Interference-Resilient Information Exchange

Seth Gilbert* Rachid Guerraoui* Dariusz R. Kowalski† Calvin Newport‡

Abstract

This paper examines information exchange in the context of a multi-channel radio network subject to malicious interference. In each round, up to t channels may be disrupted, preventing successful message transmission. This disruption models the various forms of electromagnetic interference common in a radio setting. We focus on the well-known *gossip* problem, which has participants attempt to disseminate their initial values.

We present upper and lower bounds on deterministic solutions to the gossip problem. Our results are based on a new combinatorial tool: the *multiselector*. This mathematical object helps decompose reliable multi-channel communication into a problem of *simultaneous* selection, and, in this sense, is of independent interest. We demonstrate both upper and lower bounds on the size of multiselectors, and draw connections between the object and both classic selectors and hash functions.

Using multiselectors as our key tool, we study gossip under two conditions. In the first, the total number of available channels is much larger than t. We present an optimal $O(n)$ time gossip solution in this setting. In the second, we assume only $t + 1$ channels are available, the minimum number of channels for which the problem is solvable. We present an algorithm that runs in time exponential in $t + 1$, and then derive a lower bound of $\Omega(2^{t+1}/\sqrt{t+1})$ rounds—showing an exponential in t to be unavoidable. We conclude with a brief discussion of how the time complexity evolves as the number of channels moves between these two extremes. Our results provide a significant improvement over the best existing multi-channel gossip solution: an oblivious algorithm that runs in time $\Omega((en/t)^t)$.

Keywords: Gossip, multi-channel radio networks, multiselectors, selectors, combinatorics.

*I&C School of Computer and Communication Sciences, EPFL, 1015, Lausanne, Switzerland.
†Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK.
‡CSAIL, MIT, Cambridge MA., USA.

1 Introduction

We study the fundamental problem of information exchange, often called *gossip*. In this problem, a set of n processes $P = \{p_1, \ldots, p_n\}$ are initialized with values $\{v_1, \ldots, v_n\}$, respectively. These values are called *rumors*. The problem has each process attempt to learn as many rumors as possible.

We study *deterministic* gossip protocols in the context of a single-hop radio network, i.e., all processes are within communication range. The network operates in synchronous rounds. In each round, each process chooses a single channel $c \in \{1, \ldots, C\}$ and either *transmits* or *listens* on channel c. If *exactly one* process transmits on channel c, then every process listening on channel c receives that message. Otherwise, the listening processes receives nothing. (We do not assume collision detection.)

The network is subject to *malicious* interference that can prevent the processes from communicating. This is an important issue in wireless networks where a malicious device can disrupt communication by broadcasting noise, thus "jamming" the electromagnetic spectrum. Similar effects can be derived from non-malicious devices accidently contending on the same portion of the spectrum: e.g., the wireless telephone interfering with the nearby 802.11 base station. We assume a malicious adversary that can disrupt up to t channels in each round. When the adversary chooses to disrupt some channel $c \in \{1, \ldots, C\}$, none of the processes listening on channel c receive a message. Throughout this paper, we assume that t is polynomially smaller than n, that is, for some $\epsilon < 1/6$, $t = O(n^\epsilon)$.

For $t \geq 1$, the processes cannot learn *all* the rumors. More precisely, for every algorithm A, there is an execution of A in which: (1) No process learns more than $(n - t)$ rumors; and (2) at least t processes learn no more than one rumor each. To see why, let P' be an arbitrary set of t processes and consider the case where the adversary disrupts all communication by processes in P'. For each $p_i \in P'$, no other process ever learns rumor v_i, and no process in P' ever learns any rumor other than its own.

The best we can hope to achieve in this setting is $(n-t)$-to-$(n-t)$ gossip: eventually, all but t processes learn all but t rumors.[1] We call this variant: *almost-complete gossip*. This variant is solvable provided $t < C$. (Otherwise, the adversary can disrupt *every* channel in *every* round preventing all communication.)

To study the problem of gossip, we introduce a new combinatorial object which we call a *multiselector*. A *multiselector* is a non-trivial generalization of a combinatorial object known as a *selector* [8, 25]. The selector has proved useful for studying algorithms in a single-channel radio network. Our generalization extends this utility to the multi-channel setting.

We begin by defining the *multiselector* concept precisely, deriving some of its constructions by highlighting relations to selectors and hashing functions [9], and then proving lower and upper bounds on its size. To prove the utility of this object in a multi-channel network, we apply these bounds to the study of the gossip problem. We focus on the two extremal cases that yield the most insight: $C \gg t$ and $C = t + 1$.
Case 1. In Section 4, we assume that $C \gg t$; i.e., the number of available channels is much larger than the number of channels that the adversary can disrupt. We describe an algorithm that runs in linear time.
Case 2. In Section 5, we assume that $C = t + 1$; i.e., the minimum possible number of available channels for which the problem is solvable. We extend the algorithm of case (a) and show that it has a running time $O\left(\left[n^3 + (t+2)^{3(t+1)}\right] \cdot \log \frac{n}{2t+1}\right)$. We then prove that $\Omega(2^{t+1}/\sqrt{t+1})$ rounds are necessary in this setting. (For completeness, we also briefly discuss the intermediate cases between these two extremes.)

In case (1), multiselectors play a key role in coordinating the processes. They allow rumors gathered at a small number of *listener* processes to be efficiently disseminated to the remaining participants. The central difficulty is safely adapting the algorithmic strategy based on the results of previous rounds, without

[1] It would be possible to generalize further and consider s-to-t gossip; however we do not proceed in this direction both for the sake of simplicity, and in order to focus on the case wherein the most insight can be discovered.

engendering disruption from processes that have a different view due to adversarial interference. The multiselector helps ensure that *almost* all the processes have the same view of the current status, bounding the ignorance in the system. In case (2), multiselectors play an additional role in the aggregation of information, allowing processes with an incomplete view of the system state to efficiently coordinate with an unknown set of more informed processes. The multiselectors prove crucial for allowing an efficient solution to the gossip problem. Notice the improvement of our bounds over the best existing solution: an $\Omega((en/t)^{t+1})$ time algorithm [21] that does not attempt to adapt to the execution in progress.

