The Wireless Synchronization Problem*

Shlomi Dolev
Ben-Gurion University
Beer-Sheva, Israel
dolev@cs.bgu.ac.il

Seth Gilbert
EPFL IC
Lausanne, Switzerland
seth.gilbert@epfl.ch

Rachid Guerraoui
EPFL IC
Lausanne, Switzerland
rachid.guerraoui@epfl.ch

Fabian Kuhn
MIT CSAIL
Cambridge, MA, USA
fkuhn@csail.mit.edu

Calvin Newport
MIT CSAIL
Cambridge, MA, USA
cnewport@csail.mit.edu

ABSTRACT

In this paper, we study the *wireless synchronization problem* which requires devices activated at different times on a congested single-hop radio network to synchronize their round numbering. We assume a collection of n synchronous devices with access to a shared band of the radio spectrum, divided into $\mathcal{F}$ narrowband frequencies. We assume that the communication medium suffers from unpredictable, perhaps even malicious interference, which we model by an adversary that can disrupt up to t frequencies per round. Devices begin executing in different rounds and the exact number of participants is not known in advance.

We first prove a lower bound, demonstrating that at least $\Omega\left(\frac{\log^2 n}{(\mathcal{F}-t)\log\log n} + \frac{\mathcal{F}t}{\mathcal{F}-t}\log n\right)$ rounds are needed to synchronize. We then describe two algorithms. The first algorithm almost matches the lower bound, yielding a running time of $O\left(\frac{\mathcal{F}}{\mathcal{F}-t}\log^2 n + \frac{\mathcal{F}t}{\mathcal{F}-t}\log n\right)$ rounds. The second algorithm is *adaptive*, terminating in $O\left(t'\log^3 n\right)$ rounds in *good* executions, that is, when the devices begin executing at the same time, and there are never more than t' frequencies disrupted in any given round, for some $t' < t$. In all executions, even those that are not good, it terminates in $O\left(\mathcal{F}\log^3 n\right)$ rounds.

*This work has been support in part by Cisco-Lehman CUNY A New MAC-Layer Paradigm for Mobile Ad-Hoc Networks, AFOSR Award Number FA9550-08-1-0159, NSF Award Number CCF-0726514, NSF Award Number CNS-0715397, the Rita Altura trust chair in computer science, and the US Air Force, EU ICT-FET under grant 215270 "Foundations of Adaptive Networked Societies of Tiny Artefacts (FRONTS)."

PODC'09, August 10–12, 2009, Calgary, Alberta, Canada.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Networks

General Terms

Algorithms, Theory

1. INTRODUCTION

Synchronization is a fundamental problem in distributed systems: a set of (possibly unreliable) machines, communicating over a (possibly unreliable) network, attempt to establish a shared temporal frame-of-reference. Such synchronization is critical for a wide variety of protocols and applications. In this paper, we consider the problem of synchronization in wireless networks. In fact, a large class of wireless algorithms operate under the assumption that synchronization has already been achieved: time is typically divided into uniform *rounds* and all the devices start in the same round. In practice, however, this is rarely the case.

The Challenges of Radio Network Synchronization.

A major challenge in wireless networks is overcoming the idiosyncratic nature of radio communication. Typically, algorithm designers assume that their protocol has exclusive access to single communication frequency on which messages can be reliably sent and received. In reality, however, an increasing amount of wireless networking occurs on the unlicensed bands of the radio spectrum. For example, the 2.4 Ghz band is used by 802.11, Bluetooth, Zigbee, cordless phones, and a growing number of proprietary devices. These bands are typically divided into independent narrowband communication frequencies; 802.11, for example, divides the 2.4 Ghz band into roughly 12 frequencies, while Bluetooth divides it into roughly 75.

A device operating in this setting must tolerate a (potentially) significant amount of unpredictable disruption. This disruption may be caused by unrelated devices running unrelated protocols; or it may be caused by nearby electronic appliances that induce electromagnetic noise. (Microwaves, for example, are notorious for causing serious disruption on the 2.4 Ghz band.) Recent studies have shown this interference to be both prevalent and harmful [20]. In addition, since the airwaves are open, disruption may also be caused

by a malicious attacker—for example, a malcontent with a signal jammer attempting to block a Starbucks base station. Distributed computing in this *age of open airwaves* is a decidedly non-trivial affair.

The Disrupted Radio Network Model.

The *disrupted radio network model* captures the core characteristics of this increasingly relevant setting. (We introduced this model in [19, 16, 15]. It has since been adopted and studied by other researchers [30, 31, 29].) We consider a single-hop radio network comprised of $\mathcal{F}$ independent communication frequencies. In each round, each device can choose one frequency on which to participate. We incarnate the diversity of possible disruption sources with a single adversary that can choose up to $t < \mathcal{F}$ frequencies per round to disrupt, preventing communication, where t is a known bound. This model is simple enough to produce strong theoretical bounds, and yet still realistic enough that the results are relevant to real world networking on shared channels.

Wireless Synchronization.

We focus on the problem of synchronizing wireless devices in a disrupted radio network. We assume that the devices come together in an *ad hoc* manner: they are activated at different times, and only a loose (arbitrarily bad) upper bound on the total number of participants is known in advance. The goal of the *wireless synchronization problem* is to establish a a global round numbering that is shared among all participants.

The resulting synchronized rounds are a crucial building block for coordinating wireless devices. For example, consider Bluetooth-style protocols that use pseudorandom frequency hopping to avoid interference; a common round numbering is needed to coordinate the choice of frequency in each round. A common round numbering also allows protocols to periodically run *initialization* and *maintenance* protocols (say, in every round r such that $r \ mod \ k = 0$); these protocols might count the currently participating devices, assign unique names, allocate a TDMA schedule, establish a group key (see, e.g., [16]), or elect a leader. (This would allow, for example, Bluetooth to operate without the need for a user to manually designate one device as the *master*.) Almost all existing protocols for such problems assume that participating devices begin the protocol at the same time. Thus, given a shared global round numbering, we can transform a variety of existing results into more robust protocols that can tolerate a more realistic ad hoc setting.

Our Results.

We begin by proving a lower bound on the efficiency of synchronization: any *regular* synchronization protocol requires at least $\Omega\left(\frac{\log^2 N}{(\mathcal{F}-t)\log\log N} + \frac{\mathcal{F}t}{\mathcal{F}-t}\log N\right)$ rounds for some device to synchronize, where N is the upperbound on participants. (A regular protocol is one in which devices behave in a uniform fashion prior to receiving their first message. Both algorithms in this paper are regular.) The first term derives from a non-trivial generalization of the argument used by [22] in analyzing the *wake-up problem*. The second term comes from demonstrating an adversarial strategy for keeping the probability of coordination low.

We continue by describing the Trapdoor Protocol, an algorithm that provides an (almost tight) solution to the problem of wireless synchronization. Every device completes the synchronization protocol within $O\left(\frac{\mathcal{F}}{\mathcal{F}-t}\log^2 N + \frac{\mathcal{F}t}{\mathcal{F}-t}\log N\right)$ rounds (with high probability). The Trapdoor Protocol runs a competition among the participants, where the winner gets to determine the round numbering scheme that is adopted by all the other participants. (The losers fall through the "trapdoor.")

