Fault-Tolerant Leader Election and Collective Coin-Flipping in the Full Information Model

Yevgeniy Dodis

February 28, 2006

Let me outline the organization of this report. In Section 1 I will define the full-information model of computation. I will then define in Section 2 the two main problems studied in this model: collective coin-flipping and leader election. In Section 3 I will describe the connections between these problems and give some basic results. Traditionally, coin-flipping protocols are divided into two classes: one-round vs. general. One-round protocols are related to the notion of influences of variables on boolean functions. We discuss this and the known lower and upper bounds for one-round coin-flipping in Section 4. General coin-flipping protocols have been traditionally solved via leader election protocols. Thus, we move directly to leader election protocols in Section 5. This will complete the overview of main known results for collective coin-flipping and leader election and their relation to each other. Section 6 summarizes these results once again, presents main open questions and talks about related problems and future directions. The next three sections talk in a bit more detail about the three assigned papers. Finally, in Section 10 I will talk about the main open problem that I considered: collective coin-flipping against *adaptive* adversaries.

1 Full-Information Model

The *full-information model* (or *perfect* information model) was introduced by Ben-Or and Linial [BL90]. In this model n *computationally unbounded* players (also called parties or processors) are trying to perform some task by means of a *single broadcast channel*. As usual, we assume that some subset of the parties can be *faulty* or malicious, and we would like to "protect" the honest parties as much as possible. Taking the worst case scenario, we assume that all the faulty parties are coordinated by a central *adversary* $\mathcal{A}$, who can corrupt up to b out of n players. Most of the papers assume and *crucially use* the fact that the adversary $\mathcal{A}$ is *static*, i.e. it decides on which parties to corrupt *before the protocol starts*. We will talk about more general *adaptive adversaries* in Section 10 and assume static adversary for now. The computation proceeds in rounds, in which each processor broadcasts a message to the other processors. The crucial complication is that the network is assumed to be *asynchronous within a round* and is synchronized only in between the rounds. For example, players cannot flip a coin by broadcasting a random bit and taking their exclusive OR: the last player to talk can completely control the output. Again taking the worst case scenario, we assume that in each round first $\mathcal{A}$ receives all the messages broadcast by the honest players, and only then decides which messages to send on behalf of the bad players. Finally, we assume that $\mathcal{A}$ never violates the protocol in the manner that can be detected (for example, if a faulty processor has to send a random bit, he does so; however, the bit need no be random).

We note that since the parties are computationally unbounded, cryptographic techniques are of no use. Together with the lack of private channels, this also means that classical multi-party computation techniques (like secret sharing) cannot be used as well. In fact, since all the parties always have the same information during the computation (aside from their possible inputs and random tapes), no reasonable notion of privacy makes sense in this model. The only thing we can try to protect against, is for the faulty parties to bias the output of the computation to some outcome "they desire". In other words, the system should be *resilient* against faulty coalitions trying to "affect the outcome". As we will see, there are several formalizations of

this notion of resilience, depending on the task at hand.

Before describing particular problems we will study, we mention the following very general result about the full-information model, proven by Goldreich, Goldwasser and Linial [GGL98]. The result right away illustrates some of the limitations of the full-information model.

Theorem 1 ([GGL98]) *Let Π be an arbitrary protocol in the full-information model, where players output elements from some finite domain D. For $v \in D$, let p_Π^v be the probability that the outcome is v if all n players are non-faulty. Then for any $v \in D$ and any $b \in [n]$ there exists a set B_v of faulty players of cardinality b that can force the outcome of Π to v with probability at least $(p_\Pi^v)^{1-b/n}$.*

2 Definitions and Main Known Results

We consider two most natural problems in the full-information model: *collective coin-flipping* and *leader election*. Briefly, collective coin-flipping is a problem of collectively generating a random bit, such that bad players cannot bias this bit too much. And leader election is a problem of collectively choosing a single representative or a *leader* among n players, in such a way that the probability of choosing a faulty player as a leader is not too large. We now define these problems more precisely (in the definitions below the number of players is always denoted by n and is typically omitted from the notation).

Definition 1 *For a coin-flipping protocol Π and a subset of players B,*

- $p_\Pi^1(B)$, probability of forcing 1, *denotes be maximum over possible strategies of players in B of the probability that the output of Π is 1.*
- $a_\Pi^1(B)$, probability of not avoiding 1, *denotes be minimum over possible strategies of players in B of the probability that the output of Π is 1.*
- $p_\Pi^1 = a_\Pi^1 = p_\Pi^1(\emptyset) = a_\Pi^1(\emptyset)$, natural probability of 1, *probability of 1 with no faulty players.*
- $p_\Pi^0(B) = 1 - a_\Pi^1(B)$, $a_\Pi^0(B) = 1 - p_\Pi^1(B)$, $p_\Pi^0 = a_\Pi^0 = 1 - p_\Pi^1 = 1 - a_\Pi^1$ *have similar meaning.*
- $p_\Pi(B) = \max(p_\Pi^0(B), p_\Pi^1(B))$, probability of forcing some output.
- $a_\Pi(B) = \min(a_\Pi^0(B), a_\Pi^1(B)) = 1 - p_\Pi(B)$, probability of not avoiding any output.
- Π *is called* (B, ε)-resilient, *if irrespective of the strategy of players in B, we have:*

$$\varepsilon \le \Pr(coin = 1) \le 1 - \varepsilon$$

  Equivalently, $a_\Pi(B) \ge \varepsilon$ (or $p_\Pi(B) \le 1 - \varepsilon$).
- $a_\Pi(b)$ *denotes the minimum of $a_\Pi(B)$ over all sets B of size b.*

Definition 2 *For a leader election protocol Π and a subset of players B,*

- $e_\Pi(B)$ *denotes the minimum over possible strategies of players in B of the probability of choosing a non-faulty leader, i.e. a leader* not *belonging to B.*
- Π *is called* (B, ε)-resilient, *if irrespective of the strategy of players in B, we have:*

$$\Pr(leader \notin B) \ge \varepsilon$$

  Equivalently, $e_\Pi(B) \ge \varepsilon$.
- $e_\Pi(b)$ *denotes the minimum of $e_\Pi(B)$ over all sets B of size b.*

The quantities $a_\Pi(\cdot)$ and $e_\Pi(\cdot)$ are sometimes called the *resilience (or success) probability* of the corresponding coin-flipping or leader election protocol.

Definition 3 *For a coin-flipping or a leader election protocol* Π*,*

- *If* Π *is* (B, ε)*-resilient for every coalition* B *of at most* b *players, we say that* Π *is* (b, ε)-resilient*. (i.e.,* $a_\Pi(b) \geq \varepsilon$ *for coin-flipping, and* $e_\Pi(b) \geq \varepsilon$ *for leader election).*
- *When the number of players* n *is a parameter of the protocol,* b *and* ε *are the functions of* n*. If* Π *is* $(b(n), \varepsilon(n))$*-resilient and there exists a constant* $\varepsilon_0 > 0$ independent of n *such that* $\varepsilon(n) \geq \varepsilon_0$*, we say that* Π tolerates $b(n)$ *faulty players, or is* $b(n)$-resilient *(i.e., we omit* $\varepsilon(n)$ *from the notation). The largest such* $b(n)$ *is called the* resilience threshold *of* Π*.*

Thus, resilience for a coin-flipping protocol means that every outcome happens with some positive probability, i.e. faulty players cannot force any particular outcome of the coin. Note, however, that we allow them to almost completely bias the coin, as long as this bias is fixed and independent of the number of players. Resilience for a leader election protocol means that with some positive (again, maybe small) probability the elected leader is not faulty.

Finally, we quantify the resilience probability (or the "probability of success") for the best possible protocol.

Definition 4 *Let*

- $a(b(n))$ *be the maximum* $\varepsilon(n)$ *such that there exists* $(b(n), \varepsilon(n))$*-resilient coin-flipping protocol.*
- $e(b(n))$ *be the maximum* $\varepsilon(n)$ *such that there exists* $(b(n), \varepsilon(n))$*-resilient leader election protocol.*

Let us give a couple of quick examples. For coin-flipping, a trivial protocol is a "parity" protocol. Each player announces a bit, and the coin is their exclusive OR. As we saw, this protocol is not even 1-resilient; a single faulty player can control the outcome. A somewhat better one-round protocol is the "majority" protocol, where the resulting coin is the majority of the bits transmitted by the players. We know that any coalition of $O(\sqrt{n})$ players with high probability does not affect the majority, i.e. the majority is already determined when other players choose their bits at random. Thus, majority is a $\sqrt{n}$-resilient one-round coin-flipping protocol. As we will see, we can do much better, even in a single round.

For leader election, we give a toy example (dating to [BL90]) of a "perfect" leader election protocol tolerating (for $n \geq 3$) any single faulty player. That is, for any choice of the faulty player, he cannot increase his odds of being elected above the minimal probability of $1/n$ (minimal, since a faulty player can behave honestly). In our notation, $e(1) = 1 - 1/n$. In this protocol player 1 selects a random player i among players $2, \ldots n$. Player i then announces 1 as a leader with probability $1/n$, and otherwise elects a random player different from 1 and himself as a leader. We see that ideally every player is a leader with probability $1/n$. Faulty player 1 cannot increase his odds of being a leader, since honest player i will elect him as a leader with probability $1/n$ (however, 1 can increase chances of any $j \neq 1$ to be a leader by never selecting $i = j$). And any other player i can do any harm only if player 1 selects i, but then i cannot elect himself as a leader (but can elect anyone else). Above example illustrates a peculiar feature of the leader election problem in that the "success" (leader is not faulty) depends on the collection of faulty players B that good players do not even know exist! For example, in the protocol above a faulty player can severely bias from $1/n$ the probabilities with which honest players get selected as leaders (in fact, this must be the case by Theorem 1). But this still constitutes a good leader election protocol, as long as the faulty player cannot increase his own odds of election.

The theorem below (referred thereafter as the "Main theorem") states the main best known results about collective coin-flipping and leader election. We will discuss these and other related results more precisely in the next sections.

Theorem 2 (Main Theorem) *The following are the best known results concerning collective coin-flipping and leader election.*

1. *$e(b) = 1 - \Theta(\frac{b}{n})$ and $a(b) = \frac{1}{2} - \Theta(\frac{b}{n})$. [BL90, S89, AN93]*
2. *There are no coin-flipping or leader election protocols for $b \geq n/2$. [S89, BN]*

   For the next three items we assume that $b < n/2$ and $b = (1-\delta)n/2$.
3. *Coin-flipping and leader election are equivalent in terms of resilience probability: $a(b) = \Theta(e(b))$. [F99]*
4. *$e(b) = \Omega(\delta^{1.65})$ and $e(b) = O(\delta^{1-\varepsilon})$, for any $\varepsilon > 0$ (same results hold for $a(b)$ by 2. above). [F99]*
5. *For any $\delta > 0$ there exist b-resilient leader election and coin-flipping protocols proceeding either in $\log^* n + O(1)$ rounds and $O(\log n)$ bits per round, or in $O(\log n)$ rounds with 1 bit per round. [RZ98, F99]*
6. *There are no $\Omega(n)$-resilient coin-flipping protocols proceeding in $(\frac{1}{2} - \varepsilon)\log^* n$ rounds (for any $\varepsilon > 0$) with 1 bit per round. [RSZ99]*
7. *There are no $\omega(n/\log n)$-resilient one-round 1 bit per round coin-flipping protocols, as well as $\Omega(n)$-resilient one-round $o(\log n)$ bits per round coin-flipping protocols. [KKL89]*
8. *There exist $\Omega(n/\log^2 n)$-resilient one-round 1 bit per round coin-flipping protocols. [AL93]*

3 Coin-Flipping vs. Leader Election

Let us start by observing that leader election is "at least as difficult" as coin-flipping. Namely, any protocol for leader election can be used for coin-flipping as well by letting the elected leader flip the coin. We note that when the honest leader is selected, he will indeed flip a random coin. In our notation this means that $a(b) \geq \frac{1}{2}e(b)$. In particular, b-resilient leader election implies b-resilient coin-flipping (as we will see later, the "converse" is also true).