2 Related Work

Selectors. Selectors were originally introduced by Komlos and Greenberg [25]; the term "selector" was coined later by [8] in the context of group property testing. Given a set $S \subseteq P$, we say that a second set S' selects 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 individually selected by some set in the selector. This definition was generalized by [7] to an (n, k, r)-selector, which guarantees that at least r of the elements are individually selected by some sets in the selector.[2] There are a variety of results on the size of differently parameterized selectors; for example, in [25], it was shown that there exist $(n, k, 1)$-selectors of size at most $O(k \log n/k)$, and [23] shows how to explicitly construct k-selectors of size $O(k\mathsf{polylog}(n))$.

Selectors have been used in a variety of contexts to enable communication in single-channel radio networks. For example, in [17, 18], selectors are used to schedule the radio transmissions in such a way that there is sufficiently low contention on the channel; in [15, 16], selector-like structures, called radio-synchronizers, are used to synchronize radio transmissions to efficiently wake-up sleeping devices.

Our notion of a multiselector generalizes a selector in that it *simultaneously* selects a set of elements. Where a selector ensures that some set of elements are individually selected, a multiselector ensures that some set (or sets) of elements are selected at the same time by the same set. This proves useful in a multi-channel radio network, where simultaneous behavior on different channels is a useful tool for increasing throughput or circumventing interference.

Radio Networks. There exists much research in reliable communication on a *multiple-access channel* without malicious interference: initially, in the context of Ethernet networks (c.f., [25,35]), and, later, in the context of radio networks. (For research on broadcast in radio networks, see, for example [1,2,10,28,29,32]; for research on gossip in radio networks, see [3, 11, 13, 14]). Much of the research in this area assumes the devices behave correctly. They focus on the problem of channel contention.

Unreliable **Radio Networks.** Recently, there has been some interest in crash-tolerant communication in radio networks (c.f., [19, 20, 30, 31]). There has also been some work on Byzantine-resilient broadcast in radio networks [4,26]; however, the adversary cannot disrupt the channels. For the setting where adversarial disruption (and corruption) is possible, there exist two common approaches in the literature. The first approach assumes that messages may be corrupted *at random*, according to some known distribution; Pelc and Peleg [34], for example, considered the problem of broadcast in this model. The second approach assumes a worst-case adversary that can corrupt or block messages, but bounds the number of messages for which this is allowed; modelling, for example, a limited energy budget. Koo et al. considered the problem of broadcast, assuming that the adversary's budget is known *a priori* [27]; Gilbert et al. [22] considered a variety of communication problems (including broadcast) for the case where the adversary's budget is unknown.

Unreliable ***Multi-Channel*** **Radio Networks.** The data capacity of multi-channel radio networks is studied studied in [5, 33], but with no malicious behavior. Some existing systems used pre-determined shared "secrets" to perform pseudo-random frequency hopping (e.g., Bluetooth [6]); these techniques can be used

[2] Note that unlike a multiselector, the r elements are not selected simultaneously.

to gossip in the presence of malicious disruption. It is often unreasonable, however, to assume the existence of shared secrets for all possible sets of wireless devices that might one day attempt to communicate.

The present paper, along with [21], are the first, to our knowledge, to consider multi-channel networks subject to malicious disruption in which processes do not possess *a priori* shared secrets. Dolev et al. [21] considered *oblivious* algorithms (that is, algorithms that do not adapt to the execution in progress). They showed for the special case of $t = 1$, that there exists a tight upper and lower bound for gossip of $\Theta(n^2/C^2)$. They extended their upper bound for general t, achieving running time $O((en/t)^{t+1})$. In this paper, we use multiselectors to produce adaptive solutions that outperform the optimal oblivious solutions in [21].

3 Multiselectors

We introduce *multiselectors*, a combinatorial tool that generalizes the idea of *selectors* [8, 25]. We provide upper and lower bounds on their size, which are then useful in establishing our bounds for gossip.

3.1 Definitions

We begin by defining a basic multiselector that selects exactly one set of size k simultaneously:

Definition 1. An **(*n,c,k*)-multiselector**, where $n \geq c \geq k \geq 1$, is a sequence of functions $M_1, M_2, \ldots, M_m$ from $P \to [1, c]$ 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 multiselector has size m. A more general multiselector can be used to select many sets of size k simultaneously; a general multiselector is a generalization of a selector, and also of a multiselector:

Definition 2. A **generalized (*n,c,k,r*)-multiselector**, 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 $P \to [0, c]$ 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.

3.2 Upper Bound

We now show that there exist (n, c, k)-multiselectors and determine their size based on the relationship of k to c. 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)-multiselector of size at most:*

$$\begin{aligned} \text{if } (c = k) \quad &: \quad \frac{ke^c}{\sqrt{2\pi c}} \ln \frac{en}{k} \\ \text{if } (c/2 < k < c) \quad &: \quad ke^k \ln \frac{en}{k} \\ \text{if } (k \leq c/2) \quad &: \quad k2^{2k^2/c} \ln \frac{en}{k} \, . \end{aligned}$$

Proof. We include here the proof for the case where $k \leq c/2$; the other two cases are similar and can be found in the appendix. Let $m = k2^{2k^2/c} \ln \frac{en}{k}$, the bound being proved.

We begin by selecting $M = M_1, \ldots, M_m$ at random, and show that with some probability greater than zero, M is a (n, c, k)-multiselector. For each M_ℓ and for each $i \in P$, choose $M_\ell(i)$ at random from $[1, c]$.