We then present a variant of the Trapdoor Protocol, known as the Good Samaritan Protocol, that is optimistic and adaptive: when all the devices begin in the same round, its running time depends only on the number of frequencies actually disrupted (rather than on the worst-case possible disruption). Specifically, if all devices begin at the same time and the adversary disrupts at most $t' < t$ of the $\mathcal{F}$ available frequencies per round, then every device is synchronized within $O(t'\log^3 N)$ rounds. In executions that do not satisfy these optimistic assumptions, the protocol terminates within $O(\mathcal{F}\log^3 N)$ rounds.

Two key challenges arise in designing an adaptive protocol. First, when only a small number of frequencies are disrupted, the protocol must complete quickly; this limits the devices to using a small number of frequencies. The adversary, however, may be able to block all of these frequencies, making it hard for a device to determine whether it has succeeded. (Recall, a device does not know how many other participants, if any, have been activated.) To solve this problem, we designate some of the participants as *good samaritans*, whose role is to help the contenders determine whether or not they have won the competition. The second challenge is dealing with newly arrived devices. Because these devices must focus their attention on a small number of frequencies (in order to complete quickly), there may be a disproportionate amount of interference on these frequencies, which may delay other devices. Coordinating the optimistic and pessimistic portion of the protocols in order to ensure that at most one contender wins the competition requires carefully managing of the newly arrived devices to prevent too much contention from building up.

2. MODEL

As in [19, 16, 15], we assume a single-hop radio network. Time is divided into synchronized slots, called *rounds*. We assume N devices—called *nodes*—that begin the execution *inactive*. At the beginning of each round, an adversary chooses which, if any, of the *inactive* nodes to *activate*. Devices have no *a priori* knowledge of the global round number: when a node is activated, it considers the current round to be the first. Nodes do not know in advance the total number of nodes $n < N$ that will eventually be activated.

The radio network consists of $\mathcal{F} \geq 1$ disjoint narrowband communication frequencies, where $N \geq \mathcal{F}$. In each round, each node chooses a single frequency on which to participate, and chooses whether to *broadcast* or *receive*. (It receives no information from other frequencies.) If two or more nodes broadcast on the same frequency, then the receivers on that frequency receive nothing, due to collision.

We assume an *interference adversary* that can disrupt up to $t < \mathcal{F}$ frequencies per round, where t is a known upper bound. By disrupting a frequency, the adversary prevents any node from receiving a message on that frequency. A node receives a message on a frequency f only if exactly one node broadcasts on f, and the adversary does not disrupt

f. The adversary chooses its behavior for round r based only on knowledge of the protocol being executed and the completed execution up to the end of round $r-1$. As previously stated, the adversary incarnates the diversity of unpredictable sources of interference that might occur on the increasingly crowded unlicensed bands. It does not *necessarily* represent a literal adversarial device.

3. WIRELESS SYNCHRONIZATION

Wireless synchronization is achieved when the activated nodes share a consistent round numbering scheme. There are five requirements:

1. *Validity:* In every round, every activated node outputs a value in $\mathbb{N}_\perp = \mathbb{N} \cup \{\perp\}$. If a node outputs a number, then we consider that to be the round number; if a not outputs $\perp$, then it has not yet determined a round number.
2. *Synch Commit:* Once a node outputs a non-$\perp$ value (in $\mathbb{N}$), it never again outputs $\perp$.
3. *Correctness:* The round number increments in each round: if a node outputs i in round r, then it outputs $i+1$ in round $r+1$.
4. *Agreement:* In every round, all non-$\perp$ outputs are the same, with high probability.
5. *Liveness:* Eventually, every *active* node stops outputting $\perp$, with probability 1.

The *synch commit* property ensures that each node knows when it has successfully synchronized. Once synchronization has been achieved, the round number continues to increment (as per *correctness*). These guarantees ensure that a synchronization routine can safely be used as a building block for a protocol that depends on round numbers. We say that an algorithm solves the wireless synchronization problem in time T if and only if liveness is achieved by round T, with high probability.

4. RELATED WORK

A substantial fraction of theoretical work on radio networks has focused on the problem of broadcast; c.f., [4, 5, 26, 24, 25, 27, 28, 3, 7, 10, 9, 13, 14]. Wireless synchronization, however, is more closely related to the wake-up problem, in which active devices attempt to awaken inactive devices with a successful broadcast (e.g., [18, 22, 8, 11]). (The wake-up problem is also related to *selectors*, introduced by Komlos and Greenberg [23], and widely used in the context of radio networks, e.g., [6, 21, 12].)

The wake-up problem is similar to wireless synchronization in that it requires coordination in a model with asynchronous activation and (typically) local round counters. By contrast, however, in this paper we assume multiple communication frequencies and adversarial interference. Also, we do not assume that a successful broadcast activates all inactive nodes (an assumption that is sometimes hard to justify in practice); a node in our model can only be synchronized after it is activated by the adversary. In some sense, the wireless synchronization problem is a generalization of the wake-up problem for more frequencies, in that requiring at least one node to broadcast alone and without disruption is necessary, but not always sufficient to solve our problem.

In the systems community, the effort to cope with crowded radio bands is dominated by the cognitive radio/network community (see [2] for an overview). This strategy uses advanced software radios to detect disruption, and applies spectrum-sharing rules to make optimal use of the available bandwidth. These solutions require specialized hardware, the ability to detect interference, and the assumption that interference is non-dynamic. Solutions in our model apply to commodity hardware (that cannot necessarily detect interference) and makes no assumptions about the behavior of other devices or electromagnetic interference.

We introduced the disrupted radio model in [15], where we studied oblivious gossip protocols, assuming that every device is activated in the same round. In [19, 16], we examined adaptive gossip protocols (both deterministic and randomized) in the same model, as well as related problems such as shared key agreement. In [16], we assumed a more malicious adversary that could also inject spoofed messages in the network. Strasser et al. adopted our model to study practical key agreement protocols [30, 31]. In a recent paper, soon to appear, Meier et al. studied the problem of coordinating a pair of nodes in a variant of our model where t is unknown [29].

5. LOWER BOUNDS

We begin by proving two lower bounds. The second bound applies to any protocol, while the first restricts itself to *regular* protocols. A protocol is regular if there exists a fixed sequence of pairs $(F_1, b_1), (F_2, b_2)...$, where each F_i is a probability distribution over frequencies and each b_i is a probability, such that for each node u and local round r, if u has not received a message through $r-1$, it chooses its frequency and whether or not to broadcast according to F_r and b_r, respectively. In other words, nodes behave in a uniform manner until they first receive a message. Both protocols considered in this paper are regular.

THEOREM 1. *Let $\mathcal{P}$ be a regular protocol that solves the wireless synchronization problem for some $\mathcal{F}$, t, and N, $\mathcal{F} > t$, with probability at least $1-1/N$. There exists an execution of $\mathcal{P}$ in which some node requires $\Omega\left(\frac{\log^2(N)}{(\mathcal{F}-t)\log\log(N)}\right)$ rounds to synchronize.*

Our bound is a generalization of the $\Omega\left(\frac{\log n \log(1/\epsilon)}{\log\log n + \log\log(1/\epsilon)}\right)$ bound for solving the wake-up problem with probability $1-\epsilon$, by Jurdzinski and Stachowiak [22][1].

