Interference-Resilient Information Exchange*

Seth Gilbert
École Polytechnique
Fédérale de Lausanne
1015 Lausanne
Switzerland

Rachid Guerraoui
École Polytechnique
Fédérale de Lausanne
1015 Lausanne
Switzerland

Dariusz R. Kowalski
U. of Liverpool
Liverpool L693BX
United Kingdom

Calvin Newport
Massachusetts Institute
of Technology
Cambridge, MA 02139
USA

***Abstract*—This paper presents an efficient protocol for reliably exchanging information in a single-hop, multi-channel radio network subject to unpredictable interference. We model the interference by an adversary that can simultaneously disrupt up to t of the C available channels. We assume no shared secret keys or third-party infrastructure. The running time of our protocol depends on the gap between C and t: when the number of channels $C = \Omega(t^2)$, the running time is linear; when only $C = t+1$ channels are available, the running time is exponential. We prove that exponential-time is unavoidable in the latter case.**

At the core of our protocol lies a combinatorial function, possibly of independent interest, described for the first time in this paper: the *multi-selector*. A multi-selector generates a sequence of channel assignments for each device such that every sufficiently large subset of devices is partitioned onto distinct channels by at least one of these assignments.

I. Introduction

We study the problem of reliable information exchange in a multi-channel single-hop radio network subject to unpredictable interference. Each device begins the execution with a value that it wants to distribute to everyone else; the goal is for as many devices as possible to learn as much information as possible.[1] This problem is at the core of many distributed applications, including: data aggregation in sensor networks, distributed data storage, fault-tolerant agreement, group membership, and mobile location services.

As practitioners readily admit, reliably exchanging information is challenging in the context of unreliable radio networks. This holds especially true for devices operating on the increasingly crowded unlicensed bands of the radio spectrum. In this setting, devices must tolerate unpredictable and perhaps even *adversarial* interference from sources as diverse as: the electromagnetic radiation of nearby microwaves; nearby devices running unrelated protocols; any combination of fading, multipath, or shadowing effects that can render communication unreliable; and actual malcontents armed with signal jammers. Shared secrets can be used to mitigate these problems via pseudo-random frequency hopping, as in Bluetooth [1], but the establishment of such secrets can be problematic in many settings. We seek solutions that do not assume such secrecy.

We model disruptive signals in the form of an adaptive adversary. We assume that it knows the protocol in advance, and hence it knows, in each round, which channels are used for communication. We assume that the adversary can disrupt up to t among the C channels at any given time. Note that this adversary is simply a useful modeling convention: it does not necessarily describe an actual malicious entity. It provides a powerful abstraction for modeling a diversity of different unpredictable sources of interference.

Our assumption that the adversary knows which channels are used can be read at least three different ways. First, this assumption captures the worst-case disruption that an adversary can achieve, even if it knows the protocol in advance. A protocol that tolerates such an adversary will work under *any* disruption patterns—whether they are adversarial or random, whether they are caused by a jamming device or by fading/multipath phenomena. A second interpretation of this assumption is that the source of disruption is a device that was formerly *honest*, but is suffering from faults. Such a faulty device is aware of any secrets shared by the non-faulty devices; any frequency hopping based on these secrets provides no security. Third, it may be possible that a malicious device can scan the C channels quickly to see which are in use, before choosing which channel to disrupt. (Jamming a channel, by contrast, requires focusing on a single channel due to the frequent use of error-correcting codes; thus a device seems onlikely to be able to rapidly jump between channels jamming them all.)[2]

Against such an adversary, reliable communication requires the simultaneous use of more channels than can be disrupted. Imagine that we identify a sequence of channel assignments that guarantees the following: for every subset of $t+1$ devices, there exists an assignment that assigns the $t+1$ devices to distinct channels. In this case, we know that at most t devices can be disrupted, as all groups of size $t+1$ have one round in which they use $t+1$ different channels, only t of which can be simultaneously disrupted. The paper shows how to solve this *simultaneous selection* problem using *multi-selectors* and

*This work was supported in part by the Engineering and Physical Sciences Research Council [grant number EP/G023018/1].

[1] Elsewhere, the problem of information exchange is occasionally referred to as *gossip*. The term *gossip* also refers to a specific randomized epidemic approach. Hence, to avoid confusion, we use the term *information exchange*.

[2] By contrast, if we assume the adversary cannot discover which channel is in use until after the transmission is complete, then there is a relatively simple randomized protocol with polynomial time complexity: the devices take round-robin turns broadcasting their data on a randomly chosen channel, while the remaining devices listen on a randomly chosen channel. If $C = t+1$ then within $O(C^2 \log n)$ time, with high probability, every device has received the data. See [2] for more on such a weak adversary.

generalized multi-selectors, two new combinatorial constructions that generalize *selectors*, classical tools for fault-free radio communication, (see [3], [4]). We show that there exist efficient multi-selectors and generalized multi-selectors and that, for certain important cases, these combinatorial objects are polynomial in length. Moreover, in these cases, we present a method for constructing polynomial length multi-selectors using hash functions. These new tools are at the core of our information exchange protocol.

In addition to avoiding adversarial disruption, we also make use of multi-selectors to *adaptively* prevent contention, that is, to determine a broadcast schedule dynamically as the execution proceeds. If the schedule induces too much contention, then the information dissemination is delayed by collisions and lost messages. (As was shown in [5], we need to adapt the broadcast schedule to the adversary's behavior; otherwise there is no sub-exponential solution.) If the devices share a synchronized view of the world, then it is easy for them to agree on a schedule that avoids contention; unfortunately, the adversary can prevent the devices from maintaining such a synchronized view, which can result in accidental contention. An important use of multi-selectors is in ensuring that the views do not diverge *too much*, which ensures that the dynamically chosen schedules result in relatively little contention.