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:

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

Denote this probability by q. Since $k \leq c/2$ we get the following estimate for q:

$$\left(\frac{c-k}{c}\right)^k \geq 4^{-k^2/c}.$$

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. □

We can then conclude that if c is sufficiently larger than k, there are efficient (n, c, k)-multiselectors:

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

The same argument as in Theorem 1 extends to bound the size of generalized multiselectors:

Theorem 3. *For every $n \geq r \geq c \geq k$ where $n \geq 2r$, there exists (n, c, k, r)-multiselectors 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 appendix.

3.3 Multiselectors and Hashing

There exists a connection between good hash functions and multiselectors when $k^2 < c$. In this section, we discuss some of these connections and derive some multiselector constructions.

First, we show how to use a universal family of hash functions to construct a (n, c, k)-multiselector. 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. [9] present such a family of size $\Theta(n^2)$. This family of hash functions is also an (n, c, k)-multiselector, 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 multiselector is of size $O(n^2)$.

Next we consider a technique that produces a more efficient construction. Assume that c is sufficiently large such that there exist $\Theta(k^2 \log n)$ prime numbers less than c. Let $p_1, \ldots, p_{\Theta(k^2 \log n)}$ be a set of $\Theta(k^2 \log n)$ distinct primes less than c. It is easy to see that for any set S of size k, for every pair $x, y \in S$, there are at most $\log n$ primes p_i such that $x = y \mod p_i$: otherwise, the difference $|x - y|$ is divisible by more than $\log n$ primes, implying that $|x - y| > n$, a contradiction. 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)-multiselector of size $O(k^2 \log n)$.

If $k^2 = c$, i.e., there are not a sufficient number of primes $\leq c$, then the two techniques can be combined. Use the second technique to reduce the range to $O(k^2 \log^2 n)$; then use the two-universal hash family of [9] to reduce the range to c. From this we conclude:

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

3.4 Multiselectors and Selectors

It is also possible to construct multiselectors using selectors. The resulting construction is not particularly efficient, but illustrates the connection between selectors and multiselectors. The construction may be reasonable when k is close to C, in which case the exponential behavior is unavoidable (see Theorem 6). We assume (from [12, 24]) that for every k, F_k is a selector of size $f_k = O(k \log n)$.

Our construction proceeds inductively. As the base case, notice that an $(n, c, 1)$-multiselector is trivial, and an $(n, c, 2)$-multiselector is easily instantiated as an 2-selector. Assume that for every $c' \leq c$, $k' < c'$, $M^{(c,\, k')}$ is a (n, c', k')-multiselector of size $m_{c',\, k'} = O(k' \log^{2c} n)$. We construct $M^{c,\, k}$ as follows:

---

for $x = 1$ **to** f_k **do**
  for $y = 1$ **to** $m_{c-1,\, k-1}$ **do**
    let $\ell = xy$
    for every $i \in P$ **do**
      if $F_x(i) = 1$ **then**
        $M_\ell^{c,\, k}(i) \leftarrow c$
      else
        $M_\ell^{c,\, k}(i) \leftarrow M_y^{c-1,\, k-1}(i)$

We prove the following theorem in the appendix:

Theorem 5. *$M^{(c\ ,k)}$ is a (n, c, k)-multiselector of size $O(k^k \log^k n)$.*

3.5 Lower Bound

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

Theorem 6. *Let $M = M_1, \ldots, M_m$ be an (n, c, k)-multiselector where $n \geq 2c$ and $c \geq k$. Then M has size at least:*

$$\begin{aligned}
\text{if } (c = k) \quad &: \quad \frac{2^c}{4\sqrt{2\pi c}} \\
\text{if } (c/2 < k < c) \quad &: \quad e^{k \ln \frac{c}{c-k} - k^2/n} \cdot \frac{\sqrt{n(c-k)}}{4\sqrt{c(n-k)}} \\
\text{if } (k \leq c/2) \quad &: \quad e^{k^2/c - k^2/n} \cdot \frac{\sqrt{n(c-k)}}{4\sqrt{c(n-k)}} \,.
\end{aligned}$$

Proof. We consider here the case where $k = c$; the remaining cases are considered in the appendix. We begin by choosing a subset $S \subseteq P$ of size c at random. We proceed to 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, contradicting the assumption that M is a multiselector.

Fix some particular $\ell \in [1, m]$, and define $S_d = \{i : M_\ell(i) = d\}$, that is, the subset of P that M_ℓ maps to d. In order 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 by Lemma 12.) 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. □

Figure 1: Gossip routine for process p_i.

```
1  Gossip()_i
2    L ← a partition of the set {1, ..., c^2} into c sets of size c.
3    for e = 1 to |E| do
4      knowledgable ←Epoch(L, knowledgable, E[e])_i
5
6    L ← a partition of the set {c^2 + 1, ..., 2c^2} into c sets of size c.
7    for e = 1 to |E| do
8      knowledgable ←Epoch(L, knowledgable, E[e])_i
9
10   // Lastly, do the special epoch which attempts to transmit the final ≤ 5t values.
11   Special-Epoch(knowledgable)_i
```

4 Gossip with Unlimited Channels

We now present an algorithm for solving gossip when $C \gg t$. In order to gossip efficiently, the protocol adaptively chooses the set of processes to transmit in each round based on which processes have already succeeded in gossiping their value in a previous round. This intuition—that carefully adapting to the past is crucial—is supported by the lower bounds in [21], which show that oblivious gossip algorithms cannot be efficient. Adapting to the past proves challenging as processes do not share a uniform view of the current system state. That is, a process does not know whether or not a transmission succeeded unless told by a receiver on the channel. The adversary, however, can block this information, leaving the process ignorant.

Our algorithm helps circumvent this problem by using a $(n, c, t+1)$-multiselector to (efficiently) ensure that *almost* all the processes have the same view of the current status. Processes use the multiselector to guide their channel selection when attempt to receive updates on the system state. Because it guarantees to simultaneously select any subset of size $t + 1$, it follows that for any group of size $t + 1$ processes, there exists a round during which these processes are receiving on different channels. Therefore, at most t total can be kept ignorant by the adversary. This bound on ignorance allows efficient adaptation to continue.