We begin with a lemma regarding a simple randomized process in which m balls are thrown independently into $s+1$ bins according to a specified distribution:

LEMMA 2. *Assume that there are $m \geq 0$ balls and $s+1 \geq 1$ bins and a probability distribution $p_1 \leq p_2 \leq \cdots \leq p_{s+1}$ over the bins such that every ball independently lands in a bin according to the given distribution. Furthermore, assume that $p_{s+1} \geq 1/2$. Then, the probability that no bin receives exactly one ball is at least 2^{-s}.*

We continue with the proof of the main theorem.

[1]Farach-Colton et al. [17] improved the bound to $\Omega(\log n \log(1/\epsilon))$ using a linear program-based technique. We conjecture that the same general technique would generate a similar improvement for our bound—that is, would remove the $\log\log N$ factor.

PROOF (THEOREM 1). We bound the number of rounds required for some node to succeed in broadcasting alone on some frequency. Any solution to the wireless synchronization problem must guarantee that this event occurs. We restrict our attention to values of $n \geq n_{min} = \Omega(\log^4 N)$, and consider only a weak adversary that activates all n nodes during the same round and disrupts frequencies 1 to t in every round. Because we assume a bound N on n is known to the nodes, we define error probability $\varepsilon = 1/N$.

For each round r such that no node has yet received a message, we define *broadcast probability* p_f, for each frequency $f > t$, to be the probability that any given node chooses f and broadcasts during this round. (A single such probability exists because we have restricted our attention to regular protocols and an adversary that activates all n nodes in the same round.) The probability that exactly one node broadcasts on f in r is $np_f(1-p_f)^{n-1}$. As in [22], we call this the *success probability* and say that it is a *good* probability only if it is at least $1/\log^2 N$. We now introduce the following claim, which can be found in [22]:

CLAIM 3 ([22]). *Let $x = \lceil 4 \log\log N \rceil$, $m_i = \lfloor x/2 \rfloor + (i-1)x$ for $i = 1, 2, ..., \lfloor \lg N/x \rfloor - 1$. There exist no probability p_f such that both $2^{m_i} p_f (1-p_f)^{2^{m_i}-1}$ and $2^{m_j} p_f (1-p_f)^{2^{m_j}-1}$ are good for $i \neq j$.*

Fix some $R = o\left(\frac{\log^2(N)}{(\mathcal{F}-t)\log\log(N)}\right)$. We construct a table with R rows and one column for each i from Claim 3 such that $2^{m_i} \geq n_{min}$. Notice, the total number of columns remains $\Theta(\log N/\log\log N)$.

In this table, rows represent rounds and columns correspond to possible values of n. We set a counter for each cell (x, y) to describe the number of frequencies with good success probabilities for round x with $n = 2^{m_{i(y)}}$, where $i(y)$ is the value of i corresponding to column y. (The success probabilities are calculated by the well-defined broadcast probabilities for the corresponding round. That is, the success probabilities for a given row x are calculated for the case where no node has received a message through round x.)

Fix some column y and let $n = 2^{m_{i(y)}}$. For each round x for this n, the probability that a given non-good frequency succeeds in having a single broadcaster is less than $1/\log^2 N$. It follows from a union bound that all non-good frequencies fail in x with probability at least $(1-(\mathcal{F}-t)/\log^2 N)$. (Notice, we ignore frequencies in the range $[1, \ldots, t]$ as these are always disrupted by our adversary.)

To bound the probability that all *good* frequencies fail in x we apply Lemma 2. Specifically, let $m = n$, s equal the number of good frequencies in x, p_1 to p_s be the sorted broadcast probabilities for these good frequencies, and p_{s+1} be the probability of not broadcasting on any of these s frequencies. Because $n = \Omega(\log^4 N)$, and for each $f \leq s$, $np_f(1-p_f)^{n-1} \geq 1/\log^2 N$, it follows that $p_{s+1} \geq 1/2$ for sufficiently large N, as required by the lemma.[2] It follows that the probability that no good frequency succeeds is at least 2^{-s}. Over all R rounds for this fixed n, the probability that every frequency fails in every round is at least $\left(1-\frac{\mathcal{F}-t}{\log^2 N}\right)^R \left(2^{-S_y}\right)$, where S_y is the sum of the counters in column y. Since the algorithm, by assumption, fails only with probability $1/N$, this probability cannot be greater than $1/N$. To simplify, we first bound $\left(1-\frac{\mathcal{F}-t}{\log^2(N)}\right)^R > \frac{1}{4}$ and conclude that $2^{-S^y} \leq 4/N$. This implies that $S_y = \Omega(\log N)$.

We can apply this same argument to every column. We need, therefore, $\Theta(\log N/\log\log N)$ columns each with counters that sum to $\Omega(\log N)$. However, we know from Claim 3 that the sum of counters in any row is no greater than $(\mathcal{F}-t)$, and we only have $R = o\left(\frac{\log^2(N)}{(\mathcal{F}-t)\log\log(N)}\right)$ rows, so we fall short of this sum: There must exist at least one column with an insufficient counter sum from which we conclude there exists a value of n value for which the failure probability exceeds $1/N$. □

We continue with our second lower bound which makes no assumption of regularity:

THEOREM 4. *Let $\mathcal{P}$ be a protocol that solves the wireless synchronization problem for some $\varepsilon > 0$, $\mathcal{F}$, t, and N, $\mathcal{F} > t$, with probability at least $1-\varepsilon$. There exists an execution of $\mathcal{P}$ in which some node requires $\Omega\left(\frac{\mathcal{F}t}{\mathcal{F}-t} \cdot \log(1/\varepsilon)\right)$ rounds to synchronize.*

PROOF. Consider an execution in which only two nodes u and v participate. Assume that the adversary wakes up u and v at arbitrary different times. The two nodes cannot both have non-$\perp$ outputs before there is a round in which both nodes choose to broadcast or listen on the same undisrupted frequency. We show that the adversary can always disrupt t frequencies such that the probability of choosing the same undisrupted frequency is at most $c \cdot (\mathcal{F}-t)/(\mathcal{F}t)$ in every round for a sufficiently small constant c.

Let $\mathcal{C}_i$ be the event that u and v do not choose the same undisrupted frequency in round i after the second node is awake. Further, let $\mathcal{D}_i$ be the event that u and v do not choose the same undisrupted frequency in any of the first i rounds after the second node is awake, i.e., $\mathcal{D}_i = \bigcap_{j=1}^{i} \mathcal{C}_i$. To simplify our arguments, we also define $\mathcal{D}_0 = \Omega$ to be the event with probability 1. We will show that the adversary can disrupt frequencies such that for a constant $\gamma < 1$ and all $i \geq 1$,

$$\mathbb{P}[\mathcal{C}_i | \mathcal{D}_{i-1}] \;\geq\; P \;:=\; \max\left\{1 - \frac{1}{4t}, 1 - \frac{\mathcal{F}-t}{\mathcal{F}^2}\right\}. \qquad (1)$$

We then have $\mathbb{P}[\mathcal{D}_i] = \mathbb{P}[\mathcal{D}_{i-1} \cap \mathcal{C}_i] = \mathbb{P}[\mathcal{D}_{i-1}] \cdot \mathbb{P}[\mathcal{C}_i | \mathcal{D}_{i-1}]$, and for all $r < \ln(\varepsilon)/\ln(P) = \ln(1/\varepsilon)/\ln(1/P)$, we thus get $\mathbb{P}[\mathcal{D}_r] > \varepsilon$. The lemma then follows because for a constant $c > 0$,

$$\ln\frac{1}{P} < \frac{1}{P} - 1 \leq \min\left\{\frac{1}{4t-1}, \frac{\mathcal{F}-t}{\mathcal{F}^2-(\mathcal{F}-t)}\right\} \leq c \cdot \frac{\mathcal{F}-t}{\mathcal{F} \cdot t}.$$