The performance of our protocol depends on the relationship of t, the number of channels that can be simultaneously disrupted, and C, the total number of available channels. When the adversary can block no more than (approximately) $\sqrt{C}$ of the channels, the protocol has a linear $O(n)$ time. When the adversary can block $t = C - 1$ channels, leaving only one channel free for communication, the protocol is exponential in t. We derive from a lower bound on *multi-selectors* a proof that when $t = C - 1$, every information exchange protocol requires exponential time. In the intermediate cases where $\sqrt{C} < t < C - 1$, we show how the running times increases as the number of disrupted channels increases. (Figure 5 summarizes the performance in more details.)

In the remainder of this section, we present the basic communication model (Section I-A), we describe the problem of information exchange (Section I-B), and we discuss some related work (Section I-C). In Section II, we introduce the idea of multiselectors. In Section III, we present our basic algorithm for exchanging information when $C = \Omega(t^2)$. In Section IV, we show how to modify the protocol for the case where not as many channels are available, and we show a lower bound when $t = C - 1$. Finally, we conclude with some open questions in Section V. For proofs omitted due to space, see the full version of the paper [6].

A. Basic Model

In this paper, we consider a set of n deterministic nodes $P = \{p_1, \ldots, p_n\}$. Nodes communicate via a synchronous single-hop radio network with multiple-access channels (MAC). In each round, each node chooses a single channel $x \in \{1, \ldots, C\}$ and either *transmits* or *listens* on channel x. If exactly one node transmits on channel x, then every node listening on x receives that message. Otherwise, the listening nodes receive nothing. We do not assume collision detection.

The network is subject to interference that can prevent communication. We assume that an adaptive adversary can disrupt up to t channels in each round. When the adversary chooses to disrupt some channel $x \in \{1, \ldots, C\}$, none of the nodes listening on channel x receive a message. We assume that t is polynomially smaller than n: for some $\epsilon < 1/6$, $t = o(n^\epsilon)$. In real networks, the number of nodes tends to be much larger than the number of channels; since $t < C$, it is not unrealistic to assume that n is significantly larger than t.

B. Basic Problem

We study the fundamental problem of information exchange: the nodes are initialized with values $\{v_1, \ldots, v_n\}$. Each node attempts to learn as many values as possible. For $t \geq 1$, it is impossible for *all* the nodes to learn *all* the values. To see why, consider the case where the adversary disrupts communication by some set P' of t different nodes. In this case, no node in P' learns any value other than its own, and no node not in P' learns the value of a node in P'. Thus, the best we can hope to achieve is $(n-t)$-to-$(n-t)$ information exchange: eventually, all but t nodes learn all but t values. We call this variant: *almost-complete information exchange*.

C. Related Work

Selectors were first introduced by Komlos and Greenberg [3], and have been widely studied, particularly in the context of group property testing and radio networks (e.g., [4], [7]–[9]). Given a set $S \subseteq P$, a set S' is said to select an element $i \in P$ if $S \cap S' = \{i\}$. A k-selector is a sequence of sets $S_1, \ldots, S_m$ where for each set S of size k, at least 1 of the elements in S is selected by some set S_i. A multi-selector generalizes a selector in that it *simultaneously* selects a set of elements. We come back to this notion later in the paper.

Much research has been devoted to information exchange in the context of single-channel, fault-free radio networks (e.g., [3], [10]–[18]), particularly with respect to channel contention. There has been some research on *crash failures* in radio networks (e.g., [19]–[21]), and also on *Byzantine-resilient* broadcast [22], [23]. However, in these latter papers, communication is reliable and not subject to adversarial disruption. There have been two main approaches for coping with disruption. The first assumes that messages may be corrupted *at random* (e.g., [24]); the second bounds the number of messages that the adversary can transmit or disrupt, due, for example, to a limited energy budget (e.g., [25], [26]).

Some systems use pseudo-random frequency hopping based on shared "secrets" to avoid disruption (e.g., Bluetooth [1]). It is often unreasonable, however, to assume the existence of shared secrets for all possible sets of wireless devices that may eventually want to communicate.

The present paper, along with [2], [5], are the first, to the best of our knowledge, to consider multi-channel networks subject to malicious disruption in which nodes do not possess *a* *priori* shared secrets. Dolev et al. [5] consider *oblivious* (non-adaptive) protocols. They prove, for the special case of $t = 1$, a tight bound of $\Theta(n^2/C^2)$ for information exchange. Extended for general t, they achieve a running time of $O((en/t)^{t+1})$. The adaptive strategies in this paper outperform the optimal oblivious solutions in [5].

Dolev et al. [2] consider randomized algorithms in the context of a weak adversary that cannot determine on which channel a node is broadcasting until the broadcast is complete. In this paper, we consider deterministic protocols, and we assume that the adversary can always determine which channels are in use.

II. Simultaneous Selection

We now introduce *multi-selectors*, a combinatorial tool that captures the idea of simultaneous selection, generalizing the classical notion of *selectors* (see [3], [4]). We then provide upper and lower bounds on the size of a multi-selector.

A. Definitions

We first define a multi-selector that selects exactly one set of size k simultaneously:

Definition 1. An (*n*,*c*,*k*)-multi-selector, where $n \geq c \geq k \geq 1$, is a sequence of functions $M_1, M_2, \ldots, M_m$ from $P \rightarrow [1, c]$ such that:

> For every subset $S \subseteq P$ where $|S| = k$, there exists some $\ell \in [1, m]$ such that M_ℓ maps each element in S to a unique value in $[1, c]$.

We say that such a multi-selector has size m. A *generalized* multi-selector selects many sets of size k simultaneously; it generalizes both selectors and multi-selectors:

Definition 2. A generalized (*n*,*c*,*k*,*r*)-multi-selector, where $n \geq c \geq k \geq 1$ and $n \geq r$, is a sequence of functions $M_1, M_2, \ldots, M_m$ from $P \rightarrow [0, c]$ such that:

> For every subset $S \subseteq P$ where $|S| = r$, for every subset $S' \subseteq S$ where $|S'| = k$, there exists some $\ell \in \{1, \ldots, m\}$ such that (1) M_ℓ maps each element in S' to a unique value in $\{1, \ldots, c\}$, and (2) M_ℓ maps each element in $S \setminus S'$ to 0.

B. Upper Bound

We now show that there exist (n, c, k)-multi-selectors and determine their size. The proof is non-constructive, and relies on the probabilistic method.