Preliminaries. For the remainder of this section, we fix the constant $c = (5t+1)^2$. Of the C available channels, our algorithm will use exactly c. Recall, we have assumed that n is 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) $n \geq c^2 t + c$.

A note on terminology: We refer to rumors as either *complete* or *incomplete*. Each rumor is initially designated as *incomplete*; when a rumor is received by at least $n - t$ processes, it is designated as *complete*. A process whose rumor is complete is itself considered to have *completed*. Given a set S of integers, we sometimes use the notation $S[k]$, $1 \leq k \leq |S|$, to refer to the k^{th} value in S under some fixed ordering of S.

Gossip. The main routine for the gossip algorithm is given in Figure 1. It proceeds in two sets of *epochs*. In each set of epochs, a set of *listeners* is chosen, and they facilitate the dissemination of incomplete rumors. The listeners' values are not disseminated, however, as they are busy listening; hence each set of epochs chooses a disjoint set of listeners: $\{1, \ldots, c^2\}$ in the first set of epochs, and $\{c^2 + 1, \ldots, 2c^2\}$ in the second set of epochs. After each set of epochs, there are at most $2t$ non-listener rumors that remain incomplete. After the two sets of epochs, at most $4t$ rumors are left incomplete in total. The final call to Special-Epoch reduces this value of incomplete rumors to t, as required.

We define E—used in the Epoch calls—recursively: let $E(1) = \lceil n/c \rceil$; i.e., the n initial rumors are scheduled c per round in the epoch. For all $r > 1$, let $E(r) = \left\lceil \frac{E[r-1](2t)}{c} \right\rceil$; i.e., for each round in the previous epoch, at most $2t$ values are not successfully transmitted, and these values are scheduled c per round

Figure 2: Epoch routine for process p_i.

```
1  Epoch(L, knowledgeable, rnds)_i
2    if knowledgable = true then
3      Let S be the set of processes that are not in L and not completed.
4      Partition S into ⌈|S|/c⌉ sets of size c.
5    for r = 1 to rnds do
6      if (knowledgable = true) and (r ≤ ⌈|S|/c⌉) then
7        if ∃k ∈ {1, ..., c} : i = S[r][k] then schedule i to transmit on channel k.
8        if ∃k ∈ {1, ..., c} : i ∈ L[k] then schedule i to receive on channel k.
9    knowledgable ← Disseminate(L[1], ..., L[c])_i
10   return knowledgable
```

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

An epoch proceeds in two parts: an aggregation and a dissemination phase. In the aggregation phase, rumors are collected at a set of c^2 listeners, c per channel. In the dissemination phase, each set of c listeners broadcasts the rumors that it has received to the other processes. A multiselector is used to derive an efficient sequence of receiving that ensures only a minimal set of receiving processes (i.e., t) can be blocked.

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

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

We now discuss in more detail how both the aggregation and dissemination phases operate in each epoch.

Aggregation. In the first phase (lines $1 - -9$), values are transmitted to the listeners in the set L. Let S be the set of processes that have not yet completed, i.e., their rumors remain unpropagated. 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, incomplete processes 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 processes "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 $2t\lfloor |S|/c \rfloor$ rumors remain incomplete.

Dissemination. In the second part of the epoch, the listeners disseminate the values to the other processes. The pseudocode for disseminate is given in Figure 3. The disseminate routine ensures the following:

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

The disseminate routine consists of two parts. In Part 1, each of the c sets of c listeners attempts to disseminate its set of rumors. For each set, each of the c listeners in the set transmits continually on a unique channel. An $(n, c, t + 1)$-multiselector M is used to schedule the non-listener processes. 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

Figure 3: Disseminate routine for process p_i.

```
1  Disseminate(L[1], ..., L[c])_i
2    let M be a (n, c, t + 1)-multiselector.
3    // Part 1: Ensure that for each listener group, all but some set of t processes receive its value set.
4    knowledgable ← true
5    for k = 1 to c do
6      for each round r = 1 to |M|
7        if ∃j ∈ {1, ..., c} : i = L[k][j] schedule i to transmit on channel j.
8        if i ∉ L[k] then schedule i to receive on channel M_r(i).
9      if i does not receive a message in any of the |M| rounds then knowledgable ← false.
10
11   // Part 2: Ensure that all but some set of t processes receive all the value sets from all the listener groups.
12   L' ← an arbitrary subset of {1, ..., n} of size c(ct + 1).
13   Partition L' into ct + 1 sets L'[1], ..., L'[ct + 1] of size c
14   for each s = 1 to ct + 1 do
15     for each r = 1 to |M| do
16       if ∃j ∈ {1, ..., c} : i = L'[s][j] schedule i to transmit on channel j
17       if i ∉ L'[s] then schedule i to receive on channel M_r(i).
18     if i receives a message in any of the |M| rounds from a node with knowledgable = true then
19       knowledgable ← true
20   return knowledgabe
```

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 processes that do not receive a value from *all* c sets of listeners.

In Part 2, we select a larger set of $c(ct+1)$ processes, 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 processes that received a message from all c sets of listeners in Part 1. All c processes in this set therefore know all the values known to each set of listeners. As in the first part, each of these sets transmits its information to the remaining processes in such a way that at most t processes can fail to learn these values.

Special Epoch. In order to transmit the remaining incomplete rumors, we execute a special epoch. The pseudocode for Special-Epoch is given in Figure 4. The special epoch operates somewhat differently, as there are very few rumors left to transmit. As before, we use listeners to collect the values; we need a set of listeners whose values are already completed. As mentioned, up to $4t$ rumors might be incomplete after the two sets of epochs. An additional t processes might be complete but not aware of this because they were blocked by the adversary in the final epoch. This leaves at most $5t$ processes that are not complete and knowledgeable. We choose a set of $c^2(5t+1)$ possible listeners, and divide them into $5t+1$ sets of size c^2; it is easy to see that at least one of these sets contains only processes that are complete and knowledgeable.