It remains to prove Inequality (1). Let us consider a particular round i. We define:

$$p_j \;:=\; \mathbb{P}[u \text{ selects frequency } j \text{ in round } i | \mathcal{D}_{i-1}]$$
and
$$q_j \;:=\; \mathbb{P}[v \text{ selects frequency } j \text{ in round } i | \mathcal{D}_{i-1}].$$

Note that p_j and q_j only depend on the protocol u and v use and on the strategy of the adversary in the first $j-1$ rounds. As the adversary knows all this information, it also

[2] n_{min} is an overestimate of a minimum n needed to guarantee a sufficiently large p_{s+1}. We omit the calculations to preserve space.

know p_j and q_j for all frequencies j. Let J^+ be the set of undisrupted frequencies in round i. We have

$$\mathbb{P}[\overline{\mathcal{C}_i}|\mathcal{D}_{i-1}] = 1 - \mathbb{P}[\mathcal{C}_i|\mathcal{D}_{i-1}] \;=\; \sum_{j\in J^+} p_j \cdot q_j. \qquad (2)$$

The adversary can maximize $\mathbb{P}[\mathcal{C}_i|\mathcal{D}_{i-1}]$ by disruption the frequencies with the t largest p_jq_j products. To simplify the analysis (and w.l.o.g.) assume that the frequencies are reordered such that $p_1q_1 \geq p_2q_2 \geq \cdots \geq p_\mathcal{F}q_\mathcal{F}$. We then get $\mathbb{P}[\overline{\mathcal{C}_i}|\mathcal{D}_{i-1}] = \sum_{j=t+1}^{\mathcal{F}} p_jq_j$. We define $x_j := \sqrt{p_jq_j}$. As a consequence of the inequality of arithmetic and geometric means, we have $x_j \leq (p_j+q_j)/2$ and thus $\sum_{j=t+1}^{\mathcal{F}} x_j \leq 1$. To obtain an upper bound on $\mathbb{P}[\overline{\mathcal{C}_i}|\mathcal{D}_{i-1}]$, we thus have to find $x_1 \geq x_2 \geq \cdots \geq x_\mathcal{F}$ such that $\sum_{j=1}^{\mathcal{F}} x_j \leq 1$ and $\sum_{j=t+1}^{\mathcal{F}} x_j^2$ is maximized. Clearly, the x_j's that maximize the sum satisfy $\sum_{j=1}^{\mathcal{F}} x_j = 1$. We claim that the maximum is obtained[3] for some integer $k \leq \mathcal{F}$, $x_j = 1/k$ for $j \leq k$ and $x_j = 0$ for $j > k$. For the sake of contradiction, assume that this is not the case. Let $s > t$ be the maximum frequency for which $x_s > 0$. For a real number $\varepsilon > 0$, we define $x'_j = x_j - \varepsilon/(s-1)$ and $x''_j = x_j + \varepsilon/(s-1)$ for $j < s$, as well as $x'_s = x_s + \varepsilon$ and $x''_s = x_s - \varepsilon$. Note that we can choose $\varepsilon > 0$ such that $x'_s \leq x'_t$ and $x''_s \geq 0$. We have:

$$\sum_{j=t+1}^{\mathcal{F}} x_j'^2 - \sum_{j=1}^{\mathcal{F}} x_j^2 \quad > \quad 2\varepsilon \cdot \left(x_s - \frac{s-1-t}{s-1}\right)$$

and

$$\sum_{j=t+1}^{\mathcal{F}} x_j''^2 - \sum_{j=1}^{\mathcal{F}} x_j^2 \quad > \quad 2\varepsilon \cdot \left(\frac{s-1-t}{s-1} - x_s\right).$$

Hence, either $\sum_{j=t+1}^{\mathcal{F}} x_j'^2 > \sum_{j=t+1}^{\mathcal{F}} x_j^2$ or $\sum_{j=t+1}^{\mathcal{F}} x_j''^2 > \sum_{j=t+1}^{\mathcal{F}} x_j^2$, a contradiction to the assumption that $x_1, \ldots, x_\mathcal{F}$ maximizes $\sum_{j=t+1}^{\mathcal{F}} x_j^2$. We can therefore assume that there is an integer $k \leq \mathcal{F}$ such that $x_j = 1/k$ for $j \leq k$ and $x_j = 0$ for $j > k$. We then have:

$$\mathbb{P}[\overline{\mathcal{C}_i}|\mathcal{D}_{i-1}] \leq \sum_{j=t+1}^{\mathcal{F}} x_j^2 = \frac{k-t}{k^2}$$

which is maximized for $k = \min\{\mathcal{F}, 2t\}$. This proves Inequality (1) and therefore completes the proof. □

Theorems 1 and 4 combine for the following result:

THEOREM 5. *Let $\mathcal{P}$ be a regular protocol that solves the wireless synchronization problem for some $\mathcal{F}$, t, and N, $\mathcal{F} > t$, with probability at least $1 - 1/N$. Then there exists an execution of $\mathcal{P}$ in which some node requires:*

$$\Omega\left(\frac{\log^2 N}{(\mathcal{F}-t)\log\log(N)} + \frac{\mathcal{F}t}{\mathcal{F}-t}\log N\right)$$

rounds to synchronize.

6. THE TRAPDOOR PROTOCOL

In this section, we present *The Trapdoor Protocol*, a randomized leader-based solution to the wireless synchronization problem that solves the problem in time:

$$O\left(\frac{\mathcal{F}}{\mathcal{F}-t}\log^2 n + \frac{\mathcal{F}t}{\mathcal{F}-t}\log n\right)$$

[3]Note that since we maximize a continuous function over a compact domain, the maximum exists.

6.1 Description of the Protocol

Fix $\mathcal{F}' = min\{\mathcal{F}, 2t\}$. When a node is first activated, we say that it is a *contender*. Each contender proceeds through $\lg N$ *epochs*. (For simplicity of notation, assume N is a power of 2.) Each of the first $\lg N - 1$ epochs is of length $\ell_E = \Theta\left(\frac{\mathcal{F}'}{\mathcal{F}'-t}\log N\right)$ rounds. The final epoch is of length $\ell_E^+ = \Theta\left(\frac{(\mathcal{F}')^2}{\mathcal{F}'-t}\log N\right)$ rounds.

At the beginning of round r of epoch e, every contender chooses a frequency f uniformly at random from $[1, \ldots, \mathcal{F}']$. It then broadcasts a "contender" message on frequency f with probability $\frac{2^e}{2N}$ (see Figure 1). The message is labelled with the contender's *timestamp*, a pair (r_a, uid), where r_a is the number of rounds the contender has been active, and uid is a unique identifier.[4] Otherwise, it listens on frequency f. If a contender receives a message from another contender, and the sender has a larger timestamp (by lexicographic order), then the receiver is *knocked out* (i.e., the trapdoor opens beneath its feet). A node that is knocked out continues to listen on a random channel (choosen from $[1, \ldots, \mathcal{F}']$) in every round. If a contender completes all $\lg N$ epochs without being knocked out, then it becomes a leader.