Theorem 1. *For every $n \geq c \geq k$, there exists an (n, c, k)-multi-selector of size:*

$$\begin{array}{rcl} c = k & : & \frac{ke^c}{\sqrt{2\pi c}} \ln \frac{en}{k} \\ c/2 < k < c & : & ke^k \ln \frac{en}{k} \\ k \leq c/2 & : & k2^{2k^2/c} \ln \frac{en}{k} \end{array}$$

Proof: We include here the proof for the case where $k \leq c/2$; the other cases are similar and can be found in the full version of the paper [6]. Let $m = k2^{2k^2/c} \ln \frac{en}{k}$, the desired bound.

For each M_ℓ, for each $i \in P$, choose $M_\ell(i)$ at random from $[1, c]$. We show that with some probability > 0, M is an (n, c, k)-multi-selector. Fix an arbitrary set $S \subseteq P$ where $|S| = k$. Consider a particular M_ℓ. We calculate the probability that each element of S is assigned a unique element in $[1, c]$. Since there are $\binom{c}{k}k!$ good mappings from k elements to $[1, c]$, and c^k total mappings of k elements to $[1, c]$ sets, we conclude that:

$$\Pr\{S \text{ is uniquely mapped}\} \quad = \quad \frac{\binom{c}{k}k!}{c^k} \quad = \quad \frac{c!}{(c-k)!c^k} \; .$$

Since $k \leq c/2$ we get the following estimate which we denote as q:

$$\Pr\{S \text{ is uniquely mapped}\} \geq \left(\frac{c-k}{c}\right)^k \geq 4^{-k^2/c} = q \; .$$

The probability that S is not well-mapped for all M_ℓ is at most $(1-q)^m$. Since $m = q^{-1} \cdot k \ln \frac{en}{k}$, the probability that S is not well-mapped for all M_ℓ is at most $e^{-k \ln \frac{en}{k}} \leq \left(\frac{k}{en}\right)^k$. Since there are only $\binom{n}{k} < \left(\frac{en}{k}\right)^k$ possible subsets S of size k, we argue (by a union bound) that the probability of some S being incorrectly mapped by all M_ℓ is at most $\binom{n}{k} \cdot \left(\frac{k}{en}\right)^k$, which is smaller than 1, implying the conclusion. ■

If c is sufficiently larger than k, there are efficient (n, c, k)-multi-selectors:

Corollary 2. *For every $n \geq c \geq k^2$, there exists an (n, c, k)-multi-selector of size $O(k \log(n/k))$.*

The same argument extends to bound the size of generalized multi-selectors:

Theorem 3. *For every $n \geq r \geq c \geq k$ such that $n \geq 2r$, there exists (n, c, k, r)-multi-selectors of size $O\left(r\frac{(c+1)^r e^k}{k^k} \log(en/r)\right)$ or $O\left(r\frac{(c+1)^r}{(c-k)^k} \log(en/r)\right)$.*

The proof can be found in the full version of the paper [6].

C. Constructing Multi-Selectors

There exists a connection between good hash functions and multi-selectors when $k^2 < c$. (In general, however, for other values of k and c, it is not immediately clear how multi-selectors relate to hash functions.) We discuss some of these connections and derive some multi-selector constructions.

First, we show how to use a universal family of hash functions to construct a (n, c, k)-multi-selector. A (two)-universal family of hash functions is a set of functions from universe P to some domain $\{1, \ldots, c\}$ such that for each pair $x, y \in P$, at least a $(1 - 1/n)$ fraction of the hash functions map x and y to a unique value. Carter et al. [27] present such a family of size $\Theta(n^2)$. This family of hash functions is also an (n, c, k)-multi-selector, for any $k < \sqrt{c}$: consider some set S of k elements; for each of the $O(k^2) = O(c)$ pairs, there are $\leq n$ hash functions that collide; thus there are at most $O(cn) < O(n^2)$ hash functions for which elements of S collide. The resulting multi-selector is of size $O(n^2)$.

We now derive a more efficient construction. Assume that c is sufficiently large such that there exist $p_1, \ldots, p_{\Theta(k^2 \log n)}$, a set of $\Theta(k^2 \log n)$ distinct primes less than c. Fix a set $S \subseteq P$ of size k. For every pair $x, y \in S$, there are at most $\log n$ primes p_i such that $x = y \mod p_i$. Thus there is some prime p_i such that none of the $\Theta(k^2)$ pairs in S collide. This results in an (n, c, k)-multi-selector of size $O(k^2 \log n)$.

If $k^2 = c$ then there are not a sufficient number of primes $\leq c$; the two techniques can be combined. The second technique reduces the channel range to $O(k^2 \log^2 n)$ (using the Prime Number Theorem to demonstrate sufficient prime numbers) using $O(k^2 \log n)$ mappings; the two-universal hash family of [27] reduces the channel range to c, multiplying each mapping by $O(k^4 \log^4 n)$. From this we conclude:

Theorem 4. *For every $n > c > k^2$, we can construct a (n, c, k)-multi-selector of size $O(k^6 \log^5 n)$.* □

It is also possible to construct multi-selectors using selectors. The resulting construction is not particularly efficient, but illustrates a connection between selectors and multi-selectors. The following theorem can be found in the full version of the paper [6]:

Theorem 5. *For every n, c, k, there exists a construction of a (n, c, k)-multi-selector of size $O(k^k \log^k n)$.*

D. Lower Bound

In this section, we prove a lower bound on the size of an (n, c, k)-multi-selector.

