calculating equality, *bitwise-and* or *bitwise-or* have the same round complexity as the 3-player game. We also conjecture that even for a randomized algorithm, $2\beta + \Theta(\lg|V|)$ rounds are needed. A future research direction is to extend our results to multihop environments.

Ackowledgments

We are grateful to H. Attiya and G. Chockler for their comments and discussions.

References

1. Brown, T.X., James, J.E., Sethi, A.: Jamming and sensing of encrypted wireless ad hoc networks. Technical Report CU-CS-1005-06, UC Boulder (2006)
2. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: Spins: Security protocols for sensor networks. Wireless Networks **8**(5) (2002) 521–534

3. Karlof, C., Sastry, N., Wagner, D.: Tinysec: A link layer security architecture for wireless sensor networks. In: Embedded Networked Sensor Systems. (2004)
4. Koo, C.Y.: Broadcast in radio networks tolerating byzantine adversarial behavior. In: Principles of Distributed Computing. (2004) 275–282
5. Bhandari, V., Vaidya, N.H.: On reliable broadcast in a radio network. In: Principles of Distributed Computing. (2005) 138–147
6. Pelc, A., Peleg, D.: Broadcasting with locally bounded byzantine faults. Information Processing Letters **93**(3) (2005) 109–115
7. Drabkin, V., Friedman, R., Segal, M.: Efficient byzantine broadcast in wireless ad hoc networks. In: Dependable Systems and Networks. (2005) 160–169
8. Pelc, A., Peleg, D.: Feasibility and complexity of broadcasting with random transmission failures. In: Principles of Distributed Computing. (2005) 334–341
9. Clementi, A., Monti, A., Silvestri, R.: Optimal f-reliable protocols for the do-all problem on single-hop wireless networks. In: Algorithms and Computation. (2002) 320–331
10. Chlebus, B.S., Kowalski, D.R., Lingas, A.: The do-all problem in broadcast networks. In: Principles of Distributed Computing. (2001) 117–127
11. Kranakis, E., Krizanc, D., Pelc, A.: Fault-tolerant broadcasting in radio networks. In: European Symposium on Algorithms. (1998) 283–294
12. Clementi, A., Monti, A., Silvestri, R.: Round robin is optimal for fault-tolerant broadcasting on wireless networks. J. Parallel Distributed Computing **64**(1) (2004) 89–96
13. Koo, C.Y., Bhandari, V., Katz, J., Vaidya, N.H.: Relibable broadcast in radio networks: The bounded collision case. In: Principles of Distributed Computing. (2006)
14. Stahlberg, M.: Radio jamming attacks against two popular mobile networks. In: Helsinki University of Technology Seminar on Network Security. (2000)
15. Negi, R., Perrig, A.: Jamming analysis of mac protocols. Technical report, Carnegie Mellon University (2003)
16. Hu, Y., Perrig, A.: A survey of secure wireless ad hoc routing. IEEE Security and Privacy Magazine **02**(3) (2004) 28–39
17. Gupta, V., Krishnamurthy, S., Faloutsos, S.: Denial of service attacks at the mac layer in wireless ad hoc networks. In: Military Communications Conference. (2002)
18. Abramson, N.: The aloha system - another approach for computer communications. Proceedings of Fall Joint Computer Conference, AFIPS **37** (1970) 281–285
19. Metcalf, R.M., Boggs, D.R.: Ethernet: Distributed packet switching for local computer networks. Communications of the ACM **19**(7) (1976) 395–404
20. Bar-Yehuda, R., Goldreich, O., Itai, A.: On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. Journal of Computer and System Sciences **45**(1) (1992) 104–126
21. Woo, A., Whitehouse, K., Jiang, F., Polastre, J., Culler, D.: Exploiting the capture effect for collision detection and recovery. In: Workshop on Embedded Networked Sensors. (2005) 45–52
22. Clementi, A., Monti, A., Silvestri, R.: Selective families, superimposed codes, and broadcasting on unknown radio networks. In: Symposium on Discrete algorithms, Philadelphia, PA, USA (2001) 709–718
23. Kowalski, D.R.: On selection problem in radio networks. In: Principles of Distributed Computing, New York, NY, USA, ACM Press (2005) 158–166