As soon as some contender becomes a leader, it chooses a numbering scheme for the rounds, and begins to output a round number in every round. From that point onwards, in every round, it chooses a channel at random (from $[1, \ldots, \mathcal{F}']$) and sends a message containing the numbering scheme with probability 1/2. Any node that receives a message from a leader immediately abandons the protocol, adopts the specified numbering scheme, and begins to output a round number in every round.

6.2 Analysis

In this section, we analyze the Trapdoor Protocol. Throughout, when we say that a claim holds "with high probability," we mean with probability $1 - 1/N$. (It is easy to generalize the protocol such that the probability of error is polynomially small in N.) We begin, however, with some notation. We assume a global round counter (unknown to the individual nodes). We denote by p_u^r the probability that node u broadcasts in round r. For each round r, we define the broadcast weight $W(r) = \sum_u p_u^r$. We also note the following two probability facts, which follow from the standard approximation that $(1-x)^x \approx 1/e$, and by rewriting $(1-p)$ as $((1-p)^p)^{1/p}$.

FACT 6.

- *For any $p > 0$: $(1-p) < e^{-p}$.*
- *For any $p \leq 1/2$: $(1-p) \geq (1/4)^p$.*

We first bound the probability that a contender is knocked out in a given round r. The probability that a contender is knocked out depends on the broadcast weight of the contenders that have a larger timestamp. We thus define $S(r, u)$ as follows:

[4]A unique identifier can be generated at random when a node is first activated by choosing an integer uniformly at random from a range $[1, \ldots, cN^2]$, where the constant c is chosen to be sufficiently large, as a function of the desired error probability.

Epoch #	1	2	...	$N-1$	$\lg N$
Length	$\Theta\left(\frac{\mathcal{F}'}{\mathcal{F}'-t}\log N\right)$	$\Theta\left(\frac{\mathcal{F}'}{\mathcal{F}'-t}\log N\right)$	...	$\Theta\left(\frac{\mathcal{F}'}{\mathcal{F}'-t}\log N\right)$	$\Theta\left(\frac{(\mathcal{F}')^2}{\mathcal{F}'-t}\log N\right)$
Prob.	$1/N$	$2/N$	...	$1/4$	$1/2$

Figure 1: Epoch lengths and contender broadcast probabilities for the Trapdoor Protocol.

DEFINITION 7. *For round r and node u: We define $S(r)$ to be the set of nodes that are active (i.e., contenders or leaders) in round r. We define $S(r,u) \subseteq S(r)$ to be the set of active nodes that are leaders or have a timestamp larger than the timestamp of u in r. We define $W(r,u)$ to be the broadcast weight of the nodes in $S(r,u)$.*

The first lemma shows that if for some round r, $W(r) = \Theta(\mathcal{F}')$, and for some node u, a constant fraction of this weight is in $W(r,u)$, then u is knocked out with a constant probability. This captures the intuition that when $W(r) = \Theta(\mathcal{F}')$, a constant amount of probability mass is expected on each frequency.

LEMMA 8. *Fix a round r and a node u that is a contender at the beginning of round r. Assume that $c\mathcal{F}' \leq W(r) \leq 3c\mathcal{F}'$ and $W(u,r) \geq W(r)/c'$ for some positive constants $c, c' \geq 1$. Then node u is knocked out in round r with probability at least:*

$$\frac{\mathcal{F}'-t}{2\mathcal{F}'}\left(\frac{c}{c'}\right)\left(\frac{1}{4}\right)^{3c}.$$

PROOF. Fix f to be the frequency chosen by u in round r. We first bound the probability that u listens on channel f, and f is not disrupted by the adversary, to be at least $(\mathcal{F}'-t)/2\mathcal{F}'$ (since $(1-p_u^r) \geq 1/2$).

We next bound the probability that exactly one node from $S(r,u)$ broadcasts, and no other node broadcasts on f, as at least:

$$\sum_{v \in S(r,u)}\left(\frac{p_v^r}{\mathcal{F}'}\prod_{w \neq u,v}\left(1-\frac{p_w^r}{\mathcal{F}'}\right)\right).$$

Noting that $c\mathcal{F}'/c' \leq W(r,u)$, and $W(r) \leq 3c\mathcal{F}'$, and applying Fact 6, this is at least $\left(\frac{c}{c'}\right)\left(\frac{1}{4}\right)^{3c}$, yielding the desired bound. □

We apply Lemma 8 to derive a bound on the probability that the total weight gets *too* high. The key idea is that once the broadcast weight reaches $\Theta(\mathcal{F}')$, the probability of being knocked out becomes sufficiently high to bring the broadcast sum back down. In this sense, the probability mass in our system behaves like a self-regulating feedback circuit: when it grows too large, it reduces itself.

LEMMA 9. *Let R be a round in which there is at most one leader at the start of R. For all rounds $r \leq R$: $W(r) < 6\mathcal{F}'$, with high probability.*

PROOF. Assume for contradiction that the weight exceeds $6\mathcal{F}'$ by R. Let r be the largest round less than R such that $W(r) \leq 2\mathcal{F}'$ and $W(r+1) > 2\mathcal{F}'$. Notice that between rounds r and $r+\ell_E$, the broadcast weight of activated nodes can at most double. Moreover, at most N new nodes can be activated, each with weight $1/N$. Thus $W(r+\ell_E) \leq 2W(r)+1 < 6\mathcal{F}'$. It follows that $R \geq r+\ell_E$.

We divide the rounds from $r+1$ to $r+\ell_E$ into a constant number of equal-sized *intervals*, and argue that in each such interval, the total weight is decreased by a constant fraction. For appropriate choice of constants, this implies that $W(r+\ell_E) \leq 2\mathcal{F}'$, which contradicts our choice of r.

Fix some round r' that is the first round in one of the intervals. Choose non-leader node u such that $W(r',u) \geq 2W(r')/3$ (the existence of such a non-leader u relies on our restriction on the number of leaders). For each round r'' during the interval, there are two possibilities: (1) $W(r'',u) \geq W(r')/3$: in this case, we can apply Lemma 8 with c and the appropriate value of c', and conclude that u is knocked out with probability $\Theta(\frac{\mathcal{F}'}{\mathcal{F}'-t})$ in round r'' (notice, because the increase in weight is bounded in this interval, we can fix a value of c' that works throughout the interval); (2) $W(r'',u) < W(r')/3$: in this case, we can conclude that at least $1/3$ of the broadcast weight $W(r')$ has been knocked out between the rounds r' and r''. If case (2) occurs even once, then the weight has been reduced by a constant fraction in this interval, as desired. Otherwise, case (1) occurs in each of the $\Theta(\frac{\mathcal{F}'}{\mathcal{F}'-t}\log N)$ rounds of the interval, and node u is therefore knocked out with high probability. Taking a union bound over all non-leader nodes u where $W(r',u) \geq 2W(r')/3$, we conclude that with high probability enough of these low timestamp nodes are knocked out to also reduce the weight by a constant fraction. Either way we have reduced the weight at the start of the interval by at least $1/3$ by the end. Over a constant number of such intervals, the weight decreases below $2\mathcal{F}'$, contradicting our choice of r. □

We conclude with the main theorem statement. The key argument concerns agreement, which shows that there is at most one leader, with high probability. It relies on Lemma 9 to show that the total broadcast weight in the system remains sufficiently low, and hence any second (potential) leader will be knocked out as it advances through its final epoch.