Theorem 6. *For some $m > 0$, let $M = M_1, \ldots, M_m$ be an (n, c, k)-multi-selector where $n \geq 2c$ and $c \geq k$. Then M has size at least:*

$$\begin{array}{rcc} c = k & : & \dfrac{2^c}{4\sqrt{2\pi c}} \\[2ex] c/2 < k < c & : & e^{k \ln \frac{c}{c-k} - k^2/n} \cdot \dfrac{\sqrt{n(c-k)}}{4\sqrt{c(n-k)}} \\[2ex] k \leq c/2 & : & e^{k^2/c - k^2/n} \cdot \dfrac{\sqrt{n(c-k)}}{4\sqrt{c(n-k)}} \end{array}$$

Proof: We consider the case where $k = c$; for the remaining cases, see the full version of the paper [6]. We begin by choosing a subset $S \subseteq P$ of size c at random. We calculate the probability that S is correctly mapped by some M_ℓ. We show that if $m < \frac{2^c}{4\sqrt{2\pi c}}$, then this probability is smaller than one, thus the probability that a random set S violates the definition of multi-selector M is positive. By the probabilistic argument, such a set S exists, which contradicts that M is an (n, c, k)-multi-selector.

Fix some $\ell \in [1, m]$, and define $S_d = \{i : M_\ell(i) = d\}$, that is, the subset of P that M_ℓ maps to d. To calculate the probability that M_ℓ correctly maps each element of S to a unique element of $[1, c]$, we first approximate the number of subsets of P that are correctly mapped by M_ℓ: $\prod_{d=1}^{c} |S_d| \leq (n/c)^c$. (The inequality follows from the relationship between the arithmetic and geometric means.) Since there are $\binom{n}{c}$ sets of size c, and since $(n - c) \geq n/2$, we conclude (via Stirling's approximation) that the probability that S is correctly mapped by M_ℓ is at most

$$\frac{n^c}{c^c \binom{n}{c}} \leq \frac{n^c}{\frac{n^n}{(n-c)^{n-c} \cdot 4\sqrt{2\pi c}}} = \frac{4\sqrt{2\pi c}}{\left(\frac{n}{n-c}\right)^{n-c}} \leq \frac{4\sqrt{2\pi c}}{2^c} \; .$$

Thus, the probability that S is correctly mapped by *any* of the m functions is at most $m \cdot 4\sqrt{2\pi c} 2^{-c}$ (by a union bound). If $m < \frac{2^c}{4\sqrt{2\pi c}}$, then with positive probability the set S is not correctly mapped by any of the M_ℓ, resulting in a contradiction. ■

III. Reliable Information Exchange

We now present our protocol for solving the problem of reliable information exchange. In this section, we assume that $C \gg t$, specifically $C = \Theta(t^2)$. In Section IV, we show how to adapt this protocol to the case where $C = t + 1$, and conclude with a discussion of the remaining cases.

The protocol adaptively chooses a set of nodes to transmit in each round based on which nodes have already succeeded in previous rounds. Adapting to the past proves challenging as nodes do not share a uniform view: a node does not know *a priori* which transmissions succeeded, unless it was listening on that channel. Our protocol circumvents this challenge by using a $(n, c, t + 1)$-multi-selector to ensure that *almost* all the nodes have the same view. Nodes use a multi-selector to guide their channel selection when attempting to receive updates on the system state; since a multi-selector guarantees the simultaneous selection of any subset of size $t + 1$, it follows that for any group of size $t + 1$ nodes, there exists a round during which these nodes are listening on different channels. Therefore, at most t total can be kept ignorant by the adversary. This bound on ignorance allows efficient and consistent adaptation.

Preliminaries: For the remainder of this section, we fix the constant $c = (5t + 1)^2$. Of the C available channels, our protocol will use exactly c. Recall here that n is assumed to be large compared to t, specifically, that $t = o(n^\epsilon)$ for some $\epsilon < 1/6$. It follows: (a) $n \geq c^2(5t+1)+5t$; and (b) $n \geq c^2 t+c$.

We refer to values as either *complete* or *incomplete*. Initially, each value is *incomplete*; when a value is received by at least $n - t$ nodes, it is designated as *complete*; the node at which it originated is said to have *completed*. We use the notation $S[k]$ to refer to the k^{th} value in a set S under some fixed ordering. When given a set S comprised of sets, we use $S[j][k]$ to refer to the k^{th} value of the j^{th} set also under some fixed ordering.

Information Exchange: The main routine for the information exchange protocol is in Figure 1. It consists of two parts, each consisting of a set of *epochs*. In each part, a set of *listeners* is chosen, and they facilitate the dissemination of incomplete values. The listeners' own initial values are not disseminated, however, as they are busy listening; hence each part chooses a disjoint set of listeners: $\{p_1, \ldots, p_{c^2}\}$ in the first part, and $\{p_{c^2+1}, \ldots, p_{2c^2}\}$ in the second part. Each part disseminates (i.e., *completes*) all but at most $2t$ non-listener

Figure 1: Information exchange routine for node p_i.

```
InfoExchange$()_i$                                                        ▷ $E$ defines the length of each epoch.
    $L \leftarrow$ a partition of the set $\{1, \ldots, c^2\}$ into $c$ sets of size $c$.    ▷ Each $L[k]$ is a set of listeners.
    for $e = 1$ to $|E|$ do                                               ▷ First set of epochs:
        $knowledgeable \leftarrow$Epoch$(L, knowledgeable, E[e])_i$

    $L \leftarrow$ a partition of the set $\{c^2 + 1, \ldots, 2c^2\}$ into $c$ sets of size $c$.
    for $e = 1$ to $|E|$ do                                               ▷ Second set of epochs:
        $knowledgeable \leftarrow$Epoch$(L, knowledgeable, E[e])_i$

    ▷ Lastly, do the special epoch which attempts to transmit the final $\leq 4t$ values.
    Special-Epoch$(knowledgeable)_i$
```

values. Thus, after the two parts, at most $4t$ values are left incomplete in total. The final call to Special-Epoch reduces the number of incomplete values from $4t$ to t, as required.

The function $E(r)$ bounds the length of epoch r and the number of epochs. We define it recursively. Let $E(1) = \lceil n/c \rceil$. For all $r > 1$, let $E(r) = \left\lceil \frac{2t \cdot E(r-1)}{c} \right\rceil$. The sequence terminates when $E(r) = 1$. Notice that $|E| = O(\log n)$ and $\sum E = O(n/c)$.