Next, we identify the set of the $\leq 5t$ *special* processes, i.e., those that are either unknowledgeable or incomplete. In the special epoch, we use a $(n, c, 5t)$-multiselector. In some round, this multiselector will assign the $k \leq 5t$ special processes that are actually incomplete each to a different channel to transmit during the same round. Dissemination proceeds as before.

Performance. Each epoch e spends $E(e)$ rounds during the aggregation phase, resulting in $O(n/c)$ rounds of 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 $(5t+1)|M|$

Figure 4: Special Epoch routine for process p_i.

```
1  Special-Epoch(knowledgable)_i
2    let M be an (n, c, 5t)-multiselector.
3    if (knowledgable = false) or (i has not completed) then special ← true else special ←false
4    if knowledgable = true then
5      L ← set of c^2(5t + 1) smallest processes that have completed in a previous epoch.
6      Partition L into (5t + 1) sets L_1, ..., L_{t+1} of size c^2.
7      Partition each L_k into c sets L_k[1], ..., L_k[c] of size c.
8    for s = 1 to 5t + 1 do
9      for r = 1 to |M| do
10       if special = true then schedule i to transmit on channel M_r(i)
11       if ∃k : i ∈ L_s[k] then schedule i to receive on channel k.
12     Disseminate(L_s[1], ..., L_s[c])_i
```

where M is a multiselector 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 rumors are complete. More precisely, the gossip protocol has round complexity $O(n/t^2 + t^5 \log^2 n)$.*

5 Gossip with Limited Channels

We consider here the case where $C = t + 1$: the minimal number of channels for which gossip is possible. We first describe how to adapt the algorithm of Section 4 to this setting. We then present a lower bound (by reduction to a multiselector) showing that the time complexity is inherently exponential in t. Finally, for the sake of completeness, we briefly discuss the intermediate cases where $t + 1 < C < (5t + 1)^2$.

5.1 Algorithm Description

In this section, we modify the gossip routing to use only $C = t + 1$ channels. The disseminate protocol described in Section 4 can be used without modification. We replace, however, Epoch and Special-Epoch with Limited-Epoch (appendix, Figure 5) and Limited-Special-Epoch (appendix, Figure 6), respectively. The main difficulty addressed by these two new routines is the fact that only $t + 1$ processes can transmit concurrently during a given round. It follows that if *any* of these scheduled processes are unknowledgeable and therefore choose not to transmit to avoid contention, then $\leq t$ will attempt to transmit, and the adversary can prevent *any* values from completing in that round.

In order to circumvent this problem, we use a $(n, C, C, 2t + 1)$-generalized-multiselector in the aggregation phase of Limited-Epoch. Processes know at the beginning of a round if they are scheduled or if they are unknowledgeable. Such processes will attempt to transmit according to the schedule described by the generalized multiselector.[3] The multiselector guarantees that one of its partitions will simultaneously select the $t + 1$ processes 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. Thus, we modify the definition of E: for $r > 1$, $E(r) = \lceil \frac{E(r-1)t}{C} \rceil$.

When Gossip calls Limited-Special-Epoch, there are at most $3t$ incomplete processes—t from each set of epochs, and as many as t additional processes that finished the final epoch in an unknowledgeable state

[3]For simplicity, in the pseudocode, when we say a process transmits on channel $M_r(i)$, if $M_r(i)$ maps to 0—which is possible with a generalized multiselector—this is equivalent to being scheduled *not* to transmit.

and thus act as if they have not completed. In Limited-Special-Epoch, processes that are not complete or are unknowledgeable are labeled *special*. A $(n, C, C, 3t)$-generalized-multiselector is used to ensure that all subsets of size $t+1$ of these up to $3t$ special processes get an opportunity to transmit concurrently. It follows that at most t are blocked from transmitting. As before, we attempt many different sets of listeners to make sure at least one is comprised of c^2 processes that are complete and knowledgeable. For this set, the Disseminate call will work as before, spreading $n-t$ rumors to $n-t$ processes.

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 thus conclude:

Theorem 10. *When $C = t+1$, the gossip protocol terminates in $O\left(\left[n^3 + (t+2)^{3(t+1)}\right] \cdot \log \frac{n}{2t+1}\right)$, or more specifically, $O\left(t(c+1)^{2t+1}\log \frac{n}{2t+1}\left[n + t(c+1)^{t-1}\right]\right)$ rounds.*

5.2 Lower Bound

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

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

Proof. Consider a protocol that solves almost-gossip in m rounds for all executions. We use this protocol to construct a $(n, C, t+1)$-multiselector of length m, and then invoke Theorem 6 to conclude the proof. We construct the multiselector by simulating the gossip protocol in the following way, for each round:

- Every process 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 process that is scheduled to transmit on some channel is simulated as if it transmits its message.

For each round r, we construct a function M_r of the $(n, C, t+1)$-multiselector as follows: if a process i listens on channel k, then $M_r(i) \leftarrow k$; otherwise, if process 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 $(n, C, t+1)$-multiselector: 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 construct a real execution in which no element in S ever receives a message. Assume that prior to round r, no process in S has received any message. We can conclude that in round r, the processes in S behave according to the simulation used in constructing M. Thus the elements in S listen on no more than t channels. (Some may also transmit in that round; some may do nothing; in any case, they learn nothing.) The adversary blocks exactly these $\leq t$ channels, maintaining the invariant that no process in S has received a message. This is a contradiction. Since M is a $(n, C, t+1)$-multiselector, the result follows from Theorem 6 with $C = t+1$. □

5.3 Gossip with Other Bounded Numbers of Channels

We have discussed the case where C is unlimited and where it was minimal. For completeness, we briefly addresses the intermediate possible values. When $C < 2t+1$, the aggregation phase requires the use of generalized multiselectors 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 algorithm described in Section 4 for unlimited channels, where the multiselectors used by the dissemination routine are sized appropriately; as C grows the running time decreases, as the more available channels reduces the size of the multiselectors used in Disseminate and Special-Epoch. For $2t+1 < C < 5t+1$, we must use a hybrid algorithm in which Disseminate stays the same, but Special-Epoch must use generalized multiselectors as in Limited-Special-Epoch. It is straightforward (but tedious) to calculate the associated running times.