THEOREM 10. *The Trapdoor Protocol solves the wireless synchronization problem in time:*

$$O\left(\frac{\mathcal{F}}{\mathcal{F}-t}\log^2 N + \frac{\mathcal{F}t}{\mathcal{F}-t}\log N\right).$$

PROOF. It is easy to see that Properties 1–3 of the wireless synchronization problem follow directly from the definition of the protocol. To establish Property 4, agreement, we show that at most one contender becomes leader. Let u be the process with the largest timestamp, i.e., u is the first process to be activated (with ties broken by identifier). As no contender can knock out u, it is clear that u becomes leader. Assume for the sake of contradiction that at least

one other node also becomes leader. Let v be the first to become leader from among these other nodes. Consider a round r in node v's final epoch. By Lemma 9, we know that $W(r) \leq 6\mathcal{F}'$, with high probability. We also know that u sends a message to receiving v on a non-disrupted channel with probability at least $(\mathcal{F}'-t)/(4(\mathcal{F}')^2)$, and that no other contender broadcasts on the same channel with probability at least $\prod_{w \neq u,v}(1-p_w^r)$. This latter expression is at least $(1/4)^6 = O(1)$. Hence with probability $\Omega((\mathcal{F}'-t)/(\mathcal{F}')^2)$, node u knocks out node v. Note, the final epoch length is $\Theta(\frac{(\mathcal{F}')^2}{(\mathcal{F}'-t)} \lg N)$, so over the full final epoch, we conclude that node v is knocked out with high probability. As this holds for all v (by a union bound), we conclude that u is the only leader, with high probability. Property 5 follows from the fact that for every process $w \neq u$, once u becomes leader, w receives a message from u with probability $\Omega((\mathcal{F}'-t)/(\mathcal{F}')^2)$.

Finally, we convert from $\mathcal{F}'$ to $\mathcal{F}$ in the final running time by noting that $\frac{\mathcal{F}'}{\mathcal{F}'-t} = \Theta(\frac{\mathcal{F}}{\mathcal{F}-t})$ and $\frac{(\mathcal{F}')^2}{\mathcal{F}'-t} = \Theta(\frac{\mathcal{F}t}{\mathcal{F}-t})$. □

7. THE GOOD SAMARITAN PROTOCOL

The Trapdoor Protocol tolerates up to t frequencies disrupted per round. For practical networks, however, there are often significantly lower levels of interference. In this section, we present an optimistic, *adaptive* protocol that can terminate faster in executions with low interference. *The Good Samaritan Protocol* guarantees that when $n \geq 2$ nodes are activated at the same time, and when no more than $t' < t$ frequencies are disrupted per round, the protocol terminates within $O(t' \log^3 N)$ rounds—a significant improvement over the Trapdoor Protocol when t' is much smaller than t. For general executions, it guarantees termination within $O(\mathcal{F} \log^3 N)$ rounds, only a factor of $\log N$ slower than the Trapdoor Protocol.

Throughout this section, we assume that $t \leq \mathcal{F}/2$ (the protocol can be modified to work for any constant fraction of $\mathcal{F}$), and we model the adversary as *oblivious*, meaning that it can be described as a fixed sequence of probability distributions over sets of frequencies to disrupt. This constraint simplifies the analysis, but because our nodes always select frequencies randomly and independently, it does not prohibitively weaken the adversary. To simplify notation, we assume that t' is a power of 2.

7.1 Description of the Protocol

As in the Trapdoor Protocol, every node begins as a *contender*, and each contender tries to become a *leader*. Once a leader, it dictates the round synchronization. Unlike the Trapdoor Protocol, a contender is not knocked out when it receives a message from another contender; instead, it is downgraded and becomes a *good samaritan*; a good samaritan attempts to help the contenders to become leaders. If a samaritan receives a message from another samaritan, it is *knocked out* and becomes passive.

Notice that while a node is participating in the Good Samaritan Protocol, it ignores timestamps. That is, when a contender receives a message from another contender, it is downgraded to a good samaritan *even if it has a larger timestamp* than the sneder. Similarly, a samaritan is knocked out whenever it receives a message from another samaritan, regardless of its timestamp.

Basic structure.

Each node proceeds through $\lg F$ super-epochs, before falling back to the Trapdoor Protocol, modified as described below. If all n nodes start at the same time, and if the adversary blocks only $t' < t$ frequencies, then all nodes complete the synchronization protocol by the end of super-epoch $\log 2t'$. (See Figure 2 for an overview of the round structure.)

Each super-epoch consists of $\lg N + 2$ epochs. In epoch $e \leq \lg N$, we define probability $p_e = 2^e/2N$; for the final two epochs, define $p_e = 1/2$. This probability reflects the probability that a node broadcasts. Each epoch consists of $s(k) = \Theta(2^k \log^3 N)$ rounds. In super-epoch k, during each epoch $e \leq \lg N$, a contender or a samaritan behaves as follows. In each round, with probability $1/2$ it chooses a frequency at random from the range $[1 \ldots 2^k]$; and with probability $1/2$ it chooses a frequency at random from the range $[1 \ldots \mathcal{F}]$. With probability p_e, it decides to broadcast; with probability $(1-p_e)$ it decides to listen.

Contenders and samaritans behave differently in the last two epochs of each super-epoch. In each round, with probability $1/2$, they choose a frequency at random from the range $[1 \ldots 2^k]$ and broadcast with probability p_e as before. With probability $1/2$, however, the round is designated as *special*: in this case, a contender chooses some $d \in [1 \ldots \mathcal{F}]$ uniformly at random, and selects a frequency at random from the range $[1 \ldots 2^d]$. It then broadcasts with probability $1/2$, and listens with probability $1/2$.

Becoming the leader.

The goal of the contender is to successfully send a sufficient number of messages during epoch $\lg N + 1$ of a super-epoch; if this occurs, then it can go on to become leader. The only way it can discover whether its messages were successfully *sent*, however, is by learning from a good samaritan that its messages were successfully *received*. When a good samaritan receives a message from a contender u in some round r, where: (a) round r is part of epoch $\lg N + 1$; (b) round r is not designated as special by either the contender or the samaritan, and (c) both the contender and the samaritan were awakened in the same round, then the samaritan records the fact that round r was successful for u.

Let $s(k)$ be the length of an epoch in super-epoch k. If a contender in super-epoch k learns from a good samaritan that it was successful in at least $s(k)/2^{k+6}$ rounds of epoch $lgN + 1$, then it becomes a leader. We say that an epoch in which a contender sends a sufficient number of successful messages is a *critical* epoch.

Afterward.

Once a contender becomes a leader, it continues in every round to first choose an integer $d \in [1 \ldots \mathcal{F}]$ at random, and then to choose a frequency at random in the range $[1 \ldots 2^d]$, and then to broadcast its preferred round numbering with probability $1/2$. Any node that receives a message from a leader immediately abandons its protocol, adopts the specified round numbering, and ceases to contend.