Epochs: In each call to Epoch, some set of incomplete values are *completed*; i.e., disseminated to at least $n - t$ nodes. At the end of an epoch, each node is designated as *knowledgeable* or *unknowledgeable* based on the outcome of the epoch: a knowledgeable node knows the results of all preceding epochs, including the current set of completed values; an unknowledgeable node does not have this information.

The epoch pseudocode is in Figure 2. For each epoch, we are given (1) a set of listeners L, (2) a flag $knowledgeable$, indicating the status of node i, and (3) a number $rnds$ indicating the length of the aggregation phase. We conclude:

Lemma 7. *If some epoch begins with s incomplete nodes in the set $P \setminus L$, then at the end of the epoch, there are at most $2t\lfloor s/c \rfloor$ incomplete nodes in $P \setminus L$.*

Aggregation: In the first phase of an epoch (lines 2–9), values are transmitted to the listeners in the set L. Let S be the set of nodes that have not yet completed. The set S is divided into subsets of size c, each of which is scheduled to transmit in one of the subsequent $\lfloor |S|/c \rfloor$ rounds. Only knowledgeable nodes can calculate S; thus only nodes that are both knowledgeable and incomplete transmit.

Throughout, c listeners are scheduled to listen on each channel. In each of these rounds, the adversary can block up to t; moreover, up to t of the nodes "scheduled" to transmit in a round may in fact be unknowledgeable and hence not transmit. Thus, in each round, at most $2t$ values are not successfully received by the listeners. By the end of the aggregation phase, only $2t\lfloor |S|/c \rfloor$ values remain incomplete.

Dissemination: In the second phase of an epoch, the listeners disseminate their information. The pseudocode for Disseminate is in Figure 3. The disseminate routine ensures:

Lemma 8. *If some value v is known to a set of listeners when the disseminate routine begins, then the value is complete at the end of the disseminate routine.*

In Part 1 (lines 3–9), each of the c sets of c listeners attempts to disseminate its set of values. For each set (lines 5–9), each of the c listeners in the set transmits continually on a unique channel (line 7). An $(n, c, t+1)$-multi-selector M is used to schedule the non-listener nodes (line 8). While the listeners are broadcasting, the non-listeners choose which channel to receive on according to M. This ensures that for any set of $t + 1$ non-listeners, there is some round in which they are all receiving simultaneously on different channels. As a result, at most t can be disrupted by the adversary. Since there are c sets of listeners, this results in at most ct nodes that do not receive a value from *all* c sets of listeners.

In Part 2 (lines 11–19), we select a larger set of $c(ct + 1)$ nodes, which we partition into sets of size c. (Recall that $n \geq c(ct + 1)$.) At least one of these $ct + 1$ partitions consists only of nodes that have received a message from all c sets of listeners in Part 1. Thus, all the nodes in the set know all the values known to all the sets of listeners. As before, each of these sets transmits its information to the remaining nodes in such a way that at most t nodes can fail to learn these values.

Special Epoch: In order to transmit the remaining values, we execute a special epoch. The pseudocode for Special-Epoch is in Figure 4. The special epoch operates somewhat differently, as there are very few values left to transmit. As before, we use listeners to collect the values; we need to choose a set of listeners that have already completed. Recall, up to $4t$ values may be incomplete after the two sets of epochs. An additional t nodes might be complete but not aware of it because they are unknowledgeable. This leaves at most $5t$ nodes that are not complete and knowledgeable. We refer to these as *special* nodes. We choose a set of $c^2(5t+1)$ possible listeners, and divide them into $5t + 1$ sets of size c^2; at least one of these sets contains only nodes that are complete and knowledgeable. We use a $(n, c, 5t)$-multi-selector to ensure that in some round, each of the $k \leq 5t$ special nodes is assigned to a different channel to transmit; at most t can be blocked. Dissemination proceeds as before.

Performance: Each epoch e spends $E(e)$ rounds during the aggregation phase, resulting in $O(n/c)$ rounds of

Figure 2: Epoch routine for node p_i.

```
Epoch($L, knowledgeable, rnds$)$_i$                                  ▷ $L$ is an array of sets of listeners.
    $S \leftarrow \emptyset$
    if $knowledgeable =$ true then
        let $S$ be the set of nodes that are not in $L$ and not completed.
        Partition $S$ into $\lceil |S|/c \rceil$ sets of size $c$.          ▷ Denote by $S[k]$ the $r$th such set.
    for $r = 1$ to $rnds$ do
        if ($knowledgeable =$ true) and ($r \leq \lceil |S|/c \rceil$) then
            if $\exists k \in \{1, ..., c\} : i = S[r][k]$ then schedule $i$ to transmit on channel $k$.
            if $\exists k \in \{1, ..., c\} : i \in L[k]$ then schedule $i$ to receive on channel $k$.
    $knowledgeable \leftarrow$Disseminate($L[1], \ldots, L[c]$)$_i$
    return $knowledgeable$
```

Figure 3: Disseminate routine for node p_i.

```
Disseminate($L[1], \ldots, L[c]$)$_i$                                 ▷ Each $L[k]$ is a set of former listeners.
    let $M$ be a $(n, c, t+1)$-multiselector.
    ▷ Part 1: Ensure that for each listener group, all but some set of $t$ nodes receive its value set.
    $knowledgeable \leftarrow$ true
    for $k = 1$ to $c$ do
        for each round $r = 1$ to $|M|$
            if $\exists j \in \{1, ..., c\} : i = L[k][j]$ schedule $i$ to transmit on channel $j$.
            if $i \notin L[k]$ then schedule $i$ to receive on channel $M_r(i)$.
        if $i$ does not receive a message in any of the $|M|$ rounds then $knowledgeable \leftarrow$ false.

    ▷ Part 2: Ensure that all but some set of $t$ nodes receive all the value sets from all the listener groups.
    $L' \leftarrow$ an arbitrary subset of $\{1, \ldots, n\}$ of size $c(ct+1)$.
    Partition $L'$ into $ct+1$ sets $L'[1], \ldots, L'[ct+1]$ of size $c$
    for each $s = 1$ to $ct+1$ do
        for each $r = 1$ to $|M|$ do
            if $\exists j \in \{1, ..., c\} : i = L'[s][j]$ schedule $i$ to transmit on channel $j$
            if $i \notin L'[s]$ then schedule $i$ to receive on channel $M_r(i)$.
        if $i$ receives a message in any of the $|M|$ rounds from a node with $knowledgeable =$ true then
            $knowledgeable \leftarrow$ true
    return $knowledgeable$
```