References

[1] Noga Alon, Amotz Bar-Noy, Nathan Linial, and David Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43(2):290–298, October 1992.

[2] 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. *Journal of Computer and System Sciences*, 45(1):104–126, 1992.

[3] R Bar-Yehuda, A Israeli, and A Itai. Multiple communication in multi-hop radio networks. *SIAM Journal on Computing*, 22(4):875–887, 1993.

[4] Vartika Bhandari and Nitin H. Vaidya. On reliable broadcast in a radio network. In *The Proceedings of the International Symposium on Principles of Distributed Computing*, pages 138–147, 2005.

[5] Vartika Bhandari and Nitin H. Vaidya. Connectivity and capacity of multi-channel wireless networks with channel switching constraints. Technical report, University of Illinois at Urbana-Champaign, January 2007.

[6] Bluetooth Consortium. *Bluetooth Specification Version 2.1*, July 2007. Available at `http://www.bluetooth.com/NR/rdonlyres/F8E8276A-3898-4EC6-B7DA-E5535258B056/6545/Core_V21_EDR.zip`.

[7] A. De Bonis, L. Gasieniec, and U. Vaccaro. Generalized framework for selectors with applications in optimal group testing. In *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 81–96, 2003.

[8] Annalisa De Bonis, Leszek Gasieniec, and Ugo Vaccaro. Optimal two-stage algorithms for group testing problems. *SIAM Journal on Computing*, 34(5):1253–1270, 2005.

[9] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions (extended abstract). In *Proceedings of the ninth annual ACM symposium on Theory of computing*, pages 106–112, New York, NY, USA, 1977. ACM.

[10] Bogdan S. Chlebus, Leszek Gasieniec, Alan Gibbons, Andrzej Pelc, and Wojciech Rytter. Deterministic broadcasting in unknown radio networks. In *SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 861–870, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[11] Bogdan S. Chlebus, Leszek Gasieniec, Andrzej Lingas, and Aris T. Pagourtzis. Oblivious gossiping in ad-hoc radio networks. In *DIALM '01: Proceedings of the 5th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 44–51, New York, NY, USA, 2001. ACM Press.

[12] Bogdan S. Chlebus and Dariusz R. Kowalski. Almost optimal explicit selectors. In *Fundamentals of Computation Theory*, pages 270–280, 2005.

[13] B.S. Chlebus and D.R. Kowalski. Gossiping to reach consensus. In *The Proceedings of the Symposium on Parallel Algorithms and Architectures*, 2002.

[14] B.S. Chlebus and D.R. Kowalski. Robust gossiping with an application to consensus. *Journal of Computer and System Sciences*, 72:1262–1281, 2006.

[15] M. Chrobak, L. Gasieniec, and D. R. Kowalski. The wake-up problem in multi-hop radio networks. In *SODA*, 2004.

[16] M. Chrobak, L. Gasieniec, and D. R. Kowalski. The wake-up problem in multi-hop radio networks. *SIAM Journal of Computing*, 36(5):1453–1471, 2007.

[17] M. Chrobak, L. Gasieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. In *The Proceedings of the Symposium on Foundations of Computer Science*, 2000.

[18] M. Chrobak, L. Gasieniec, and W. Rytter. Fast broadcasting and gossiping in radio networks. *Algorithms*, 43(2):575–581, 2002.

[19] A. Clementi, A. Monti, and R. Silvestri. Optimal f-reliable protocols for the do-all problem on single-hop wireless networks. In *Algorithms and Computation*, pages 320–331, 2002.

[20] A. Clementi, A. Monti, and R. Silvestri. Round robin is optimal for fault-tolerant broadcasting on wireless networks. *Journal of Parallel and Distributed Computing*, 64(1):89–96, 2004.

[21] Shlomi Dolev, Seth Gilbert, Rachid Guerraoui, and Calvin Newport. Gossiping in a multi-channel radio network: An oblivious approach to coping with malicious interference. In *The Proceedings of the International Symposium on Distributed Computing*, 2007.

[22] Seth Gilbert, Rachid Guerraoui, and Calvin Newport. Of malicious motes and suspicious sensors: On the efficiency of malicious interference in wireless networks. In *The Proceedings of the International Conference on Principles of Distributed Systems*, December 2006.

[23] Piotr Indyk. Explicit constructions of selectors and related combinatorial structures, with applications. In *Proc. of SODA*, 2002.

[24] W. Kautz and R. Singleton. Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory*, 10:363–377, Oct. 1964.

[25] J. Komlos and A.G. Greenberg. An asymptotically fast non-adaptive algorithm for conflict resolution in multiple access channels. *IEEE Trans. Inf. Theory*, March 1985.

[26] C-Y. Koo. Broadcast in radio networks tolerating byzantine adversarial behavior. In *The Proceedings of the International Symposium on Principles of Distributed Computing*, pages 275–282, 2004.

[27] Chiu-Yuen Koo, Vartika Bhandari, Jonathan Katz, and Nitin H. Vaidya. Reliable broadcast in radio networks: The bounded collision case. In *The Proceedings of the International Symposium on Principles of Distributed Computing*, 2006.

[28] D. Kowalski and A. Pelc. Time of deterministic broadcasting in radio networks with local knowledge. *SIAM Journal on Computing*, 33(4):870–891, 2004.

[29] Dariusz R. Kowalski and Andrzej Pelc. Deterministic broadcasting time in radio networks of unknown topology. In *Foundations of Computer Science*, pages 63–72, 2002.

[30] E. Kranakis, D. Krizanc, and A. Pelc. Fault-tolerant broadcasting in radio networks. In *The Proceedings of the Annual European Symposium on Algorithms*, pages 283–294, 1998.