We note a trivial upper bound that $e(b) \leq 1 - \frac{b}{n}$. Indeed, faulty player can simply behave honestly, and some b honest players must be elected with probability at least b/n. Ben-Or and Linial [BL90] elegantly showed a similar upper bound for coin-flipping; namely, $a(b) \leq \frac{1}{2} - \Omega(\frac{b}{n})$. On the lower bound front, a series of papers showed protocols proving that $e(b) \geq 1 - O(\frac{b}{n})$ for larger and larger values of b. Ben-Or and Linial did it for $b = O(n^{0.63})$. Ajtai and Linial [AL93] implicitly showed it (by applying the same transformation as [BL90] to the functions they construct. Russell and Zuckerman [RZ98] made this connection explicit) for $b = O(n/\log^3 n)$. Ajatai and Linial [AL93], improving the analysis of Saks [S89], showed it for $b = O(n/\log n)$ and, finally, Alon and Naor [AN93] showed it for all b. We will talk about these protocols more a bit later. As we just observed, since $a(b) \geq \frac{1}{2}e(b)$, we get $a(b) \geq \frac{1}{2} - O(\frac{b}{n})$. Collecting all these results, we get part 1. of the Main theorem; namely, $e(b) = 1 - \Theta(\frac{b}{n})$ and $a(b) = \frac{1}{2} - \Theta(\frac{b}{n})$.

This might seem to be the end of the story, but there are a lot of important questions left, as is seen from the other results in the Main theorem. The most basic is what is the maximum number of faulty players we can tolerate (aside from being linear in n). The next observation (implying part 2. of the Main theorem) made by Saks [S89] (formal proof appears in [BN]) is that $n/2$ faulty players can completely control the outcome of the coin flip in any coin-flipping protocol.

Lemma 1 ([S89]) *For any coin-flipping protocol Π, if B_0 and B_1 is an arbitrary partition of the n players, either $p^0_\Pi(B_0) = 1$, or $p^1_\Pi(B_1) = 1$. In particular, $a(b) = 0$ for $b \geq n/2$.*

This is shown by first transforming Π into a protocol where at each round only one player moves (which only increases the resilience), and then using the induction on the number of rounds. By making $|B_0| = |B_1| = n/2$, we see that there are no $(n/2)$-resilient coin-flipping protocols. Same result holds for the leader election problem as well by our previous observation (if not, a coin-flipping protocol which first elects a leader who then flips a coin would have non-zero resilience probability as well). Thus, $n/2$ faulty players can guarantee that one of them will be chosen as a leader.

We note that the result of Lemma 1 cannot be strengthened. For example, a "majority" coin flip (where each player broadcasts a random bit and their majority is the resulting coin) produces both 0 and 1 with positive probability when $b < n/2$ (as $n/2+1$ honest players can be lucky to agree on this value. Of course, this could happen with probability as tiny as $2^{-n/2-1}$). The big question comes whether resilient protocols exist for $b < n/2$. We start by considering coin-flipping protocols first.

Traditionally (dating back to [BL90] who introduced the problem), protocols for collective coin-flipping are split into two classes: one-round protocols vs. general (many-round) protocols. The reason for this is that one-round protocols have a very nice interpretation. Namely, such protocols can be identified without loss of generality with a single boolean function $f : X^n \to \{0,1\}$. All players are supposed to select a random $x_i \in X$, and the resulting coin flip is just $f(x_1, \ldots, x_n)$. Of course, faulty players will first wait to get x_i's from the honest players, and only then set their x_i's. For example, the "majority" protocol above was an example of such a protocol. The treatment of one-round protocols is very elegant and is related to the notion of *influence of variables on Boolean functions*. We will discuss this and the known results for one-round coin-flipping in Section 4.

Unfortunately, we will see that one-round coin-flipping protocols (i.e. boolean functions) are not as powerful as general (many round protocols), so let us turn to those. Interestingly enough, *all* such coin-flipping protocols considered in the literature first elected a leader who then flipped the coin. One might ask whether this usage of leader election protocols for coin-flipping is a coincidence (and one can design significantly better "direct" coin-flipping protocols), or there is no significant loss by going through a seemingly more difficult leader election. This question was beautifully resolved only recently by Feige [F99], who showed the "converse reduction": any coin-flipping protocol tolerating $b < n/2$ players (other cases are not interesting as nothing can be done anyway) can be transformed into a leader election protocol having the same (up to a constant factor) resilience probability. More precisely,

Theorem 3 ([F99]) *For any $b < n/2$, $a(b) = \Theta(e(b))$. Moreover, the transformation from leader election to coin-flipping takes only one extra round where one player sends a single bit. And the transformation from coin-flipping to leader election can either be made with $O(\log n)$ extra rounds with 1 bit of communication per round (per player), or with $O(\log^* n)$ rounds with $O(\log n)$ bits of communication per round (per player).*

We will talk about it in more detail in Section 9, but let us just briefly mention the above transformation from coin-flipping to leader election. Feige showed a protocol for electing *two leaders*, such that (when $b < n/2$) with constant probability γ *at least one of the leaders is non-faulty*. This protocol can be made with the round and bit complexities stated in the theorem. Then the players flip the coin (by running the given collective coin-flipping subroutine Π) which determines who of the two leaders is selected as a final leader. We see that the overall resilience probability of this protocol is at least $\gamma a_\Pi(b) = \Omega(a_\Pi(b))$.

We see that unless we insist on one-round coin-flipping protocol, the tasks of coin-flipping and leader election are essentially equivalent in terms of resilience probability (proving part 3. of the Main theorem). In particular, the resilience thresholds are the same for both problems (later we show that this threshold is essentially $n/2$ indeed). Therefore, in Section 4 we describe the results on one-round coin-flipping protocols (pretty much covering parts 7. and 8. of the Main theorem), while in Section 5 we describe various leader election protocols (each of which gives a corresponding coin-flipping protocol with essentially the same parameters), covering parts 4. and 5. of the Main theorem. In between these sections we mention the "stand-alone" lower bound of [RSZ99] on the number of rounds for a coin-flipping protocol, which is part 6. of the Main theorem.

4 One-Round Coin-Flipping

As we pointed out, one-round coin-flipping has a very nice interpretation. Without loss of generality we can assume that each player i transmits a random element x_i in some set X, and the coin is some function $f(\vec{x}) = f(x_1, \ldots, x_n)$, where $f : X^n \to \{0,1\}$. Thus, one-round protocol is simply a function f. Fix any coalition B of b faulty players. The protocol then proceeds as follows. First honest players transmit random

x_i's for $i \notin B$. Faulty player collect this information, and maliciously set their x_i so as to bias the resulting coin flip $f(\vec{x})$.

4.1 Influence of Variables on Boolean Functions

We now define the notion of *influence* that plays an important role in one-round coin-flipping protocols.

Definition 5 *For a function $f : X^n \to \{0,1\}$, define*

- $I_f^1(B) = p_f^1(B) - p_f^1$, influence of B towards 1.
- $I_f^0(B) = p_f^0(B) - p_f^0$, influence of B towards 0.
- $I_f(B) = I_f^0(B) + I_f^1(B)$, influence of B on f.
- $I_f(b)$ *is the maximum over all sets B of size b of $I_f(B)$.*

Note that after honest players broadcast their values, three things that can happen for a given B. The function f can already be fixed to 1 (say, this happens with probability q^1), it can be fixed to 0 (probability q^0), or f can still be undetermined (probability q^*). We note that $p_f^1(B) = q^1 + q^*$, since faulty players can set f to 1 if it is already set to 1 or is undetermined. Analogically, $p_f^0(B) = q^0 + q^*$. Using this, the definition of $I_f(B)$ and the facts that $p_f^0 + p_f^1 = 1$, $q^0 + q^1 + q^* = 1$, we get

$$I_f(B) = I_f^0(B) + I_f^1(B) = p_f^0(B) + p_f^1(B) - 1 = q^* \tag{1}$$

Thus, influence of B on f, aside from being the sum of its influences towards 0 and 1, is also the the probability of f being still *undetermined* when players not in B choose their inputs at random. This interpretation will be much more convenient to work with. To see the relationship between influence on the function and one-round coin-flipping, we note the following simple Lemma.

Lemma 2

- *If $I_f(b) > 1 - 2\varepsilon$, then f is not (b,ε)-resilient.*
- *If $I_f(b) \le \varepsilon$ and $a_f \ge 2\varepsilon$, then f is (b,ε)-resilient.*

Proof: For the first item, fix B such that $I_f(B) > 1 - 2\varepsilon$. From Equation (1), $I_f(B) = p_f^0(B) + p_f^1(B) - 1$, so $p_f(B) \ge \frac{1}{2}(1 + I_f(f)) > 1 - \varepsilon$, and hence f is not (b,ε)-resilient.

For the second item, take any B. Note that $p_f^1(B) \ge p_f^1 = a_f^1 \ge a_f$. Then $p_f^0(B) = 1 + I_f(B) - p_f^1(B) \le 1 + I_f(B) - a_f \le 1 + \varepsilon - 2\varepsilon = 1 - \varepsilon$. Similarly, $p_f^1(B) \le 1 - \varepsilon$. Thus for any B, $p_f(B) \le 1 - \varepsilon$. ■

When given a particular f with $a_f = \Omega(1)$[1], it is typically the case that there exists a *sharp threshold* $t(n)$ satisfying the following property. If $b(n) = o(t(n))$, we have $I_f(b(n)) = o(1)$, implying (by above Lemma) that f is $b(n)$-resilient. But if $b(n) = \omega(t(n))$, then $I_f(b(n)) = 1 - o(1)$, implying (by above Lemma) that f is not $b(n)$-resilient. Thus, a good one-round coin-flipping protocol essentially reduces to finding f where the "influence threshold" $t(n)$ is as large as possible.

4.2 Results for Sending 1 Bit

Traditionally, the only ground set X considered was $X = \{0,1\}$, i.e. $f : \{0,1\}^n \to \{0,1\}$, and we concentrate on this case as well for now, addressing the more general case later. A particularly interesting case corresponds to $|B| = 1$, i.e. influence of a single (boolean) variable on a function f (or influence of a single faulty player on the coin flip). This case has also a nice interpretation in *game theory*, which typically assumes only one faulty player. Also, it will be a building block for proving lower bounds for influences of larger sets. Finally,

[1] In fact, we usually consider f with $p_f^0 \approx p_f^1 = \frac{1}{2} + o(1)$, since we want a nearly perfect coin flip when all the players are honest. Sometimes, this is even made a requirement for coin-flipping protocols.

we note that for $B = \{i\}$ (and $X = \{0,1\}$), we have $I_f^0(\{i\}) = I_f^1(\{i\}) = \frac{1}{2} I_f(\{i\})$ (because, if f is unfixed and there is only one bit to set, one of the settings is $f = 0$ and the other is $f = 1$).

The next two theorems summarize the best known lower and upper bounds about $f : \{0,1\}^n \to \{0,1\}$.