Figure 4: Special Epoch routine for node p_i.

```
Special-Epoch($knowledgeable$)$_i$
    let $M$ be an $(n, c, 5t)$-multiselector.
    $special \leftarrow$ false
    if ($knowledgeable =$ false) or ($i$ has not completed) then $special \leftarrow$ true
    if $knowledgeable =$ true then
        $L \leftarrow$ set of $c^2(5t+1)$ smallest nodes that have completed in a previous epoch.
        Partition $L$ into $(5t+1)$ sets $L_1, \ldots, L_{t+1}$ of size $c^2$.
        Partition each $L_k$ into $c$ sets $L_k[1], \ldots, L_k[c]$ of size $c$.
    for $s = 1$ to $5t+1$ do
        for $r = 1$ to $|M|$ do
            if $special =$ true then schedule $i$ to transmit on channel $M_r(i)$
            if $\exists k : i \in L_s[k]$ then schedule $i$ to receive on channel $k$.
        Disseminate($L_s[1], \ldots, L_s[c]$)$_i$
```

aggregation. Each epoch e performs $c|M| + (ct+1)|M|$ rounds of dissemination. By Corollary 2, we conclude that $|M| = O((t+1)\log n/(t+1))$; and thus during $O(\log n)$ epochs, there are $O(ct^2 \log^2 n)$ rounds of dissemination. Finally, we observe that the special epoch aggregation has running time $(5t+1)|M|$ where M is a multi-selector of size at most $O(t \log n/(5t))$ (again by Corollary 2). Thus the special epoch has round complexity $O(t^2 \log n/t)$, along with $O(t)$ disseminations. Summing these costs and substituting in for $c = O(t^2)$ and $t = o(n^{1/6})$, we conclude that:

Theorem 9. *Within $O(n)$ rounds, all but t values are complete. More precisely, the information exchange protocol has round complexity $O(n/t^2 + t^5 \log^2 n)$.*

IV. Limiting the Number of Channels

We consider here the case where there are fewer than t^2 channels available. We first describe how to adapt the protocol of Section III to the setting where $C = t+1$, the minimal number of channels for which information exchange is feasible. We then present a lower bound showing that the time complexity is inherently exponential in t. Finally, we briefly discuss the intermediate cases where $t+1 < C < \Theta(t)^2$.

A. Protocol Description

In this section, we modify the information exchange routine to use only $C = t+1$ channels. The disseminate protocol (Section III) can be used without modification. We replace, however, Epoch and Special-Epoch with Limited-Epoch (Figure 6) and Limited-Special-Epoch (Figure 7), respectively.

The key problem addressed is as follows: since only $t+1$ channels are available, if any of the $t+1$ nodes scheduled in a round are unknowledgeable and therefore choose not to transmit, then the adversary can disrupt all $\leq t$ nodes that do broadcast. In order to circumvent this problem, we use a $(n, C, C, 2t+1)$-generalized-multi-selector in the aggregation phase of Limited-Epoch. Nodes know at the beginning of a round if they are scheduled or if they are unknowledgeable. Such nodes will attempt to transmit according to the schedule described by the generalized multi-selector. The multi-selector guarantees that one of the rounds will simultaneously select the $t+1$ nodes that are actually scheduled to transmit during this round of the epoch, some of which might be unknowledgeable. From this we conclude that at least 1 incomplete value is transmitted to the listeners for each round of the schedule. The function E is redefined as follows: for $r > 1$, $E(r) = \lceil \frac{E(r-1)t}{C} \rceil$. In this case, Limited-Special-Epoch only has to cope with at most $3t$ "special" nodes—t from each set of epochs, and as many as t additional unknowledgeable nodes. A $(n, C, C, 3t)$-generalized-multi-selector is used to ensure that all subsets of size $t+1$ of these (no more than) $3t$ special nodes get an opportunity to transmit concurrently.

Performance: The total running time of the aggregation phases is now $O(n|M_a|)$, where $|M_a| = O((2t+1)(C+1)^{2t+1} \log n/(2t+1))$ by Theorem 3 and the fact that $e < t+1$. Dissemination has running time $(Ct+1)|M_d|$, where in this case $|M_d| = O((t+1)e^{t+1} \log n/(t+1))$ by Theorem 1; the number of disseminations is bounded by n/t. Finally, the special epoch costs a factor of $O(t)$ more than a regular epoch. We conclude (with some loose approximations) that:

Theorem 10. *When $C = t+1$, the information exchange protocol terminates in time:*

$$O\left(n(C+1)^{3t} \log \frac{n}{t}\right)$$

B. Lower Bound

In this section, we show that if $C = t+1$, every information exchange protocol is exponential in t.

Theorem 11. *Every almost-complete information exchange protocol where $C = t+1$ requires at least time $\Omega(2^{t+1}/\sqrt{t+1})$.*

Proof: Consider a protocol that solves almost-complete information exchange in m rounds. We construct a $(n, C, t+1)$-multi-selector of length m, and invoke Theorem 6 to conclude the proof. We construct the multi-selector by simulating the information exchange protocol in each round:

- Every node that is scheduled to listen is simulated as if it receives no messages in that round (as if the adversary had disrupted the channel).
- Every node that is scheduled to transmit on some channel is simulated as if it transmits its message.

(Notice, the resulting simulation might violate our model assumptions by allowing more than t channels to be disrupted.) For each round r of this simulated execution, we construct M_r as follows: if a node i listens on channel k, then $M_r(i) \leftarrow k$; otherwise, if node i does not listen on any channel (either because it transmits or because it does nothing), then M_r maps i to 1, a default.