[31] Evangelos Kranakis, Danny Krizanc, and Andrzej Pelc. Fault-tolerant broadcasting in radio networks. *Journal of Algorithms*, 39(1):47–67, April 2001.

[32] E. Kushelevitz and Y. Mansour. An omega(d log(n/d)) lower bound for broadcast in radio networks. In *The Proceedings of the International Symposium on Principles of Distributed Computing*, 1993.

[33] Pradeep Kyasanur and Nitin H. Vaidya. Capacity of multi-channel wireless networks: Impact of number of channels and interfaces. In *Proc. of Mobicom*, August 2005.

[34] Andrzej Pelc and David Peleg. Feasibility and complexity of broadcasting with random transmission failures. In *The Proceedings of the International Symposium on Principles of Distributed Computing*, pages 334–341, 2005.

[35] D. E. Willard. Log-logarithmic selection resolution protocols in a multiple access channel. *SIAM Journal of Computing*, 15(2):468–477, 1986.

A Appendix

Figure 5: Epoch routine for process p_i where $C = t + 1$.

```
1  Limited-Epoch(L, knowledgeable, rnds)_i
2    let M be a (n, C, C, 2t + 1)-generalized-multiselector.
3    S ← ∅
4    if knowledgable = true then
5      Let S be the set of processes that are not in L and not completed.
6      Partition S into ⌈|S|/c⌉ sets of size C.
7    for r_1 = 1 to rnds do
8      if (r_1 ≤ ⌈|S|/C⌉) then
9        for r_2 = 1 to |M| do
10         if i ∉ L and ((i is not knowledgable) or (i ∈ S[r_1])) then schedule i to transmit on channel M_{r_2}(i).
11         if ∃k ∈ {1, ..., C} : i ∈ L[k] then schedule i to receive on channel k.
12   knowledgable ←Disseminate(L[1], ..., L[C])_i
13   return knowledgable
```

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

```
1  Limited-Special-Epoch(knowledgable)_i
2    Let M be an (n, C, C, 3t)-multiselector
3    special ← false
4    if (knowledgable = false) or (i has not completed) then special ← true
5    if knowledgable = true then
6      L ← set of c^2(3t + 1) smallest processes that have completed in a previous epoch.
7      Partition L into (3t + 1) sets L_1, ..., L_{t+1} of size c^2.
8      Partition each L_k into c sets L_k[1], ..., L_k[c] of size c.
9    for s = 1 to 3t + 1 do
10     for r = 1 to |M| do
11       if special = true then schedule i to transmit on channel M_r(i)
12       if ∃k : i ∈ L_s[k] then schedule i to receive on channel k.
13     Disseminate(L_s[1], ..., L_s[c])_i
```

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

$$\begin{aligned} \text{if } (c = k) &: \frac{ke^c}{\sqrt{2\pi c}} \ln \frac{en}{k} \\ \text{if } (c/2 < k < c) &: ke^k \ln \frac{en}{k} \\ \text{if } (k \leq c/2) &: k2^{2k^2/c} \ln \frac{en}{k} \,. \end{aligned}$$

Proof. Let $m_1 = \frac{ke^c}{\sqrt{2\pi c}} \ln \frac{en}{k}$, $m_2 = ke^k \ln \frac{en}{k}$, and $m_3 = k2^{2k^2/c} \ln \frac{en}{k}$, the three different bounds.

We begin by selecting $M = M_1, \ldots, M_m$ at random, and show that with some probability greater than zero, M is a (n, c, k)-multiselector. For each M_ℓ and for each $i \in P$, choose $M_\ell(i)$ at random from $[1, c]$.

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:

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

Let denote this probability by q. We consider three cases. For $k = c$, q is at least $\sqrt{2\pi c}e^{-c}$, by Stirling inequality. For $c/2 < k < c$ we again use Stirling inequality to get the lower estimate

$$\frac{\sqrt{2\pi c}c^c}{2\sqrt{2\pi(c-k)}(c-k)^{c-k}e^k c^k} \geq e^{-k}$$

for q. In the remaining case $k \leq c/2$ we get the following lower estimate for q:

$$\left(\frac{c-k}{c}\right)^k \geq 4^{-k^2/c} \,.$$

Denote the lower estimates on q obtained in the above cases by q_1, q_2, q_3, respectively. The probability that S is not well-mapped for all M_ℓ is at most $(1 - q_j)^{m_j}$, for $j = 1, 2, 3$ depending on the case. Since in all cases $m_j = q_j^{-1} \cdot k \ln \frac{en}{k}$, the probability that S is not well-mapped for all M_ℓ is always 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 in turn smaller than 1, implying the desired conclusion. □

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

Proof. Fix $m = 2r\frac{(c+1)^r}{(c-k)^k}\log(en/r)$. We start with selecting $M = M_1, \ldots, M_m$ at random, and show that with a positive probability, M is a (n, c, k, r)-multiselector. For each M_ℓ and for each $i \in P$, choose $M_\ell(i)$ at random from $[0, c]$.

Fix sets S, S' where $S' \subseteq S \subseteq P$, $|S| = r$, and $|S'| = k$. Consider a particular M_ℓ. We calculate the probability that each element of S is assigned a unique element in $[1, c]$, and each element in $S \setminus S'$ is mapped to 0. Since there are $\binom{c}{k}k!$ good mappings from k elements to $[1, c]$, and $(c+1)^r$ total mappings of r elements in S to $[0, c]$ sets, using Stirling bounds we conclude that:

$$\Pr\{S \text{ is uniquely mapped by } M_\ell\} = \frac{\binom{c}{k}k!}{(c+1)^r} \geq \frac{(c-k)^k}{(c+1)^r}\ .$$

Alternatively, with a simpler approximation, we could conclude:

$$\Pr\{S \text{ is uniquely mapped by } M_\ell\} = \frac{\binom{c}{k}k!}{(c+1)^r} \geq \frac{k^k}{e^k(c+1)^r}\ .$$