Theorem 4 *Take* any $f : \{0,1\}^n \to \{0,1\}$. *Then*

a. *There is* monotone g *with* $p_g^1 = p_f^1$ *such that for all sets* B,

$$I_f^0(B) \geq I_g^0(B),\ I_f^1(B) \geq I_g^1(B),\ I_f(B) \geq I_g(B)$$

so the least "influenced" functions are monotone. [BL90]

b. *There exists a variable* i *with influences* $I_f^0(\{i\}) = I_f^1(\{i\}) = \frac{1}{2} I_f(\{i\}) \geq a_f \cdot \Omega(\frac{\log n}{n})$. *[KKL89]. Thus, if* $a_f = \Omega(1)$, *we get* $I_f(1) = \Omega(\frac{\log n}{n})$.

c. *Above result implies that for every* $b < n/2$ *we have*

$$a_f(b) \leq \left(1 - \Omega\left(\frac{\log n}{n}\right)\right)^b \cdot a_f \tag{2}$$

In particular, some $\omega(n/\log n)$ *variables control the outcome of* f *with probability* $1 - o(1)$, *so* no function f is $\omega(n/\log n)$-resilient. *[KKL89]*

d. *If* $b > \beta n$ *where* $\beta > \frac{1}{3}$, *then* $a_f(b) \leq 2^{-\Omega(n)}$. *[RSZ99]*

e. *If* $b \geq n/2$, *then* $a_f(b) = 0$, *so some outcome can be forced. [S89]*

Theorem 5 *There exist (different for parts a,b) functions* $f : \{0,1\}^n \to \{0,1\}$ *with* $p_f^0 \approx p_f^1 = \frac{1}{2} + o(1)$ *and*

a. *All variables* i *have influences* $I_f(\{i\}) = O(\frac{\log n}{n})$. *[BL90]*

b. *All sets* B *of size at most* $O(\frac{n}{\log^2 n})$ *have influences* $I_f(B) = O((|B| \log^2 n)/n)$ *([RZ98], extending the analysis of [AL93]). In particular, there exist* $\Omega(n/\log^2 n)$*-resilient functions. [AL93]*

Let us briefly comment on the above results in their chronological order. Ben-Or and Linial [BL90] introduced the notions of influences and using boolean functions for constructing one-round coin-flipping protocols. They constructed a simple function where each variable has influence $O(\log n/n)$ (Theorem 5.a). Essentially, you split the input into blocks of size roughly $\log n$, and make $f = 1$ if at least one block of the input is the all-1 block. Using the isoperimetric inequality, they showed that every (non-trivial) function should have a variable with influence $\Omega(1/n)$, and conjectured that the right answer is $\Omega(\log n/n)$, matching the example above. The conjecture was affirmatively resolved by Kahn, Kalai and Linial [KKL89] (Theorem 4.b). They used a beautiful connection between Fourier analysis and the influences of variables on *monotone* functions. The restriction to monotone was without loss of generality, since already Ben-Or and Linial observed that the least "influenced" functions are monotone (Theorem 4.a. Essentially, if there is an "1-0"-edge violating monotonicity, simply swap it, and keep doing it until the function is monotone). I will not go into details, but essentially the formulas for Fourier coefficients for monotone f and the influences of individual variables are very similar.

The result of Kahn, Kalai and Linial immediately implies the following iterative process to find a set of k variables that can essentially force f to some value (as stated in Theorem 4.c). Without loss of generality assume $a_f = a_f^1 \leq a_f^0$. Find an influential variable with $I_f(\{i\}) \geq a_f^1 \cdot \Omega(\log n/n)$. Without loss of generality, assume $i = n$. Define f_1 on $x_1, \ldots x_{n-1}$ as follows. For all assignments to $x_1, \ldots, x_{n-1}$ that force f to 1 (irrespective of x_n), make $f_1 = 1$. Otherwise, make $f_1 = 0$. In other words, we simply fix f to 0 when the assignment to $x_1, \ldots, x_{n-1}$ leaves f undetermined (and we know, there is $I_f(\{n\}) = a_f^1 \cdot \Omega(\log n/n)$ fraction of such assignments). If is not hard to argue that $a_f^1(1) \leq a_{f_1}^1 \leq (1 - \Omega(\log n/n)) a_f^1$. Repeating

this $b < n/2$ times we get Equation (2). This implies that no function is $\omega(n/\log n)$-resilient (part 7. of the main Theorem) and brings us back to the main question we started from: what is the maximum resilience of a boolean function we can hope for?

Ben-Or and Linial [BL90] defined the "iterated majority of 3" function that was resilient against $\Omega(n^{\log_3 2}) \approx \Omega(n^{0.63})$ players. Finally, Ajtai and Linial [AL93] proved the existence of a function that is $\Omega(n/\log^2 n)$-resilient (Theorem 5.b and part 8. of the Main theorem), by proving that all the sets of size $\varepsilon n/\log^2 n$ have influence at most ε (thus, resilience follows from Lemma 2). Their function will be discussed in more detail in Section 7. Unfortunately, it is non-constructive, as the existence is proven via the probabilistic method. We point out that [RZ98] somewhat extended their analysis to all the sets of size at most $n/\log^2 n$ (as stated in Theorem 5.b), and used it for a constant-round leader election protocol, discussed later.

Finally, Theorem 4.e was already observed in Lemma 1, and Theorem 4.d is a trivial observation that can be easily proved using Chernoff bound. Still, it improves the "inverse polynomial" bound that follows from Equation (2) for $b > n/3$ to an exponential bound.

To summarize, the optimum resilience threshold for one-round coin-flipping (where each player sends one bit) lies somewhere between $n/\log^2 n$ and $n/\log n$. In particular, *linear coalitions cannot be tolerated in one round/one bit schemes.*

4.3 Sending More than 1 Bit

We conclude by briefly talking about players sending more than 1 bit, i.e. $f : X^n \to \{0,1\}$, where $X = \{0,1\}^c$. We note that if such function is $b(n)$-resilient, then if we let $N = nc$ and make c new players simulate one original player, we get a $b(N/c)$-resilient function $f' : \{0,1\}^N \to \{0,1\}$. Turning it backwards, if there is no $b(n)$-resilient function $f : \{0,1\}^n \to \{0,1\}$, then there is no $b(nc)$-resilient function $f : X^n \to \{0,1\}$, where $X = \{0,1\}^c$. By applying Theorem 4.c, we get (see also part 7. of the Main theorem)

Corollary 1 *There is no $\Omega(n)$-resilient function $f : X^n \to \{0,1\}$, where $X = \{0,1\}^{o(\log n)}$. [KKL89, RSZ99]*

It is an open question, first raised only in [RSZ99], of whether there are one-round (or even constant round) $\Omega(n)$-resilient protocols and nothing is known beyond the Corollary above.

4.4 Lower Bound on the Round Complexity of Coin-Flipping

Before jumping to leader election, we mention another nice lower bound for coin-flipping; this time, for the number of rounds (it also holds for leader election, but looks much less interesting in that setting). Russell, Saks and Zuckerman [RSZ99] showed that (part 6. of the Main theorem)

Theorem 6 ([RSZ99]) *Every $\Omega(n)$-resilient coin-flipping protocol where players send one bit per round must take $(\frac{1}{2} - \varepsilon)\log^* n$ rounds, for every $\varepsilon > 0$.*

Not surprisingly, one of the main component of this result is the strengthening of the result of Kahn, Kalai and Linial [KKL89]. Namely, Russell, Saks and Zuckerman showed that when $b \geq \Omega(n/\log n)$, not only there are sets of size b that can essentially fix a function f, but there are "a lot" of such sets. In particular, a random set of size b has a "decent" chance of being that influential. The quotes are because these numbers are quite small, but so is the $\log^* n$ bound which is being shown. Then a tricky induction on the number of rounds completes the proof. We will discuss this result more in Section 8.

5 Leader Election

We now turn to describing the protocols for leader election. Figure 5 (augmented from [RZ98]) briefly illustrates some of the the results in their chronological order.

Source	Threshold	Rounds	Bits/Round/Player	Constructive?
[BL90]	$O(n^{0.63}/\log n)$	1	$\log n$	Yes
[S89]	$O(n/\log n)$	n	$\log n$	Yes
[AN93]	$O(n)$	$n^{O(1)}$	1	Yes
	$(\frac{1}{3}-\delta)n$	$O(n)$	1	No
[BN]	$(\frac{1}{2}-\delta)n$	$O(n)$	1	No
[CL95]	$O(n)$	$(\log n)^{O(1)}$	1	Yes
[ORV94]	$O(n)$	$O(\log n)$	$n^{O(1)}$	Yes
	$(\frac{1}{2}-\delta)n$	$O(\log n)$	$n^{O(1)}$	No
[RZ98]	$(\frac{1}{2}-\delta)n$	$\log^* n + O(1)$	$\log n$	Yes+hard
	$(\frac{1}{2}-\delta)n$	$\log n$	1	Yes+hard
([AL93])	$O(n/(\log^{(r)} n)^3)$	r	$\log n$	For $r \geq 3$
[F99]	$(\frac{1}{2}-\delta)n$	$\log^* n + O(\log 1/\delta)$	$\log n$	Yes+simple!
	$(\frac{1}{2}-\delta)n$	$\log n + O(\log 1/\delta)$	1	Yes+simple!

Figure 1: Various Protocols for Leader Election.

5.1 One-Round Leader Election using Coin-Flipping

We said that all interesting coin-flipping protocols, except for one-round protocols, are done through leader election. Ironically, the very first leader election protocol of [BL90] was done using one-round coin-flipping. Namely, assume that f is a b-resilient function (with expectation 1/2 for simplicity, even though it is not important) such that $p_f(t) = \frac{1}{2} + O(t/b)$ (all resilient functions we considered are of this form). Then performing in parallel $\log n$ coin flips to determine the leader turns out to be $(b/\log n)$-resilient *one-round* leader election protocol. Indeed, the probability of choosing a bad leader with any t faulty players can be upper bounded (using the union bound) by

$$t\left(\frac{1}{2} + O\left(\frac{t}{b}\right)\right)^{\log n} = \frac{t}{n}\left(1 + O\left(\frac{t}{b}\right)\right)^{\log n} = O\left(\frac{t}{n}\right) = o(1)$$

provided $t = O(b/\log n)$. Since the best resilient function Ben-Or and Linial constructed was $\Omega(n^{0.63})$-resilient, this is roughly the best leader election they got. However, if one uses $\Omega(n/\log^2 n)$-resilient function of Ajtai and Linial [AL93], we get one-round $\Omega(n/\log^3 n)$-resilient leader election. This is *exactly* the one-round leader election protocol of Russell and Zuckerman [RZ98] mentioned in Figure 5 (for $r = 1$).

We note, however, that since there are no $\omega(n/\log n)$-resilient functions, this method can never give $\omega(n/\log^2 n)$-resilient leader election. The subsequent "direct" protocols described next easily beat this bound.

5.2 Initial Protocols

The first elegant protocol for leader election was proposed by Saks [S89]. The protocol is called *baton passing*. It starts with player 1 holding the baton. He then randomly selects another player and passes him the baton. This player randomly selects the next player who still has not had the baton, and so on. The last player to hold the baton is declared as a leader. We see that if everybody is honest, each player except player 1 is selected as a leader with probability $1/(n-1)$. The intuition behind this protocol is the following. Each round can be viewed as eliminating another player. Clearly, faulty players *always* pass the baton to an honest player, in order to deterministically eliminate an honest player (it can be easily shown to be an optimal adversarial strategy). But then this honest player will select a player at random, so there is a reasonable chance he will eliminate a faulty player. Saks analyzed this protocol and showed that its resilience threshold is $\Theta(n/\log n)$. Ajtai and Linial [AL93] found the threshold more exactly; in particular we can make $b(n) = \frac{n}{(2+\varepsilon)\log n}$ for any $\varepsilon > 0$. We will talk about it some more in Section 7.

The big question (raised already by Ben-Or and Linial [BL90]) for some time was whether there are leader election protocols tolerating linear-size coalitions. The first affirmative result for this question was given by Alon and Naor [AN93]. They addressed the question from a generic standpoint. We know that without loss of generality any protocol can be transformed in to one where at each round exactly one player moves. Moreover, this player simply sends a single random bit. Thus, any protocol can be viewed as a complete binary tree of depth d (where d is the number of rounds), where each internal node as well as each leaf are labelled by a player (the player who moves now for the internal nodes and the elected leader for the leaves). Thus, the question is whether there exists a complete binary tree of finite depth d and a labeling of its nodes by numbers from 1 to n such that the resulting leader election protocol tolerates a linear number of faulty players. Instead of looking for explicit protocols (which they also did later in this paper), they simply constructed a random such tree (i.e. each node is labelled by a random player), and argued that with high probability this tree (for $d = O(n)$) defines a leader election protocol that is $(n/4)$-resilient. Adding a few hacks, they moved the resilience threshold to $(\frac{1}{3} - \delta)n$. Boppana and Narayanan [BN] showed that these hacks were not necessary by improving the analysis of Alon and Naor. In particular, they showed that with $d = O(n)$, the original random tree with high probability forms a leader election protocol tolerating $(\frac{1}{2} - \delta)n$ faulty players, for any $\delta > 0$. In light of Lemma 1, this result shows that the optimal resilience threshold for leader election (and coin-flipping) is essentially $n/2$. Unfortunately, the random tree protocol is non-constructive. It also takes $d = O(n)$ rounds.