If an unsynchronized node exits the last super-epoch without becoming a leader, then it continues to execute the modified Trapdoor protocol. Specifically, in every round, it flips a coin with probability $1/2$, and either executes a round of the Trapdoor protocol, or a special round of the Good Samaritan Protocol:

Super Epoch #	$k = 1$ to $\log F$					
Epoch #	1	2	...	$\lg N$	$\lg N + 1$	$\lg N + 2$
Length	$\Theta\left(2^k \log^3 N\right)$	$\Theta\left(2^k \log^3 N\right)$	...	$\Theta\left(2^k \log^3 N\right)$	$\Theta\left(2^k \log^3 N\right)$	$\Theta\left(2^k \log^3 N\right)$
Probability	$1/N$	$2/N$	...	$1/2$	$1/2$	$1/2$
Probability of Choosing Frequency f	$\begin{cases} \mathbb{P}[f] = 1/2^{k+1} + 1/2\mathcal{F} & : f \le 2^k \\ \mathbb{P}[f] = 1/2\mathcal{F} & : f > 2^k \end{cases}$				$\mathbb{P}[f] = \frac{2^{\lfloor \lg (F/f) \rfloor + 1} - 1}{2\mathcal{F} \lg \mathcal{F}} + 1/2^{k+1}$	

Figure 2: Epoch structure, broadcast prob., and frequency distributions for the Good Samaritan Protocol.

- With probability 1/2, it executes the Trapdoor protocol, where each epoch is of length $\Theta(\mathcal{F} \log^3 n)$, i.e., at least four times as long as the longest epoch of the Good Samaritan Protocol. (When executing the Trapdoor protocol, timestamps are again used to determine when a node is knocked out.)

- With probability 1/2, it executes a special round of the Good Samaritan protocol: it chooses an integer $d \in [1 \ldots \log \mathcal{F}]$ at random, and then chooses a frequency at random in the range $[1 \ldots 2^d]$, broadcasting a message with probability 1/2.

Any contender that has not yet begun the modified Trapdoor protocol that receives a message is downgraded. We refer to the $\lg \mathcal{F}$ super-epochs as the *optimistic* portion of the Good Samaritan protocol, and the modified Trapdoor Protocol as the *fallback* portion.

7.2 Analysis

We now provide a brief outline of the analysis for the Good Samaritan Protocol, focusing on where it differs from the Trapdoor Protocol. We begin by showing that at most one leader is elected during the optimistic portion of the Good Samaritan Protocol.

Assume for the sake of contradiction that two nodes u and v both become leaders during the optimistic portion of the Good Samaritan Protocol, and the u begins its critical epoch no later than v. Let w be the good samaritan that assists w in becoming leader. Let k be the super-epoch in which v becomes leader.

We treat the random choices made by all other nodes, as well as the choices of the adversary, as fixed and independent of the choices made by u, v, and w. For each such set of random/adversarial choices, we bound the probability that u and v both become leaders. We observe that in each round during v's critical epoch, the adversary disrupts some frequencies, and the other nodes may broadcast on some frequencies. Let $p^a(r)$ be the fraction of frequencies that are neither disrupted by the adversary in round r, nor broadcast on by any of the other nodes. (Note that $p^a(r)$ is a random variable that depends on the random choices of the other nodes.) We then calculate that the probability of node v successfully sending a message to the samaritan w in round r during the critical epoch is $\le \frac{p^a(r)}{2^{k+3}}$. If w receives a sufficient number of message from v, then there must be a sufficient number of rounds in which the frequencies are not too disrupted:

LEMMA 11. *If v becomes leader, then there are at least $s(k)/32$ rounds in the critical epoch for v such that $p^a(r) \ge 1/32$, with high probability.*

(Otherwise, node v succeeds, in expectation, in sending at most $s(k)/2^{k+7}$ messages to w, which is insufficient.) From this we can obtain a contradiction, by arguing that, with high probability, v must receive a message from u during one of those $s(k)/32$ "good" rounds. (In expectation, u delivers $s(k)/2^{k+14} \log N$ messages to v, and $s(k) \ge \Omega(2^k \log^2 N)$.)

LEMMA 12. *At most one node becomes leader during the optimistic portion of the Good Samaritan protocol, with high probability.*

The next step is to show that the modified Trapdoor Protocol continues to work. As in the previous analysis in Lemma 8, we can show that if the broadcast weight gets too high, then the probability of getting knocked out is high, which leads again to a bound on the total broadcast weight. Some additional care is needed, as now we must bound separately the three different weights: W_1, the weight of Good Samaritan contenders, W_2, the weight of samaritans, and W_3, the weight of contenders in the modified Trapdoor Protocol:

LEMMA 13. *Let R be a round with at most one leader, and c be some positive constant $c > 3/2$. For all rounds $r \le R$, with high probability: (a) $W_1(r) < 3c\mathcal{F}$; (b) $W_1(r) + W_2(r) < 9c\mathcal{F}$; (c) $W_3(r) \le 3c\mathcal{F}$.*

As in the original analysis, we can then show that the first contender to become leader successfully knocks out any later contenders. Some care is needed, as the smallest of the channels may be too congested. Instead, we rely only on channels in the range $[\mathcal{F}/4, \ldots, \mathcal{F}]$ to show that contenders are knocked out.

LEMMA 14. *The modified Trapdoor protocol chooses only one leader, with high probability.*

Finally, we examine the interactions between the optimistic portion of the protocol and the modified Trapdoor Protocol. Assume some node v becomes leader during the optimistic portion, and some other node u becomes leader during the fallback protocol. There are two cases to consider.

Assume that node u began its final epoch first. We conclude that during every round of v's critical epoch, u is contending or acting as leader. It is easy to see, by the same argument as in Lemma 12, that node u knocks out v. (Here we take advantage of the fact that with probability 1/2, u

performs a special round of the Good Samaritan protocol, and hence with probability $1/\lg F$ chooses the same range of channels being used by v. This also relies on the fact that there are at least $s(k)/32$ "good" rounds during v's critical epoch.)

By contrast, consider the case where v begins its critical epoch no later than u begins its final epoch. In this case, since u's final epoch is at least four times longer than v's, we conclude that after v becomes leader, it has sufficiently many rounds to knock out u, preventing u from becoming the leader. (Here we also take advantage of the fact that the broadcasts weights are bounded, hence not causing too much congestion.)

Putting together Lemmas 11, 12, 13, and 14, we conclude:

THEOREM 15. *The Good Samaritan protocol chooses at most one leader, with high probability.*

It remains only to show that when all the nodes start at the same time, and when only t' channels are disrupted in each round, then a leader is elected. This proceeds much as before, first bounding the total broadcast weight, this time focusing on the first $2t'$ channels:

LEMMA 16. *Let R be a round in the first $\lg N$ epochs with at most one leader, and c be some positive constant $c > 3/2$. For all rounds $r \leq R$, with high probability: (a) $W_1(r) < 6ct'$; (b) $W_1(r) + W_2(r) < 18ct'$.*

We then argue that by the end of epoch $\log N$ there is exactly one contender and one good samaritan remaining. This follows from the same argument as was used in Theorem 10 to show that there is exactly one leader remaining in the Trapdoor Protocol, except here focusing only on the first $2t'$ channels:

LEMMA 17. *By the end of epoch $\lg N$, there is one contender and one samaritan, with high probability.*

We now observe that the good samaritan receives sufficiently many messages from the contender in epoch $\lg N + 1$, and that the contender receives enough messages from the good samaritan in epoch $\log N + 1$. This follows from a simple probability calculation, as there are no longer any other contenders/samaritans causing unexpected disruptions.