Denote by q the probability that S is uniquely mapped by M_ℓ (it is the same for all $\ell \leq m$). The probability that S is not well-mapped for *all* M_ℓ is at most $(1-q)^m$. Since $m = q^{-1} \cdot 2r\ln\frac{n}{r}$, the probability that S is not well-mapped for all M_ℓ is at most $e^{-2r\ln\frac{en}{r}} \leq \left(\frac{r}{en}\right)^{2r}$. Since there are only $\binom{n}{r} \cdot \binom{n}{k} < \left(\frac{en}{r}\right)^{2r}$ possible subsets S, S' of size r, k, respectively, we argue (by a union bound) that the probability of some S being incorrectly mapped by all M_ℓ is at most $\binom{n}{r} \cdot \binom{n}{k} \cdot \left(\frac{r}{en}\right)^{2r}$, which is in turn smaller than 1, implying the desired conclusion. □

Theorem 6. *Let $M = M_1, \ldots, M_m$ be an (n, c, k)-multiselector where $n \geq 2c$ and $c \geq k$. Then M has size at least:*

$$\begin{aligned} \text{if } (c = k) \quad &: \quad \frac{2^c}{4\sqrt{2\pi c}} \\ \text{if } (c/2 < k < c) \quad &: \quad e^{k \ln \frac{c}{c-k} - k^2/n} \cdot \frac{\sqrt{n(c-k)}}{4\sqrt{c(n-k)}} \\ \text{if } (k \leq c/2) \quad &: \quad e^{k^2/c - k^2/n} \cdot \frac{\sqrt{n(c-k)}}{4\sqrt{c(n-k)}} \,. \end{aligned}$$

Proof. Fix some particular $\ell \in [1, m]$, and define $S_d = \{i : M_\ell(i) = d\}$, that is, the subset of P that M_ℓ maps to d. In order 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: $\binom{c}{k} \prod_{d=1}^{k} |S_d| \leq \binom{c}{k} (n/c)^k$. (The inequality follows by Lemma 12 in the appendix.) Since there are $\binom{n}{k}$ sets of size k, and since $(n - c) \geq n/2$, we conclude (via Stirling's approximation) that the probability q that S is correctly mapped by M_ℓ is at most

$$\frac{\binom{c}{k}(n/c)^k}{\binom{n}{k}} \leq \frac{c^c n^k (n-k)^{n-k}}{n^n c^k (c-k)^{c-k}} \cdot \frac{2\sqrt{2\pi c} \cdot 2\sqrt{2\pi(n-k)}}{\sqrt{2\pi n} \cdot \sqrt{2\pi(c-k)}} = \frac{c^{c-k}(n-k)^{n-k}}{(c-k)^{c-k} n^{n-k}} \cdot \frac{4\sqrt{c(n-k)}}{\sqrt{n(c-k)}}$$

$$= \frac{\frac{c^c}{(c-k)^c}}{\frac{n^n}{(n-k)^n}} \cdot \frac{\left(\frac{c-k}{c}\right)^k}{\left(\frac{n-k}{n}\right)^k} \cdot \frac{4\sqrt{c(n-k)}}{\sqrt{n(c-k)}} < \frac{\left(\frac{c-k}{c}\right)^k}{\left(\frac{n-k}{n}\right)^k} \cdot \frac{4\sqrt{c(n-k)}}{\sqrt{n(c-k)}} \,,$$

where the last inequality follows from the fact that the function $f(x) = \frac{x^x}{(x-a)^x}$ decreases while $x > a$ goes to the infinity. Thus for $k \leq c/2$ we get the upper bound

$$e^{-k^2/c + k^2/n} \cdot \frac{4\sqrt{c(n-k)}}{\sqrt{n(c-k)}} \,,$$

and for $c/2 < k < c$ we obtain

$$e^{-k \ln \frac{c}{c-k} + k^2/n} \cdot \frac{4\sqrt{c(n-k)}}{\sqrt{n(c-k)}} \,.$$

Thus, the probability that S is correctly mapped by *any* of the m functions is at most $m \cdot q$ (by a union bound). In both cases it is smaller than 1, therefore with positive probability the set S is not correctly mapped by any of the M_ℓ, resulting in a contradiction. □

Theorem 5. *$M^{(c\ ,k)}$ is a (n, c, k)-multiselector of size $O(k^k \log^k n)$.*

Proof. Choose some set S of size k. By definition, there is some x such that S_x selects some i from S. Let $\widehat{S} = S - \{i\}$. By definition there is some y such that $M_y^{c-1\ ,k-1}$ correctly maps $\widehat{S}$ to channels in the range $[1, c-1]$. $M_{xy}^{c,\ k}$ thus satisfies the requisite conditions. The size of $M^{c,\ k}$ follows immediately by construction. □

Lemma 12. *Let $a_1, \ldots, a_\ell$ be a sequence of positive integers, and let $\overline{a}$ be the average of the a_i. Then:*

$$\prod_{j=1}^{\ell} a_j \leq \overline{a}^{\ell}$$

Proof. We show that for any $a_1, \ldots, a_\ell$ that have average $\overline{a}$, their product is maximized when each $a_i = \overline{a}$. We use Lagrange multipliers. We maximize the following function:

$$f(a_1, \ldots, a_n) = \prod_{i=1}^{n} a_i + \lambda \left(\sum_{i=1}^{\ell} a_i - \ell\overline{a}) \right) .$$

We then take the derivative with respect to a_k, and set the result equal to zero.

$$\frac{\partial f}{\partial a_k} = \frac{\prod_{i=1}^{\ell} a_i}{a_k} + \lambda = 0 \, .$$

Solving for a_k, we conclude that for all k:

$$a_k = \frac{\prod_{i=1}^{\ell} a_i}{-\lambda} \, .$$

This implies that the maximum is achieved when all the a_k are equal, leading to the conclusion that they are each equal to $\overline{a}$. □