Alon and Naor [AN93] also described a *constructive* protocol for leader election tolerating βn players for some very small constant β. This construction is quite complicated. However, it follows a very common paradigm used in most of the successive leader election protocols. Namely, at each stage they considerably reduce the number of players who have a chance of becoming a leader. This process is called *electing a committee*. With high probability, the committee should not have a significantly larger fraction of faulty player than are present originally. Then we typically recurse on this committee, usually stopping when we reach some constant or $\log n$, at which point we can use the non-constructive protocol of Alon and Naor.[2]

Cooper and Linial improved the constructive protocol of Alon and Naor so that it now takes "only" $O(\log^{17} n)$ rounds (with 1 bit per round). This protocol is also quite complicated and tolerates $O(n)$ players with a tiny constant in front of n.

A better round complexity of $O(\log n)$ (but polynomial in n communication complexity) was achieved by Ostrovsky, Rajagopalan and Vazirani [ORV94]. They split the protocol into two parts. In the first part they find a committee of size $O(\log n)$, after which they can either recurse, or use the sequential protocol of Alon and Noar. The first stage can either be done constructively using random walks on expander graphs (and tolerating only $O(n)$ bad player's with a small constant in front), or done non-constructively tolerating the optimal threshold of $(\frac{1}{2} - \delta)n$. The disadvantage is high communication complexity in each round.

The next important paper significantly improving the round and bit complexities for leader election was the paper by Russell and Zuckerman [RZ98]. The main result of the paper is an *explicit* $(\log^* n + O(1))$-round protocol tolerating $(\frac{1}{2} - \delta)n$ bad players. In each round the number of players is reduced from n to $(\log n)^c$ (implying the claimed number of rounds; here c is some large constant depending on δ). The communication is only $O(\log n)$ bits per round. This is done by constructing an appropriate explicit family of allowable committees of size polynomial in n, and then using a special one-round sampling protocol they developed. Both of these steps are very non-trivial and complicated (for example, the set of allowable committees is constructed using an extractor and the sampling uses explicit constructions of hitting sets for combinatorial rectangles). Other protocols of Russell and Zuckerman mentioned in Figure 5 are done similarly. We note that the constant round protocol uses the resilient function of Ajtai and Linial [AL93] at the last round. Protocols of [RZ98] already show part 5. of the Main theorem, but an alternative simpler way is coming in the next section.

[2] As we can already see, this paradigm is really terrible for adaptive adversaries, see Section 10.

5.3 Simple Protocols of Feige [F99]

The reason I went through all these details is to compare the above protocols with the recent beautiful and simple protocol of Feige [F99]. In particular, we note the following undesirable features of any of the previous protocols tolerating $(\frac{1}{2} - \delta)n$ faulty players:

- The protocol depends on the knowledge of $\delta > 0$.
- The resilience probability $e(b)$, while constant for every constant δ, is extremely small. For example, at best it decays exponentially fast (even worse) in $1/\delta$.
- The protocols are either non-constructive, or are extremely complicated.

Fiege eliminated all the above features by giving very simple and efficient leader election protocols. We will talk about them more in Section 9, but let us see the flavor. The main idea is as simple as the baton passing idea of Saks. It is called the *Lightest Bin protocol* (denoted by LB). In its simplest form, each player sends a random bit. Imagine that the player puts a ball with his name in bin 0 if his bit is 0, and in bin 1 otherwise. The "lightest" bin is chosen (i.e. the bin with fewer balls), and the players who put their names in this bin continue to elect the leader among them recursively. Intuitively, if the number of players is large, the bins are almost perfectly balanced after only honest players throw their balls. Faulty player cannot put too many balls into one bin, since this bin is going to be heavier then. The optimal strategy for them is to make the final bins of essentially the same size, since this allows them to put the largest number of bad players into the lightest bin. But then, since honest players are balanced out, the faulty players are forced to split essentially in half as well. Thus, we almost do not increase the *fraction* of bad players, while halving the number of active players.

With a few minor details, this protocol alone gives $\log n$ rounds, 1 bit per round, and tolerates the optimal fraction of $(\frac{1}{2} - \delta)n$ faulty players. In addition, it does not depend on δ, and even its success probability of $\delta^{O(\log 1/\delta)}$ is way better than previously known protocols. The protocols can be done more aggressively by letting each player transmit more than one bit, i.e. by having each player put his ball randomly into one of ℓ bins, and then selecting the lightest bin among those. As it turns out, we can afford to have up to $n/\log(n)^{O(1)}$ bins, i.e. to go from n down to $(\log n)^{O(1)}$ players in a round, and still almost preserve the fraction of bad players. This collapses the number of rounds to $) \log^* n + O(\log 1/\delta))$ with $O(\log n)$ bits per round. It also matches the best results of Russell and Zuckerman (but is much simpler and more efficient), as well as shows part 5. of the Main theorem again.

Feige did not stop at this. In particular, the success probability of current leader election protocols, $e(b)$, is a very small function of δ (even using Feige's LB protocol). Feige studied for the first time the behavior of $e(b)$ as a function of δ (recall, $b = (\frac{1}{2} - \delta)n$). First, he generalized the leader election problem to the one of electing a committee of c players among which at least one is good (leader election corresponds to $c = 1$). Using the generalized LB protocol, he efficiently solved this problem in a similar manner to standard leader election. In particular, it could tolerate up to $n(c - \delta)/(c + 1)$ cheaters (for any $\delta > 0$, which he showed is optimal) and the probability of error is the function of δ only. The coolest special case was for $c = 2$ and $b < n/2$, where we get

Lemma 3 ([F99]) *If $b < n/2$, there is a protocol electing with at least a constant probability a good committee of size two (i.e. a committee with at least one of the two players not faulty).*

We already saw how this Lemma is used to show the equivalence of coin-flipping and leader election in Theorem 3 (and part 3. of the Main theorem). Now we will see how it gives a much better success probability for leader election. The new protocol first selects the committee of size 2 as above, and then lets the two players elect a leader among the original n players. We will discuss this beautiful sub-protocol in Section 9, but point out that he manages to achieve the probability of electing a good leader (provided one of the two players is honest) to be $\Omega(\delta^{1.65})$. Using a nice recourse to submartingales, Feige also showed that the probability of electing a good leader (equivalently, not avoiding any outcome in coin-flipping) must tend to 0 when δ approaches to 0. Overall (showing part 4. of the Main theorem),

Theorem 7 ([F99]) *For $b = (\frac{1}{2} - \delta)n$, $e(b) = \Omega(\delta^{1.65})$ and $e(b) = O(\delta^{1-\varepsilon})$ (for any $\varepsilon > 0$).*

6 Summary, Open Problems and Other Directions

We saw that leader election and coin-flipping are essentially equivalent in terms of resilience probability. For sub-linear coalitions, the best bounds to use are $e(b) = 1 - \Theta(b/n)$ and $a(b) = \frac{1}{2} - \Theta(b/n)$. When $b = (\frac{1}{2} - \delta)n$, the best resilience probability is inverse polynomially related to δ, while $n/2$ or more players cannot be tolerated. The protocols can be made very efficient (e.g., having $\log^* n + O(1)$ rounds with $O(\log n)$ bits per round or $\log n$ rounds with 1 bit per round). For one-round coin-flipping with 1 bit per round, the optimal resilience threshold is somewhere between $n/\log n$ and $n/\log^2 n$. Moreover, linear resilience cannot be achieved with fewer than $\log^* n$ rounds and 1 bit per round (but can be achieved with $\log n$ rounds) or with $o(\log n)$ bits in one round. Therefore, I feel the main interesting open questions in the classical setting concern the round and bit complexities of coin-flipping (and leader election). Here they are:

- Show there exist $\Omega(n/\log n)$-resilient functions $f : \{0,1\}^n \to \{0,1\}$ (if this is true). At least, find an *explicit* function which is $\Omega(n/(\log n)^{O(1)})$-resilient.
- Is it possible to do $\Omega(n)$-resilient coin-flipping (or leader election) in one round (with maybe high communication complexity)? In other words, are there $\Omega(n)$-resilient functions $f : X^n \to \{0,1\}$?
- What is the round complexity for $\Omega(n)$-resilient coin-flipping with 1 bit per round? Is it closer to $\log^* n$ or to $\log n$?

There are also other interesting problems in the full-information model beside collective coin-flipping and leader election. For example, Goldreich, Goldwasser and Linial [GGL98] considered general multi-party computation in the full-information model. In this setting, players try to compute probabilistic function $f(x_1, \ldots, x_n; r)$, where they now have inputs which are selected uniformly at random. The protocol is resilient if the adversary cannot force some output with a significantly higher probability than by modifying his inputs. Goldreich et al. have some nice results for bivariate functions, and observe that the general question seem to be quite difficult. However, a more immediate generalization of coin-flipping to the problem of collective sampling (equivalently, probabilistic function computation with no inputs) seems more tractable.

Without loss of generality, this question can be viewed as the one of generating uniformly at random an ℓ-bit string for some $\ell \geq 1$. Coin-flipping simply corresponds to $\ell = 1$. Reducing to this case, Goldreich, Goldwasser and Linial found a sampling protocol for any ℓ, which is much "better" than ℓ applications of the coin-flipping protocol. The resilience here is defined very generally. For *every* subset $S \subseteq \{0,1\}^\ell$, we want the adversary (controlling b players) not to be able to force the sample to be in S with probability significantly larger than the *density* $\rho(S) \stackrel{\text{def}}{=} |S|/2^\ell$ of S. From Theorem 1, for any S the adversary can force the sample to be in S with probability at least $\rho(S)^{1-b/n}$. Goldreich, Golwasser and Linial showed (by a non-trivial reduction to coin-flipping, at which stage any resilient coin-flipping protocol, like that of Alon and Naor [AN93], suffices) that

Theorem 8 ([GGL98]) *There exists a collective sampling protocol where for any set S, any b faulty players can force the outcome to be is S with probability at most*

$$O(\log(1/\rho(S)) \cdot \rho(S)^{1-O(b/n)})$$

In particular, for two parties one of whom is faulty, there is a simpler sampling protocol where the sample falls in S with probability at most $O(\rho(S)^{1/4})$.

The last result was used by Feige [F99] in his final leader election protocol as we will see in Section 9. Another *one-round* sampling algorithm with incomparable probability of error (roughly, $O(|S| \cdot 2^{-\Omega(\ell(1-b/n))})$) was designed by Russell and Zuckerman [RZ98] on a way to their leader election protocols.

The final direction that I found interesting is to talk about *adaptive adversaries*. This does not make sense for leader election (adversary can always corrupt the elected leader), but is very interesting for coin-flipping. None of the known coin-flipping protocols apply to this case (in all of them, corrupting at most two players allows to control the coin flip). In particular, how many faulty players can we tolerate? See Section 10 for more on this problem.

7 Summary of the Paper of Ajtai and Linial [AL93]

The paper contain two very different results. The first one is the tight analysis of the baton passing game to show that the threshold is at least $n/((2+\varepsilon)\log n)$. The second one is much more interesting and proves (via probabilistic method) the existence of functions $f : \{0,1\}^n \to \{0,1\}$ which are $\Omega(n/\log^2 n)$-resilient. Let me briefly discuss these results separately.