We argue that M is a $(n, C, t+1)$-multi-selector. Assume for the sake of contradiction that it is not. Then, for some set S of size $t+1$, no M_ℓ maps S to unique channels. We now construct a new execution. This time the adversary always and only disrupts the channels occupied by nodes in S, ignoring the other nodes in the system. To the nodes in S this execution looks indistinguishable from our original simulated execution (in both, they receive nothing in all rounds). Therefore, they behave the same, never occupying more than t channels. It follows that the adversary never has to disrupt more than t channels per round in this second simulation, meaning that it is feasible in our model. This feasible execution cannot solve almost-complete information exchange because none of the $t+1$ nodes in S ever receive a message. This contradicts the assumption that the protocol under consideration solves the problem in m rounds.

We can therefore conclude that the original assumption was wrong, and conclude that M is indeed an $(n, C, t+1)$-multi-selector. The lower bound then follows from applying Theorem 6 with $C = t+1$. ■

C. Generalizing the Number of Channels

We have discussed the case where $C = \Theta(t^2)$ and the case where $C = t+1$. We briefly addresses the performance of

Channels	Running time	Calculation
$C \geq (5t+1)^2$	$O(n)$	$O(n/c + ct\lvert M_1\rvert \log n + ct^2\lvert M_1\rvert + t\lvert M_2\rvert)$
$C \geq 10t$	$O\left(n2^{\frac{2t^2}{C}} + t^2 2^{\frac{50t^2}{C}} \log n\right)$	$O(n/c + ct\lvert M_1\rvert \log n + ct^2\lvert M_1\rvert + t\lvert M_2\rvert)$
$C \geq 5t$	$O\left(n2^{\frac{2t^2}{C}} + t^2 e^t \log n\right)$	$O(n/c + ct\lvert M_1\rvert \log n + ct^2\lvert M_1\rvert + t\lvert M_2\rvert)$
$C \geq 2t+1$	$O\left(n2^{\frac{2t^2}{C}} + t(C+1)^{3t} e^c \log \frac{n}{5t}\right)$	$O(n/c + ct\lvert M_1\rvert \log n + ct^2\lvert M_1\rvert + t\lvert M_3\rvert)$
$C \geq t+1$	$O\left(nt(C+1)^{2t+1} \log \frac{n}{2t+1} + t^2(C+1)^{3t} \log \frac{n}{3t}\right)$	$O(n\lvert M_4\rvert + nct\lvert M_1\rvert + ct^2\lvert M_1\rvert + t\lvert M_5\rvert)$

M_1 :	$(n, C, t+1)$-multi-selector	M_2 :	$(n, C, 5t)$-multi-selector
M_3 :	$(n, C, C, 5t)$-multi-selector	M_4 :	$(n, C, C, 2t+1)$-multi-selector
M_5 :	$(n, C, C, 3t)$-multi-selector		

Fig. 5. For each value of C, there exists an protocol that runs in the specified time. The secondary table specifies the parameters of the multi-selectors. The running time is calculated by instantiating each multi-selector with the best bound presented in Section II.

information exchange for intermediate values of C; running times are summarized in Figure 5. When $C < 2t+1$, the aggregation phase requires generalized multi-selectors as in Limited-Epoch. It follows that the running time does not differ significantly for $t+1 \leq C \leq 2t+1$. For $C \geq 5t+1$, we can use the protocol described in Section III, where the multi-selectors are sized appropriately; as C grows the running time decreases, as the greater number of available channels reduces the size of the multi-selectors. For $2t+1 < C < 5t+1$, we use a hybrid protocol in which Disseminate stays the same, but Special-Epoch uses generalized multi-selectors as in Limited-Special-Epoch. It is straightforward to calculate the associated running times which can be found in Figure 5.

V. Open Questions

Beyond the results in this paper, we believe that multi-selectors may prove to be an important tool in developing other protocols for multi-channel networks, especially given that multi-channel networks are increasingly viewed as the most promising approach for coping with malicious disruption. We expect that multi-selectors will play a key role in adapting single-channel protocols for a multi-channel environment. (This is especially the case for the large subset of single-channel protocols that are themselves based on selectors.) Moreover, much in the way that selectors have proved useful in a variety of settings, ranging from wireless communication to group property testing, we hope that multi-selectors will find a similar wide range of applications. For example there are possible connections to *renaming* and *k-set agreement*, both of which depend on simultaneously allocating a set of scarce resources (in this case, names or decision values).

Interesting open questions include: (1) deriving better constructive bounds for multi-selectors; (2) studying other algorithmic uses of multi-selectors; (3) determining the complexity of information exchange as t approaches n; (4) studying the tradeoff between the number of channels, the resilience, and the performance in terms of different complexity measures, such as energy usage; (5) studying the capacity of a wireless network under the influence of a strong, malicious adversary.

References

[1] *Bluetooth Specification V.2.1*, July 2007, http://www.bluetooth.com/NR/rdonlyres/F8E8276A-3898-4EC6-B7DA-E5535258B056/6545/Core_V21__EDR.zip.

[2] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport, "Secure communication over radio channels," in *Proceedings of the Symposium on Principles of Distributed Computing (PODC)*, August 2008.

[3] J. Komlos and A. Greenberg, "An asymptotically fast non-adaptive algorithm for conflict resolution in multiple access channels," *IEEE Trans. on Information Theory*, pp. 302–306, March 1985.

[4] A. D. Bonis, L. Gasieniec, and U. Vaccaro, "Optimal two-stage algorithms for group testing problems," *SIAM J. on Computing*, vol. 34, no. 5, pp. 1253–1270, 2005.

[5] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport, "Gossiping in a multi-channel radio network: An oblivious approach to coping with malicious interference," in *Proceedings of the Symposium on Distributed Computing (DISC)*, 2007.

[6] S. Gilbert, R. Guerraoui, D. R. Kowalski, and C. Newport, "Interference-resilient information exchange," Tech. Rep., 2008, http://lpd.epfl.ch/sgilbert/pubs/InfoExchange-TR.pdf.

[7] P. Indyk, "Explicit constructions of selectors and related combinatorial structures, with applications," in *Proceedings of the Symposium on Discrete Algorithms (SODA)*, 2002.