Finally, we conclude that once a leader is chosen, every node soon receives a synchronizing message from it, and we conclude with the main result:

THEOREM 18. *The Good Samaritan protocol solves the wireless synchronization problem with an oblivious adversary, and has the following properties:*

- *In every execution, the synchronization problem is solved within $O(\mathcal{F} \log^3 N)$ rounds.*
- *If all $n \leq N$ nodes awake in the same round, and if $n \geq 2$, and if the adversary disrupts at most $t' \leq t$ frequencies per round, then every node completes the synchronization within $O(t' \log^3 N)$ rounds.*

8. CONCLUDING REMARKS

In this paper, we introduce the *wireless synchronization problem* for disrupted single-hop radio networks. We present lower bounds on the efficiency of synchronization, and describe two new synchronization protocols. Of note, the first protocol nearly matches the lower bound, and the second has an optimistic mode in which it decides rapidly in executions that have little interference and synchronized activations.

Unsynchronized rounds.

Throughout this paper, we assumed that nodes agree in advanced on synchronized round boundaries. In general, however, slotted communication models can be transformed into non-slotted models, with a constant multiplicative cost; c.f., [1]. We believe that similar techniques can be applied to modify our protocols to work in a setting without synchronized round boundaries. We leave the consideration of the details of this transformation as important future work.

Broader implications.

The techniques developed in this paper can be used to solve a variety of problems. A common round view allows protocols that require synchronous activation to be used in our more realistic ad hoc setting. Notice, our protocols elect a unique leader as a sub-problem, and a leader combined with a common round view simplifies consensus, maintaining replicated state, and the collection and distribution of messages, among other useful problems.

Fault-tolerance.

A natural question is whether wireless synchronization can be achieved in the presence of nodes joining, leaving, crashing, and/or restarting. We can easily modify the Trapdoor Protocol to tolerate crash failures: whenever a node does not receive a message from the leader for sufficiently long (e.g., $\Omega(\mathcal{F}^2/(\mathcal{F}-t)\log N)$ rounds), it restarts. Moreover, each node delays outputting a round number until it has received sufficiently many messages from the leader (thus ensuring that every node has received a message from the leader). Assuming the adversary is oblivious (and designates failures prior to the random choices made during the execution), then a correct leader is chosen and the nodes synchronize in the same asymptotic time. On the other hand, it remains a challenging open question to perform synchronization in the presence of Byzantine failures or arbitrary state corruptions.

Other open questions.

There remain several interesting related questions to consider. First, we conjecture that the Trapdoor Protocol is optimal. This motivates work on improving our lower bound; the $\log \log N$ factor can likely be eliminated using the linear programming techniques of [17], but closing the remaining gap presents a challenge. In particular, the $(\mathcal{F} - t)$ terms is an artifact of focusing on the necessary, but not sufficient, sub-problem of achieving a single undisrupted broadcast. It also remains open whether better *optimistic* protocols can be developed, such as protocols that do not require all the nodes to begin in the same round. It would also be interesting to explore whether synchronization can be achieved deterministically (perhaps using *multi-selectors* [19]), or how our results can be adapted to multiple hops.

Beyond synchronization itself, there remain a diversity of open questions concerning the disrupted radio channel model—what other problems can we solve and what limitations are inherent? Even some basic questions—e.g., Can agreement be reached? How fast? Under what conditions—are not yet answered.

9. REFERENCES

[1] N. Abramson. The aloha system - another approach for computer communications. *Proceedings of the Fall Joint Computer Conference*, 37:281–285, 1970.

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50(13):2127–2159, 2006.

[3] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. On the complexity of radio communication. In *Proceedings of the Symposium on Theory of Computing*, 1989.

[4] R. Bar-Yehuda, O. Goldreich, and A. Itai. Efficient emulation of single-hop radio network with collision detection on multi-hop radio network with no collision detection. *Distributed Computing*, 5:67–71, 1991.

[5] 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.

[6] A. D. Bonis, L. Gasieniec, and U. Vaccaro. Optimal two-stage algorithms for group testing problems. *SIAM Journal on Computing*, 34(5):1253–1270, 2005.

[7] I. Chlamtac and S. Kutten. On broadcasting in radio networks - problem analysis and protocol design. *IEEE Transactions on Communications*, 33(12):1240–1246, 1985.

[8] B. Chlebus and D. Kowalski. A better wake-up in radio networks. In *Proceedings of the International Symposium on Principles of Distributed Computing*, 2004.

[9] B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in unknown radio networks. In *Proceedings of the Symposium on Discrete Algorithms*, 2000.

[10] B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in ad hoc radio networks. *Distributed Computing*, 15(1):27–38, 2002.

[11] M. Chrobak, L. Gasieniec, and D. Kowalski. The wake-up problem in multi-hop radio networks. In *Proceedings of the Symposium on Discrete Algorithms*, 2004.

[12] M. Chrobak, L. Gasieniec, and W. Rytter. broadcasting and gossiping in radio networks. *Journal of Algorithms*, 43:177–189, 2002.

[13] 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.

[14] A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *Proceedings of the Symposium on the Foundations of Computer Science*, 2003.

[15] 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 International Symposium on Distributed Computing*, 2007.

[16] S. Dolev, S. Gilbert, R. Guerraoui, and C. Newport. Secure communication over radio channels. In *Proceedings of the International Symposium on Principles of Distributed Computing*, 2008.

[17] M. Farach-Colton, R. J. Fernandes, and M. A. Mosteiro. Lower bounds for clear transmissions in radio networks. In *Proceedings of the Latin American Symposium on Theoretical Informatics*, 2006.

[18] L. Gasieniec, A. Pelc, and D. Peleg. The wakeup problem in synchronous broadcast systems. *SIAM Journal of Discrete Mathematics*, 14(2):207–222, 2001.

[19] S. Gilbert, R. Guerraoui, D. Kowalski, and C. Newport. Interference-resilient information exchange. In *Proceedings of the IEEE Conference on Computer Communications*, 2009.

[20] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and mitigating the impact of rf interference on 802.11 networks. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2007.

[21] P. Indyk. Explicit constructions of selectors and related combinatorial structures, with applications. In *Proceedings of the Symposium on Discrete Algorithms*, 2002.

[22] T. Jurdzinski and G. Stachowiak. Probabilistic algorithms for the wakeup problem in single-hop radio networks. In *Proceedings of the International Symposium on Algorithms and Computation*, 2002.

[23] J. Komlos and A. Greenberg. An asymptotically fast non-adaptive algorithm for conflict resolution in multiple access channels. *IEEE Transactions on Information Theory*, March 1985.

[24] D. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. In *Proceedings of the International Symposium on Principles of Distributed Computing*, 2003.

[25] D. Kowalski and A. Pelc. Time of radio broadcasting: Adaptiveness vs. obliviousness and randomization vs. determinism. In *Proceedings of the Colloquium on Structural Information and Communication Complexity*, 2003.

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

[27] D. R. Kowalski and A. Pelc. Deterministic broadcasting time in radio networks of unknown topology. In *Proceedings of the Symposium on the Foundations of Computer Science*, 2002.

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

[29] D. Meier, Y. Pignolet, S. Schmid, and R. Wattenhofer. Speed dating despite jammers. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems*, 2009.

[30] M. Strasser, S. Capkun, C. Popper, and M. Cagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *Proceedings of the Symposium on Security and Privacy*, 2008.

[31] M. Strasser, C. Popper and S. Capkun. Efficient uncoordinated FHSS anti-jamming communication. In *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing*, 2009.