7.1 Baton Passing

Recall the baton passing protocol introduced by Saks [S89]. Starting from player 1, the player holding a baton gives it to a random player who has not had it yet. The last player is the leader. We already pointed out (and it can be easily shown formally) that the best strategy for the faulty players is to always pass a baton to an honest player in order to eliminate him. It is also clear that the adversary should better make the starting player 1 honest as well in order to eliminate him. Because of that and from symmetry, it does not matter which b out of the remaining $n-1$ players the adversary corrupts.

With these comments in mind, denote by $f(s,t)$ the probability that if the current baton holder is honest, and there are s unselected honest and t unselected dishonest players left (initially, $t=b$, $s=n-1-b$), the final leader will be *dishonest*. We then get the recurrence equation

$$f(s,t) = \frac{s}{s+t} f(s-1,t) + \frac{t}{s+t} f(s-1,t-1) \tag{3}$$

with boundary conditions $f(s,1) = 1/(s+1)$ for $s \geq 0$ and $f(0,t) = 1$ for $t \geq 1$. Saks [S89] already observed this recurrence and showed by completely elementary means that for any $\varepsilon > 0$ there are constants $c = c(\varepsilon)$ and $d = d(eps)$ such that: $e_\Pi(b) \leq \varepsilon$ for $b \leq cn/\log n$, and $e_\Pi(b) > \varepsilon$ for $b \geq dn/\log n$. In other words, $\Theta(n/\log n)$ is the resilience threshold. Ajtai and Linial removed this dependence on ε by showing that if $b < n/((2+\varepsilon)\log n)$ (for any $\varepsilon > 0$), then $e_\Pi(b) \geq 1 - O(b/n) = 1 - o(1)$ (note we cannot hope for a better dependence since for any b we have $e(b) \leq 1 - b/n$).

They way they did it is by finding an *exact* solution to the recurrence above. The solution is truly monstrous and takes three pages of pure computations to prove by induction. Asymptotic analysis of this exact solution shows $f(s,t) \leq \frac{9s}{s+t} \sum k \cdot x^k$, where $x = \frac{2t \log s}{s+t} < \frac{2t \log n}{n}$. If $t < n/((2+\varepsilon)\log n)$, then $x < 1$ and $f(s,t) = O(t/(s+t))$ as we need.

7.2 $\Omega(n/\log^2 n)$-Resilient Function

The main part of the paper is a probabilistic construction of $\Omega(n/\log^2 n)$-resilient function with expectation $\frac{1}{2} + o(1)$. As we see from Lemma 2, it suffices to show that each set B of size $\varepsilon n/\log^2 n$ has influence $I_f(B) = O((|B|\log^2 n)/n) = O(\varepsilon)$. Here is a summary of this construction. We follow the mixture of the treatment in [AL93] and [RZ98]. The treatment is a bit technical, but not much can be done about it due to the nature of the construction.

Let us find *block length* c such that $(1-2^{-c})^{n/c} \approx (\ln 2)/n$. It is easy to see that $c = \log n - 2\log\log n + o(1) \approx \log n$ and $2^c \approx n/\log^2 n$. Let $\mathcal{P}$ be set set of all partitions of $[n]$ into n/c disjoint blocks of size c. We write any such partition P as $(P_1, \ldots, P_{n/c})$. Now assume we are given a boolean vector $\vec{x} = x_1, \ldots, x_n$ and an *assignment* $g : [n] \to \{0,1\}$ (of course, g can also be thought as another vector but viewing it as a function is convenient). We say that $\vec{x}$ and g *agree with respect to* P if $\vec{x}$ and g completely agree on at

least one block of P, i.e. for some block P_j we have $x_k = g(k)$, for all $k \in P_j$. We denote this "agreement" function by $agree(\vec{x}, g, P)$.

Now, our resulting function f is very simple. We pick uniformly at random and independently from each other n assignments $g_1, \ldots, g_n$ and n partitions $P^1, \ldots, P^n$.[3] Now define $f(\vec{x}) = f(x_1, \ldots, x_n) = 1$ if and only if for *all* $i = 1, \ldots, n$ we have that x and g_i agree with respect to P^i. Viewed another way, we pick n random and independent assignment/partition tuples, and say that $f(\vec{x}) = 1$ if all tuples "agree" with $\vec{x}$, i.e. each assignment has one of its corresponding partition blocks exactly the same as in the input $\vec{x}$. Define an auxiliary function $f^i(\vec{x}) = agree(\vec{x}, g_i, P^i)$. Then we can summarize our definition as

$$f(x_1, \ldots, x_n) = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n/c} \bigwedge_{k \in P_j^i} (x_k = g_i(k)) = \bigwedge_{i=1}^{n} agree(\vec{x},\ g_i,\ P^i) = \bigwedge_{i=1}^{n} f^i(\vec{x})$$

We note that f also implicitly depends on partitions $\vec{P} = (P^1, \ldots, P^n)$ and assignments $\vec{g} = (g_1, \ldots, g_n)$, but we omit this from the notation. We start from a very strange looking definition, which turns out to be crucial in understanding of what is going on.

Definition 6 *A partition P and a set B are said to* match *if for all $1 \leq \ell \leq c$, the number of blocks P_j of P with $|B \cap P_j| \geq \ell$ is at most*

$$2^\ell \left(\frac{n}{c} \binom{c}{\ell} \left(\frac{|B|}{n} \right)^\ell \right)$$

To de-mystify it at least a little bit, we note that if partition P is selected at random, the probability that certain P_j contains more than ℓ elements of B is at most $\binom{c}{k}(\frac{|B|}{n})^\ell$, so the expected number of blocks is $\frac{n}{c}\binom{c}{\ell}(\frac{|B|}{n})^\ell$. Thus, we say that P and B match if, pretending we choose P at random, the number of blocks that intersect B in ℓ places is at most 2^ℓ more than its expectation. Recall that we want to argue that no set B of size $\varepsilon n / \log^2 n$ has influence on f more than $O((|B| \log^2 n)/n) = O(\varepsilon)$. The proof proceeds in 4 steps:

1. For all $\vec{P}$ and most $\vec{g}$, the expectation of f is $\frac{1}{2} + o(1)$.
2. For any $\vec{P}$, $\vec{g}$ and any B of size at most $\varepsilon n / \log^2 n$, the influence of B on f^i is at most $1/n$.
3. If P^i and B match, then influence of B on f^i is at most $O((|B| \log^2 n)/n^2) = O(\varepsilon/n)$.
4. For most $\vec{P}$ we have that for *every* set B of size $\varepsilon n / \log^2 n$, the number of partitions P^i that do not match B is at most $o(n)$ (in fact, $n/(\log n)^{\omega(1)}$).

First, let us see why these fact suffice. Take any B of size $O(\varepsilon n / \log^2 n)$. Choose variables not in B at random. Since f is undetermined only when at least one of f^i is undetermined, we have (assuming item 4. holds for our random choice of $\vec{P}$):

$$I_f(B) \leq \sum_{i=1}^{n} I_{f^i}(B) \leq o(n) \cdot O\left(\frac{1}{n}\right) + n \cdot O\left(\frac{\varepsilon}{n}\right) = o(1) + O(\varepsilon) = O(\varepsilon)$$

Here we split the summation above into partitions P^i that do not match B (and there are $o(n)$ of those), and the ones that match B (and each contributes $O(\varepsilon/n)$ to the sum).

Let us finish by outlining proofs of steps 1.-4. above. Step 1. is simple. When we pick $x_1, \ldots, x_n$ at random, $f = 1$ if and only if for each of n assignments g_i it is not the case that all n/c blocks of P^i are

[3] We note that the fact that the number of assignments/partitions and the length of our input n are the same is coincidental, n just happens to be sufficient, so we do not introduce a new parameter.

different from the corresponding block of $\vec{x}$ (the last event happens with probability $1-2^{-c}$ for a random g_i), i.e. overall probability is (recall our choice of c)

$$\left(1-\left(1-2^{-c}\right)^{n/c}\right)^n \approx \left(1-\frac{\ln 2}{n}\right)^n \approx \frac{1}{2}$$

Step 2. is simple as well. We note that there are at least $(\frac{n}{c}-|B|)$ blocks of partition P^i that do not intersect B at all. Thus, when we set variables not in B at random, the probability that f^i is *not* fixed to 1 (upper bound on the influence of B on f^i) is at most the probability that all of these disjoint blocks do not match $\vec{x}$, i.e.

$$I_{f^i}(B) \le \left(1-2^{-c}\right)^{\frac{n}{c}-|B|} \le \left(\frac{\ln 2}{n}\right)^{1-\frac{|B|c}{n}} \le \frac{1}{n}\cdot(2^{\varepsilon}\ln 2) \le \frac{1}{n} \tag{4}$$

since $|B|c/n \le \varepsilon/\log n$, and ε is small enough.

Step 3. is a bit more difficult, but it really shows why we care that B and P^i match. We see that the only way f^i is not fixed is when *both* of the following *independent* events happen:

- every P^i_j not intersecting B contains a variable x_k for which $x_k \neq g_i(k)$ (i.e. f^i is not fixed to 1).
- there exists P^i_j meeting B that completely agrees with g_i on all x_k for $k \notin B$ (i.e. f^i is not fixed to 0).

The probability of first event is the same as the estimate we used to see that f^i is not set to 1 for step 2., which is at most $1/n$ from Equation (4). For the second event, take any class P^i_j with $|P^i_j \cap B| = \ell$. Then we get agreement with the remaining $(c-\ell)$ variables not in B with probability $2^{\ell-c}$. Now we use the fact that P^i and B match and the union bound to show that the second probability is

$$\sum_{\ell=1}^{c} 2^{\ell}\left(\frac{n}{c}\binom{c}{\ell}\left(\frac{|B|}{n}\right)^{\ell}\right)2^{\ell-c} = \frac{n}{c2^c}\left[\left(1+\frac{4|B|}{n}\right)^c - 1\right] \approx \frac{n}{c2^c}\frac{O(|B|c)}{n} \approx O\left(\frac{|B|\log^2 n}{n}\right)$$

Multiplying the two events, we get $I_{f^i}(B) = O((|B|\log^2 n)/n^2) = O(\varepsilon/n)$, as claimed.

The final step 4. is technically quite difficult, and that is where most of the work is. It is proven by using the union bound over all sets B of size $B = \varepsilon n/\log^2 n$. Thus, we can concentrate on a particular B and choose partitions at random. Since partitions are independent, we can concentrate on particular B, particular i, and choose partition P^i at random (at the end we use Chernoff bound to see what happens for all n partitions). But now pretty much we need to estimate for a particular B the probability that when we choose random P^i, there is some ℓ such that the number of blocks P^i_j with $|B \cap P^i_j| \ge \ell$ is more than 2^ℓ times its expectation. This estimate is not easy (Markov inequality is too weak for small ℓ and Chernoff cannot be applied directly), but at least we see that we get a pretty direct thing to prove. I omit further details.

8 Summary of the Paper of Russell, Saks and Zuckerman [RSZ99]

The main result of this paper is Theorem 6, which shows that one needs $\Omega(\log^* n)$ rounds for $\Omega(n)$-resilient coin-flipping with 1 bit per round. Because $\log^* n$ is not a very large number, the proof is quite messy in that the authors have to keep track of most of the constants flying around, there are a lot of "towers of exponents", and it is hard to get the intuition of what is going on. I already outlined some intuition in Section 4.4, but let me give few more details which are less messy. In this section we will only talk about 1 bit per round coin-flipping protocols. First, let me restate the main theorem of [RSZ99] in a more convenient "inductive" form (recall, $\log^{(k)} n$ means k iterated applications of log).

Theorem 9 ([RSZ99]) *Let Π be $r(n)$-round (1 bit per round) coin-flipping protocol, where $r(n) \le (\frac{1}{2}-\varepsilon)\log^* n$ for some $\varepsilon > 0$. Then Π is not $b(n)$-resilient, where*

$$b(n) = \omega\left(\frac{r(n)^2}{\log^{(2r(n)-1)} n}\cdot n\right)$$

In particular, Π *is not* $\Omega(n)$*-resilient*

From the above formula we see at least syntactically why we get stuck at $r(n) \approx \frac{1}{2}\log^* n$. The theorem is proven by induction, and there are three main steps to it: the base of the induction, the inductive step, and putting the two above together. The last step is trivial, but is very messy because of all the constants we have to be careful about, so I skip it. Let me talk about the base of the induction, which is an interesting strengthening of the result of Kahn, Kalai and Linial ([KKL89], also Theorem 4.c), and the inductive step, which the only place where we see deal with the round complexity by splitting any $r_1 + r_2$ round protocol into r_1 and r_2 round protocols, and using the inductive hypothesis on those.