[8] M. Chrobak, L. Gasieniec, and W. Rytter, "Fast broadcasting and gossiping in radio networks," *J. of Algorithms*, vol. 43, no. 2, pp. 575–581, 2002.

[9] M. Chrobak, L. Gasieniec, and D. R. Kowalski, "The wake-up problem in multi-hop radio networks," *SIAM Journal of Computing*, vol. 36, no. 5, pp. 1453–1471, 2007.

Figure 6: Epoch routine for node p_i where $C = t + 1$.

```
Limited-Epoch$(L, knowledgeable, rnds)_i$
    let $M$ be a $(n, C, C, 2t+1)$-generalized-multiselector.
    $S \leftarrow \emptyset$
    if $knowledgeable =$ true then
        Let $S$ be the set of nodes that are not in $L$ and not completed.
        Partition $S$ into $\lceil |S|/c \rceil$ sets of size $C$.
    for $r_1 = 1$ to $rnds$ do
        if $(r_1 \leq \lceil |S|/C \rceil)$ then
            for $r_2 = 1$ to $|M|$ do
                if $i \notin L$ and $(($i is not $knowledgeable)$ or $(i \in S[r_1]))$ then schedule $i$ to  transmit on channel $M_{r_2}(i)$.
                if $\exists k \in \{1, ..., C\} : i \in L[k]$ then schedule $i$ to receive on channel $k$.
    $knowledgeable \leftarrow$Disseminate$(L[1], \ldots, L[C])_i$
    return $knowledgeable$
```

Figure 7: Special Epoch routine for node p_i where $C = t + 1$.

```
Limited-Special-Epoch$(knowledgeable)_i$
    Let $M$ be an $(n, C, C, 3t)$-multiselector
    $special \leftarrow$ false
    if $(knowledgeable =$ false$)$ or $(i$ has not completed$)$ then $special \leftarrow$ true
    if $knowledgeable =$ true then
        $L \leftarrow$ set of $c^2(3t+1)$ smallest nodes that have completed in a previous epoch.
        Partition $L$ into $(3t+1)$ sets $L_1, \ldots, L_{t+1}$ of size $c^2$.
        Partition each $L_k$ into $c$ sets $L_k[1], \ldots, L_k[c]$ of size $c$.
    for $s = 1$ to $3t + 1$ do
        for $r = 1$ to $|M|$ do
            if $special =$ true then schedule $i$ to transmit on channel $M_r(i)$
            if $\exists k : i \in L_s[k]$ then schedule $i$ to receive on channel $k$.
        Disseminate$(L_s[1], \ldots, L_s[c])_i$
```

[10] A. Czumaj and W. Rytter, "Broadcasting algorithms in radio networks with unknown topology," in *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, October 2003.

[11] D. E. Willard, "Log-logarithmic selection resolution protocols in a multiple access channel," *SIAM J. of Computing*, vol. 15, no. 2, pp. 468–477, 1986.

[12] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, "A lower bound for radio broadcast," *J. of Computer and System Sciences*, vol. 43, no. 2, pp. 290–298, October 1992.

[13] R. Bar-Yehuda, O. Goldreich, and A. Itai, "On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization," *J. of Computer and System Sciences*, vol. 45, no. 1, pp. 104–126, 1992.

[14] D. Kowalski and A. Pelc, "Time of deterministic broadcasting in radio networks with local knowledge," *SIAM J. on Computing*, vol. 33, no. 4, pp. 870–891, 2004.

[15] E. Kushelevitz and Y. Mansour, "An $\Omega(Dlog(N/D))$ lower bound for broadcast in radio networks," in *Proceedings of the Symposium on Principles of Distributed Computing (PODC)*, 1993.

[16] B. Chlebus and D. Kowalski, "Robust gossiping with an application to consensus," *J. of Computer and System Sciences*, vol. 72, pp. 1262–1281, 2006.

[17] B. S. Chlebus, L. Gasieniec, A. Lingas, and A. T. Pagourtzis, "Oblivious gossiping in ad-hoc radio networks," in *Proceedings of the Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, 2001.

[18] R. Bar-Yehuda, A. Israeli, and A. Itai, "Multiple communication in multi-hop radio networks," *SIAM J. on Computing*, vol. 22, no. 4, pp. 875–887, 1993.

[19] E. Kranakis, D. Krizanc, and A. Pelc, "Fault-tolerant broadcasting in radio networks," *Journal of Algorithms*, vol. 39, no. 1, pp. 47–67, 2001.

[20] A. Clementi, A. Monti, and R. Silvestri, "Round robin is optimal for fault-tolerant broadcasting on wireless networks," *J. of Parallel and Distributed Computing*, vol. 64, no. 1, pp. 89–96, 2004.

[21] ——, "Optimal f-reliable protocols for the do-all problem on single-hop wireless networks," in *Proceedings of the International Symposium on Algorithms and Computation (ISAAC)*, 2002, pp. 320–331.

[22] C.-Y. Koo, "Broadcast in radio networks tolerating byzantine adversarial behavior," in *Proceedings of the Symposium on Principles of Distributed Computing (PODC)*, 2004, pp. 275–282.

[23] V. Bhandari and N. H. Vaidya, "On reliable broadcast in a radio network." in *Proceedings of the Symposium on Principles of Distributed Computing (PODC)*, 2005, pp. 138–147.

[24] A. Pelc and D. Peleg, "Feasibility and complexity of broadcasting with random transmission failures," in *Proceedings of the Symposium on Principles of Distributed Computing (PODC)*, 2005.

[25] C.-Y. Koo, V. Bhandari, J. Katz, and N. H. Vaidya, "Reliable broadcast in radio networks: The bounded collision case," in *Proceedings of the Symposium on Principles of Distributed Computing (PODC)*, 2006.

[26] S. Gilbert, R. Guerraoui, and C. Newport, "Of malicious motes and suspicious sensors: On the efficiency of malicious interference in wireless networks," in *Proceedings of the Conference on Principles of Distributed Systems (OPODIS)*, December 2006.

[27] J. L. Carter and M. N. Wegman, "Universal classes of hash functions (extended abstract)," in *Proceedings of the Symposium on Theory of Computing (STOC)*, 1977.