By symmetry, we will focus on the probability that B can force 1. We say that Π is α-*nontrivial* if $p^1_\Pi \geq \alpha$, i.e. the natural probability of 1 is at least α. We also say that Π is γ-*powerful* for a given B if $p^1_\Pi(B) > 1-\gamma$ (i.e. $a^0_\Pi(B) < \gamma$). Note that if Π is γ-powerful for *some* B of size b, then Π is *not* (b,γ)-resilient. Hence, it suffices to find such B for any $\gamma > 0$. The way we will do it is by choosing the set B *at random* and arguing that with non-zero probability B is γ-powerful. The following convoluted definition turns out to be the key for the induction.

Definition 7 *Let* $\delta = \delta(r, b, \alpha, \beta)$ *(written as* $\delta(r, b, \gamma)$ *when* $\alpha = \beta = \gamma$*) denote the probability that in* any *r round protocol protocol* Π *that is* α*-nontrivial, a random set* B *of size* b *is* β*-powerful.*

Then the base case of our induction is the following (for convenience, here and after I skip all the ugly constants and use $O(\cdot)$, $\Omega(\cdot)$ notation) result on 1-round protocols:

Lemma 4 ([RSZ99]) *Let* $\gamma \in (0; \frac{1}{2})$ *and* $\gamma b \geq \Omega(n/\log n)$*. Then*

$$\delta(1, b, \gamma) \geq \frac{1}{2}\left(\frac{b}{4n}\right)^{2^{O(n/\gamma b)}}$$

Despite a seemingly tiny probability, it can be viewed as a strengthening of the result of Kahn, Kalai and Linial that every function f has $\omega(n/\log n)$ influential variables (take $\gamma = o(1)$). So not only there are influential sets, but there are "a lot" of them. Recall how Kahn, Kalai and Linial get their result. They show that every boolean function has an influential variable. Then they define the function f_1 on the remaining $(n-1)$ variables by fixing f to, say 1, whenever f was still undetermined when this influential variable is not set. Then they continue this process, defining $f_2, f_3, \ldots, f_d$ on fewer and fewer variables whose natural probability of 1 becomes larger an larger. The main idea of Russell, Saks and Zuckerman was that we do not need to find an influential variable *at every step*. Rather, if a variable is "really influential", we will take it and proceed as above. Otherwise, it does not hurt to pick a *random* variable and proceed as above. Assume for simplicity and without loss of generality (Theorem 4.a) that f is monotone. Then, the new process of generating $f_1, \ldots, f_d$ can be described as follows. We will iteratively find variables $v_1, \ldots, v_d$ of our function f and define f_k to be the function of $(n-k)$ remaining variables other than $v_1, \ldots, v_k$, obtained from f by fixing $v_1 = \ldots = v_k = 1$. Given $v_1, \ldots, v_k$, we select v_{k+1} as follows (where we choose a convenient s to "balance out" the cases below):

1. If there is a variable v of f_k whose influence towards 1 is at least 2^{-s}, set $v_{k+1} = v$.
2. Otherwise, pick v_{k+1} uniformly at random among the remaining $(n-k)$ variables.

Using some ideas from [KKL89] and a very careful analysis of the above "submartingale", Russell, Saks and Zuckerman showed that

Lemma 5 *If* $\gamma \in (0; \frac{1}{2})$*,* $\gamma d \geq \Omega(n/\log n)$ *and* $s = \Theta(n/\gamma d)$*, then*

$$\Pr\left[\{v_1, \ldots, v_d\} \text{ is } \gamma\text{-powerful in } f\right] \geq \frac{1}{2}$$

Thus, we went from a deterministic process of Kahn, Kalai and Linial to a "semi-random" process where some of the variables are deterministic (rule 1.), while other are random (rule 2.). Now the point is that rule 1. cannot be applied too often, since it is easy to show that $\sum_{k=1}^{d} I^1_{f_k}(v_k) \leq p^1_{f_d} \leq 1$, so at most 2^s of v_k's can have influence towards 1 greater than 2^{-s}. Now, the lemma above allows us to set $d = b/2$. Then we have a "reasonable" chance that when we select b variables in B at random, we will be lucky to get all 2^s "influential" variables we got from rule 1. (the others are random anyway, so we do "get" them). The latter probability can be bounded by $(\frac{b}{4n})^{2^s} = (\frac{b}{4n})^{2^{O(n/\gamma b)}}$, while Lemma 5 says that if we succeed in choosing the correct "semi-random" $v_1, \ldots, v_d$, with probability at least $\frac{1}{2}$ they give us a collection of influential variables, giving the final bound in Lemma 4.

Finally, let us briefly comment on the inductive step (the second, more general, form of it below), since this is the only place where the rounds appear.

Lemma 6 ([RSZ99]) *If $\delta_1 = \delta(r_1, b_1, \gamma_1)$, $\delta_2 = \delta(r_2, b_2, \gamma_2)$, then $\delta(r_1 + r_2, b_1 + b_2, \frac{2\gamma_1}{\delta_2} + \gamma_2) \geq \frac{\delta_1\delta_2}{2}$.*
More generally, if $\delta_2 = \delta(r_2, b_2, \alpha_2, \beta_2)$, $\delta_1 = \delta(r_1, b_1, \frac{\alpha_1\delta_2}{2}, \beta_1)$, then $\delta(r_1+r_2, b_1+b_2, \alpha_1+\alpha_2, \beta_1+\beta_2) \geq \frac{\delta_1\delta_2}{2}$.

It is pretty straightforward, but there are a lot of letters involved that make it look very confusing. The most interesting part of the statement was the crucial choice of the function δ in Definition 7, that makes everything go through. Essentially, given any $(r_1 + r_2)$-round protocol Π (which is $(\alpha_1 + \alpha_2)$-nontrivial), we can define for every possible transcript $\vec{\sigma}$ of the first r_1 rounds, an r_2-round "continuation" protocol $\Pi[\vec{\sigma}]$. Here is how we choose our random set B of size (at most) $b_1 + b_2$ and estimate the chance it will be $(\beta_1 + \beta_2)$-powerful. First, we choose a random B_2 of size b_2 and let B_2 try to operate on the last r_2 rounds. For this particular B_2 there will be a set of transcripts $\vec{\sigma}$ that B_2 particularly "likes" (i.e. where we can apply the hypothesis of the theorem). Thus, given this B_2, our adversary can view the first r_1 rounds of Π as trying to force a "good" transcript $\vec{\sigma}$, which is also an r_1-round protocol with a boolean answer ($\vec{\sigma}$ is good or not). For that we choose a random B_1 of size b_1 and let him try to force the outcome of the first r_1 rounds to be good for B_2. Applying the hypothesis of the theorem to the first r_1 and the last r_2 rounds, we get that $B = B_1 \cup B_2$ will succeed in being powerful with the needed probability. Quantitatively, we would be able to show that at least $\frac{\delta_2}{2}$ fraction of B_2 (of size b_2) makes at least δ_1 fraction of B_1 (of size b_1) such, that B_1 can force with probability $(1 - \beta_1)$ the transcript $\vec{\sigma}$ to be such, that B_2 can force with probability $(1 - \beta_2)$ the continuation protocol of $\vec{\sigma}$ to output 1, i.e. B is $(\beta_1 + \beta_2)$-powerful with probability $\frac{\delta_1\delta_2}{2}$.

As I said, with some hacking applied, Lemmas 4 and 6 prove Theorem 9.

9 Summary of the Paper of Feige [F99]

Big parts of this paper were already summarized earlier (mainly in Section 3 and Section 5.3). Let us briefly recall some of the main features of this paper. Fiege looked at the following generalization of the leader election problem, where players wish to elect a committee of size c that contains *at least one* honest player (leader election corresponds to $c = 1$). Such committees are called *good*. As usual, let b be the number of faulty players, and let $k = n - b$ be the number of honest players, and all the standard definitions for leader election extend naturally to this case. Feige showed (under a slightly different definition than usual[4]) that the committee election problem is not solvable when $k \leq n/(c + 1)$. For $k > n/(c + 1)$ we define the *advantage* of good players to be $\delta = \frac{k(c+1)}{n} - 1$ (for $c = 1$ this is the same δ we talked about before).

Feige defined the following very simple Lightest Bin protocol (denoted LB) for the committee election problem. Roughly, each player puts a ball with his name randomly in the 0-bin or the 1-bin. The players who put their names in the lightest bin proceed to the next round. They stop when the number of players left is at most c. A small technical detail arising when n is not a power of 2: we do not always take the "lightest", but roughly the lightest bin. Namely, if $n = 2(c + 1)i + j$, where $-(c + 1) \leq j \leq c$ and $i \geq 1$, we define $\text{Half}(n, c) = (c + 1)i - 1$. Then, the LB protocols is just this:

[4]Roughly, a protocol is good if when honest player *know* which players are faulty, they should be able to force a good committee with probability 1.

1. Initialize $X = [n]$.
2. Repeat while $|X| > c$:
   (a) Each player in X broadcasts a random bit. Let X_0 be the set of players who broadcast 0, and X_1 be the set of players who broadcast 1.
   (b) If $|X_0| \leq \text{Half}(|X|, c)$, then $X \leftarrow X_0$. Otherwise, $X \leftarrow X_1$.
3. Output X as the committee.

Feige showed that this trivial protocol by itself succeeds with probability independent of n for any $\delta > 0$.

Theorem 10 ([F99]) *When the advantage of good players is δ, LB protocol selects a good committee with probability at least $\delta^{O(\log 1/\delta)}$.*

As we noted, we can collapse the number of rounds (by using more bins) to $(\log^* n + O(\log 1/\delta))$ and $O(\log n)$ bits per round (the basic LB above has $\log(n/(c+1)) \leq \log k \leq \log n$ rounds). As we said, this matches and is way simpler and more efficient than leader election protocols of Russell and Zuckerman [RZ98]. Applying Theorem 10 to $c = 2$ and $b \leq n/2$, we get that $\delta = 3k/n - 1 \geq 1/2$, so we elect a good committee with a constant probability, as stated in Lemma 3 and as used in Theorem 3 (to show that standard leader election and coin-flipping are equivalent in terms of success probability).

The election of 2-player committee is also used to give a leader election protocol with $e(b) = \Omega(\delta^{1.65})$ which is much-much better than the bound of $\delta^{O(\log 1/\delta)}$ we get from using LB protocol with $c = 1$. Here is a beautiful protocol for the two players (at least one of whom is honest) to select a leader among all n players. For convenience (it does not matter) we assume that there are $n + 2$ players: our two selecting players and the other n players, and two players have to choose one of these $n + 2$ players. The protocol is based on the idea of using *monotone circuits for majority*. Since the only important thing for us will be the depth of the circuit, we assume that the circuit is actually a tree. The leaves are labelled (possibly with repetitions) by one of n inputs $x_1, \ldots, x_n$ or by constants 0 or 1, and the only gates are the AND and OR gates. In addition, we assume that the gates are alternating. Valiant [V84] showed (non-constructively) that there are majority circuits of depth roughly $5.3 \log n$ (also, there are constructive circuits of depth $O(\log n)$ based on the AKS sorting network, but we will not care about constructibility soon anyway).

Here is the protocol given such a circuit of depth d (it is not our final protocol, we will change it a bit soon). We call the players the *or*-player and the *and*-player. The protocol proceeds in rounds and starts from the root (output) of the tree. The players then trace a path from the root to a leaf in the following way. If the current gate is the AND gate, the *and*-player chooses at random one of the two incoming edges. Otherwise, the *or*-player chooses at random one of the incoming edges. When a leaf is reached, its label is examined. If it is a variable x_i, player i is selected as a leader, if it is a constant 0, the *and*-player is declared a leader, and if it is a constant 1 – the *or*-player is declared as a leader. That is the whole protocol.

Lemma 7 ([F99]) *If the depth of the circuit is d, one of the two selecting players is honest and the majority of players are honest, an honest player is elected with probability at least $2^{-(d+1)/2}$.*

Proof: Assume the *or*-player is honest. Assign each variable corresponding to an honest player a value 1, and each internal node its corresponding value during the circuit computation. Since the majority of the players are good, the output of the circuit is 1 as well. We select a good leader if and only if the leaf we reach is labelled by 1 (a good player among n players, or an honest *or*-player who is labelled by the constant 1). We elect such a player if, in particular, all the gates we visit along our path from the root to a leaf are labelled by 1 (and we know it holds at the root). Take an arbitrary internal node, and assume it is labelled by 1. If this is the AND gate, then both of its children are labelled by 1. So even though the dishonest *and*-player moves, it does not matter which child he chooses – both are labelled by 1 anyway! For an OR gate, at least one of the children is labelled by 1, and the honest *or*-player will select it with probability at least $\frac{1}{2}$. Since *or*-player moves at most $(d+1)/2$ times, we get the claim of the theorem. The case when the

and-player is honest is handled in the same way, but we now label honest guys with 0 (so the majority is 0). ■

Using the $5.3 \log n$ deep circuit of Valiant [V84], we right away get a bound of $\Omega(n^{-2.65})$, irrespective of δ. This is not a terrible bound (compare, say, with $2^{-n/2}$ we get using majority), but it is not good either since it depends on n. However, it turns out we can make the bound depend on δ only (as well as make it explicit). We use the following technical result used by Valiant [V84] inside his proof.

Lemma 8 ([V84]) *Let T be a full binary alternating* AND/OR *tree with* OR *gates closest to the top, let $\alpha = 1-2(3-\sqrt{5})n/(n-1) \approx 0.24$, and let $-1 \le \delta \le 1$. Label the leaves independently at random with 0 with probability $\alpha + (1-\alpha)(1-\delta)/2$ and with 1 with probability $(1-\alpha)(1+\delta)/2$. If depth of T is $3.3\log(1/\delta)+2t$, then with probability $1-2^{-2^t}$ the circuit outputs 1 if $\delta > 0$ and 0 if $\delta < 0$.*

Now, given a two-player selection protocol with advantage δ, if the players can agree on a random labelling of the leaves of T above, where a leaf is labelled 0 with probability α and by a random candidate otherwise, the circuit could be used instead of the ideal majority circuit in the proof of Lemma 7. Indeed, when we later label honest players by 1 and faulty players by 0 in the proof of Lemma 7 (when *or*-player is honest), the overall probability that a leaf is labelled by 0 is *exactly* as stated in Lemma 8 and $\delta > 0$, so with very high probability the output is 1 (as we need, as majority is honest). The case when the *and*-player is honest is the same but now $\delta < 0$, so the output is 0.

Thus, the only thing is for the players to label the leaves of T as we pointed above. Note, we do not need (and cannot hope to have) a truly random labelling, but as long as we have a somewhat random labelling where a set of very low measure (2^{-2^t}) is avoided with constant probability, we will be done. Using appropriate encoding, the problem can be reduced to one, where two players need to agree on a random string of some length ℓ such that some set S of very small measure is avoided. But this is exactly the question solved by Goldreich, Goldwasser and Linial [GGL98] and mentioned in Theorem 8. Namely, Goldreich et al. showed a two-player sampling protocol for strings of length ℓ where the sample is forced to fall in S (for any subset S of $\{0,1\}^\ell$) with probability at most $\rho(S)^{1/4}$ (where recall that $\rho(S) = |S|/2^\ell$ is the density or the measure of S). We note that this sampling protocol takes ℓ rounds (and $\ell = poly(1/\delta)\log n$ is quite large), but is reasonably simple (roughly, players keep selecting ℓ-dimensional vectors and finding a few inner products), so we do not use any of the "heavy machinery".

Overall, we get $(d+1)/2 \approx 3.3\log(1/\delta)/2 \approx 1.65\log(1/\delta)$. Using Lemma 7, we get that we elect a good leader with probability $\Omega(2^{-1.65\log(1/\delta)}) = \Omega(\delta^{1.65})$, exactly as claimed in Theorem 7.

Finally, Feige gave an elegant prove using submartingales that $e(b) = O(\delta^{1-\varepsilon})$ for any $\varepsilon > 0$ (see Theorem 7), but I will not go into details. Feige also gave a lot of toy examples and intuition about the problem. Overall, I think it is a fantastic paper, killing a lot of open questions in the area of collective coin-flipping and leader election.

10 Coin-Flipping with Adaptive Adversaries

Finally, we move to the question of *adaptive adversaries*, i.e. the adversary can observe the messages broadcast by all the players and corrupt up to b players in the course of the protocol. First of all, the problem of leader election does not make sense in this model – the adversary can always corrupt the leader at the end. This is because of the peculiar feature of the leader election that the success depends on the coalition B of faulty players. However, collective coin-flipping still makes perfect sense. Namely, all the notions we talked about in Section 2 (like $a_\Pi(b)$) extend naturally.[5] Thus, b-resilient coin-flipping still roughly means that an adversary, who can corrupt up to b players, cannot force the coin to some value he wants with probability $1-o(1)$.

[5]The only clarification we have to make is what happens within one round when more than 1 (honest) player talks. We will again take the worst case and let the adversary order the the honest player in any way he wants (as usual, already bad players speak last). However, this is not a big limitation unless we care about the number of rounds: without loss of generality we can always let only one player talk within each round (and only get better resilience).

We note that *all* traditional coin-flipping protocols that we discussed earlier for static adversaries do not work in the adaptive setting. For example, we cannot let the players elect a leader who then flips a coin, as the adversary will wait and corrupt the elected leader. Not only that, but all the traditional coin-flipping protocols are *not even* 2*-resilient*, since at the late stages of the protocol at most two players control the coin flip. The exception is a reasonably weak majority protocol where players announce a bit, and a majority of the bits is the coin. Since this protocol is symmetric in all subsets of the players, and players' behavior is independent of anything else, it does not make a difference which coalition of b players our adversary $\mathcal{A}$ corrupts. Thus, like in the static case, majority is $\Theta(\sqrt{n})$-resilient. The big question is whether we can tolerate a larger than $\sqrt{n}$ threshold in the adaptive setting.

Interestingly enough, the problem was already considered briefly in the original paper of Ben-Or and Linial [BL90], who introduced the full-information model, coin-flipping and leader election. Ben-Or and Linial already observed that majority is $\Omega(\sqrt{n})$-resilient and raised a conjecture that this is optimal:

Conjecture 1 ([BL90]) *Majority is the optimal coin-flipping protocol against adaptive adversaries. In particular, the maximum threshold that can be tolerated is* $O(\sqrt{n})$.

The only paper addressing this conjecture is a very nice paper by Lichtenstein, Linial and Saks [LLS89]. By looking at another question that we will discuss later, they derived along the way the following result, that *seems* to strongly support the conjecture above.

Theorem 11 ([LLS89]) *If each player is allowed to broadcast at most* 1 *bit (possibly, taking* n *rounds overall), the most resilient coin-flipping protocol is indeed the majority protocol (which tolerates* $\Theta(\sqrt{n})$ *faults).*

The theorem above already shows some strong separation between static and adaptive adversaries. Recall that the result of Ajtai and Linial [AL93] says that there are $\Omega(n/\log^2 n)$-resilient functions. In other words, there are $\Omega(n/\log^2 n)$-resilient coin-flipping protocols where each player sends one bit (even in a single round!) which are secure against static adversaries. The above result says that no function (e.g., the function of Ajtai and Linial) $f : \{0,1\}^n \to \{0,1\}$, even if we spread it in any way over n rounds, can be more than $O(\sqrt{n})$-resilient against adaptive adversaries! Thus, *adaptive adversaries are strictly more powerful when each player is restricted to send only one bit.*

However, I believe that Theorem 11 supports Conjecture 1 much less than it seems to (and I will back it up!). I think that it is a very severe restriction against adaptive adversaries to let the players send only one bit. Intuitively, it seems like the only way to protect well against adaptive adversaries is to shuffle the order of the players' moves a lot; in particular, to have players send many bits in different orders with respect to each other. For example, when players send only one bit, the last players are typically much more influential than the first players. Thus, the adversary will typically let the first players talk honestly, and only when it comes to a decision-making towards the end, to start corrupting "important decision-makers" (note that the identity of these decision-makers will depend on the execution of the protocol, so static adversary might have trouble to "guess" in advance who these players are, but the adaptive adversary can wait). Hence, in order to make the last players not as much influential, we have to take restrict ourselves to very symmetric and "history-independent" protocols, like the majority. I believe the future of the adaptive coin-flipping protocols is to make a lot of rounds, where the order of the players changes dramatically, they send many bits and the answer depends on the whole history, so that the last few rounds are typically not going to make a difference.

In fact, I am now more inclined to think that Conjecture 1 is *false*, and raise a diametrically opposite conjecture that

Conjecture 2 *If there are* $b(n)$*-resilient coin-flipping protocols secure against static adversary, then there are (possibly much less efficient and having much worse resilience probability)* $b(n)$*-resilient coin-flipping protocols secure against adaptive adversaries.*

If true, the conjecture would imply that the resilience threshold for adaptive adversaries is the same as for the static ones: $n/2$. I will reduce the conjecture above to a very particular question on *extracting a somewhat random bit from an imperfect random source.*

10.1 Static to Adaptive Reduction

The reduction that might affirmatively resolve Conjecture 2 is trivial to describe. Let n be the number of players and N be some very large number, chosen later. Take any $b(n)$-resilient protocol Π secure against a static adversary. In particular, assume $a_\Pi(b(n)) \geq \gamma > 0$. Let $f : \{0,1\}^N \to \{0,1\}$ be a function whose properties we specify later. The protocol Ψ is simply this: run N invocations of Π (call the k-th invocation Π_k) to get bits $x_1, \ldots, x_N$. Make the final coin flip be $f(x_1, \ldots, x_N)$.

Of course, the crucial question is what properties we need from the function f, and why we believe such a function f exists. Given a random bit y, we say that y is γ-*nontrivial* if $\min(\Pr(y=0), \Pr(y=1)) \geq \gamma$. If the above minimum is at most γ, we say y is γ-*trivial*. We also call $\Pr(y=0)$ and $\Pr(y=1)$ the 0-*probability* and the 1-*probability* of y.

Now, we know by the assumption on Π that if the adversary $\mathcal{A}$ does not corrupt any players *during* the execution of the k-th coin flip Π_k, then x_k is γ-nontrivial. But if $\mathcal{A}$ corrupted at least one player during Π_k, all bets are off and we assume the worst case, i.e. that x_k has been maliciously set to 0 or 1 by $\mathcal{A}$. We point out that this is realistic, as most "static" coin-flipping protocols are not even 1-resilient (e.g., if they elect a leader who flips the coin). However, the second case can happen at most b times, since the adversary can corrupt at most b players, and we can set N as large as we want! Thus, our question reduces to the following. Assume we are generating one by one bits $x_1, \ldots, x_N$. The adversary $\mathcal{A}$ has the following capabilities in generating x_k after he sees the first $(k-1)$ bits $x_1, \ldots x_{k-1}$:

1. He can let 1-probability of x_k to be anywhere between γ and $1-\gamma$ (where $\gamma > 0$ is fixed).
2. He can deterministically set x_k to 0 or 1, provided he does it at most b times overall.

Thus, we can view our adversary $\mathcal{A}$ as an *imperfect random source* that emits N history dependent weakly random bits according to the rules 1. and 2. above.

Definition 8 *Call any $\mathcal{A}$ obeying rules 1. and 2. above a* (γ, b)-bounded imperfect random source. *Denote the minimum q such that $\mathcal{A}$ can make $y = f(x_1, \ldots, x_N)$ to be q-trivial by $q(\gamma, b, f, \mathcal{A})$, and let $q(\gamma, b, f) = \min_{\mathcal{A}} q(\gamma, b, f, \mathcal{A})$ (taken over all (γ, b)-bounded $\mathcal{A}$), and $q(\gamma, b) = \max_f q(\gamma, b, f)$.*

Note, that $\mathcal{A}$ is fully aware of the history $x_1, \ldots, x_{k-1}$, as well as the function f. Thus, the protocol Ψ is $b(n)$-resilient if $q(\gamma, b) \geq \gamma_0 > 0$, a constant independent of n and N.

To summarize, the question we ask is whether it is possible to extract at least one *somewhat random bit* (i.e. q-nontrivial bit for $q > 0$) from *any* (γ, b)-bounded imperfect random source $\mathcal{A}$. This is the same as to find a function f such that $f(\vec{x})$ is a somewhat random bit irrespective of ((γ, b)-bounded) $\mathcal{A}$. Let us make N our new parameter. By making it large enough, we can make b as low as $b(N) = \omega(1)$. Then we get

Lemma 9 *If there exists $b(N) = \omega(1)$ such that $q(\gamma, b(N)) \geq \gamma_0 > 0$ (for any constant $\gamma \in (0; \frac{1}{2}]$), i.e. we can extract at least one somewhat random bit from any $(\gamma, b(N))$-bounded imperfect random source, then Conjecture 2 holds. Notationally, the following implies Conjecture 2:*

$$q(\Omega(1), \omega(1)) = \Omega(1) \tag{5}$$

First, let us discuss two special cases of our imperfect random source, that have been considered in the literature, and for which Equation (5) is trivially true.

10.2 Bit-Fixing Source of Lichtenstein, Linial and Saks [LLS89]

Lichtenstein, Linial and Saks [LLS89] considered the case of $\gamma = \frac{1}{2}$ and arbitrary b. In other words, there is a sequence of N *truly* random bits emitted. The adversary can deterministically overwrite up to b of these bits. The question is whether we can extract at least one somewhat random bit from this source. Lichtenstein et al. looked at the problem from a slightly different (but almost the same) perspective. Assume we are given

a function $f : \{0,1\}^N \to \{0,1\}$. Identify it with a *language* $L = \{x \mid f(x) = 1\}$. Now the goal of our $(\frac{1}{2}, b)$-bounded (or simply b-bounded, as they call him) adversary is to make $x = x_1 \dots x_N$ belong to L (i.e. make $f(x) = 1$; note, this is slightly different from our adversary who either want $f(x) = 1$ or $f(x) = 0$, whichever he can succeed better). We let $v_L(b)$ be the probability $\mathcal{A}$ succeeds. For each $0 \le s \le 2^N$, Lichtenstein et al. found the "worst" possible language L of size s for the adversary (and the best for us), i.e. $\min_{|L|=s} v_L(b)$. It turns out that independently of b, this language is essentially the *threshold language*, i.e. the language of the form $\sum x_i \ge d$ for some value of d depending on s. In particular, the worst language with $|L| = 2^{N-1}$ (i.e. expectation of f equals to $\frac{1}{2}$) is the majority language. Also, if we want $s/2^N$ (expectation of f) to be constant, it implies that $d \approx N/2$ (up to additive $O(\sqrt{N})$), so that

- We can tolerate at most $b = O(\sqrt{N})$, i.e. $\Omega(\sqrt{N})$ interventions suffice to force $x \in L$ with probability $1 - o(1)$.
- Majority is the worst overall such language for $\mathcal{A}$.

Note, since the complement of the threshold language is also a threshold language, it also means that even if adversary can choose whether he wants $x \in L$ or $x \notin L$, majority is the worst language for $\mathcal{A}$, and also that at most $O(\sqrt{N})$ interventions can be tolerated (in order to get a somewhat random bit for our problem). To summarize,

Theorem 12 ([LLS89]) *$q(\frac{1}{2}, O(\sqrt{N})) = \frac{1}{2} - o(1)$, while $q(\frac{1}{2}, \Omega(\sqrt{N})) = o(1)$. In particular, majority is the best function f.*

Note, this result implies Theorem 11 we mentioned earlier. Indeed, in the coin-flipping protocols honest player send truly unbiased coin flips, while dishonest players send arbitrary bits. Thus, we have exactly the source in the above theorem, except adversary $\mathcal{A}$ cannot make *arbitrary interventions*, he can only intervene if the player is faulty. Assuming each player moves exactly t times, we have $N = tn$, and allow the adversary bt interventions. However, even though the adversary can corrupt players adaptively, once he corrupted the player, he cannot take it back. Thus, the above bt interventions are not arbitrary, so the above theorem cannot be applied (if it could, we would affirmatively resolve Conjecture 1 of Ben-Or and Linial). However, when $t = 1$, these interventions are indeed *arbitrary* and we get Theorem 11.

We also point that if the function f is a random function with constant expectation $e \in (\alpha; 1-\alpha)$ (e.g., $e = \frac{1}{2}$), Lichtenstein et al. observed that with high probability a *constant number of interventions* (to be precise, at most $2(1-\alpha)/\alpha$) suffice for $\mathcal{A}$ to fix f to either 0 or 1. This is simple to see, since if the adversary waits for the last b bits, with high probability f is not going to be fixed yet, so he can fix it to either 0 or 1 in the last b steps. However, it shows that since we have $b = \omega(1)$ in Conjecture 2, *random functions do not suffice even when $\gamma = \frac{1}{2}$*.

To summarize, when $\gamma = \frac{1}{2}$ Equation (5) is trivially true, even with $b = O(\sqrt{N})$, and majority achieves it. But a random function will not do the job for any $b = \omega(1)$.

10.3 Slightly-Random Source of Santha and Vazirani [SV86]

Santha and Vazirani [SV86] looked at the case $b = 0$, i.e. the adversary can set the 1-probability of any x_k based on $x_1, \dots, x_{k-1}$ to any value he wants. This source is sometimes referred as the *slightly-random source* or also *SV-source*.

On a negative side, Santha and Vazirani showed that one cannot extract q-nontrivial bits for any $q > \gamma$. Thus, the adversary $\mathcal{A}$ can always make sure that the resulting bit $f(x_1, \dots, x_N)$ is not better than any of the individual bits x_k. The way the adversary does it is roughly the following. Build a complete binary tree for f, and label each leaf by the corresponding value of f. Then the generation of $x_1, \dots, x_N$ can be viewed as a walk down the tree, and the output is the value of the leaf. At any internal node of the tree, the adversary decides on the probability the walk goes right. He can set it to any value in between γ and $1 - \gamma$. Assume without loss of generality that the number of 1-leaves is at least the number of 0-leafs. Then the strategy for the adversary is to look at which tree has the largest number of 1-leaves. He then chooses

the next bit so that it follows this subtree with (a maximal possible) probability $1-\gamma$. A simple inductive argument shows that the resulting bit is γ-trivial. Thus, $q(\gamma, 0) \leq \gamma$.

On the positive side, there are many f's that give γ-nontrivial bits, for example $f(x_1, \ldots, x_n) = x_k$ (for any k), i.e. we just output any one of the bits. Another such function is the parity function. Thus,

Theorem 13 ([SV86]) *It is possible to extract a γ-nontrivial bit from any γ-slightly-random source, and this is the best possible, i.e.*

$$q(\gamma, 0) = \gamma$$

In fact, Boppana and Narayanan [BN93], following the ideas of Alon and Rabin [AR89] and elegantly extending their techniques, showed much more. Namely, that

Theorem 14 ([AR89, BN93]) *For any $\gamma > 0$ there exists a constant $\gamma_0 > 0$ such that with probability exponentially close to 1, a* random *function f satisfies*

$$q(\gamma, 0, f) \geq \gamma_0$$

Thus, a vast majority of functions extract a *somewhat* random bit from any SV-source. Unfortunately, majority is not one of these functions. Indeed, if the adversary always sets the 1 probability of the next bit to be $1-\gamma$, the resulting bit will be 1 with probability $1 - o(1)$. In fact, Alon and Rabin [AR89] showed that *majority is the worst* bit-extracting function. Namely, $q(\gamma, 0, majority) \leq q(\gamma, 0, f)$, for any f.

To summarize, if the number of interventions $b = 0$, Equation (5) is trivially true. In fact, a random function will achieve it. However, majority does not.

10.4 Final Thoughts on the Problem

Looking at the last two sections with $\gamma = \frac{1}{2}$ and $b = 0$, we see that majority is great for the "bounded bit-fixing" case, but terrible for the "bias" case, while a random function is bad for the "bounded bit-fixing" case, but quite good for the "bias" case. The question remaining is whether there exists a function "in between" that would be resilient against both fixing $\omega(1)$ bits, and arbitrarily biasing the other bits within the range from γ to $(1-\gamma)$. Subsquent to the original draft of this survey, I resolved this question *in the negative* [Dod01].

Theorem 15 *If $b \cdot (\frac{1}{2} - \gamma) = \omega(1)$, then it is impossible to extract a slightly random bit from a (γ, b)-bounded imperfect source, irrespective of the value of N! More precisely,*

$$q(\gamma, b) \leq \frac{2}{(2-2\gamma)^b} = \frac{1}{2^{\Omega(b \cdot (\frac{1}{2} - \gamma)) - 1}} \tag{6}$$

In particular, while for $\gamma = \frac{1}{2}$ we could tolerate $b = O(\sqrt{N})$ (and even extract an *almost* perfect coin), and for $b = 0$ could deal with $\gamma = \Omega(1)$, now we cannot tolerate $b \to \infty$ for any $\gamma < \frac{1}{2}$, no matter how large N is. Also notice that the worst-case bias of any extracted coin *exponentially* approaches to $\frac{1}{2}$ as b grows.

As a simple application of this result, I also derived (see [Dod01] for details) that

Corollary 2 *Using optimal black-box reductions from static to adaptive coin-flipping one cannot tolerate more than $O(\sqrt{n})$ adaptively corrupted players. Since this bound is trivially achieved by majority, such black-box reductions cannot improve existing results in this setting. In particular, they cannot resolve Conjecture 1.*

To summarize, Conjecture 2 is false for black-box reductions if $b = \omega(\sqrt{n})$, while Conjecture 1 remains open.

References

[AL93] M. Ajtai, N. Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993.

[AN93] N. Alon, M. Naor. Coin-flipping games immune against linear-sized coalitions. *SIAM J. Comput.*, 22(2):403-417, 1993.

[AR89] N. Alon, M. Rabin. Biased Coins and Randomized Algorithms. *Advances in Computing Research*, 5:499-507, 1989.

[BL90] M. Ben-Or, N. Linial. Collective Coin-Flipping. In Slivio Micali, editor, *Randomness and Computation*, pp. 91-115, Academic Press, New York, 1990.

[BN93] R. Boppana, B. Narayanan. The Biased Coin Problem. *SIAM J. Discrete Math.*, 9(1)29–36, 1996.

[BN] R. Boppana, B. Narayanan. Perfect-information Leader Election with Optimal Resilience. *SIAM J. Comput., to appear.*

[CL95] J. Cooper, N. Linial. Fast perfect-information leader-election protocols with linear immunity. *Combinatorica*, 15:319–332, 1995.

[Dod01] Y. Dodis. New Imperfect Random Source with Applications to Coin-Flipping. *ICALP*, pp. 297–309, 2001.

[F99] U. Feige. Noncryptographic Selection Protocols. In *Proc. of* 40*th FOCS*, pp. 142–152, 1999.

[GGL98] O. Goldreich, S. Goldwasser, N. Linial. Fault-Tolerant Computation in the Full Information Model. *SIAM J. Comput.*, 27(2):506–544, 1998.

[KKL89] J. Kahn, G. Kalai, N. Linial. The Influence of Variables on Boolean Functions. In *Proc. of* 29*th FOCS*, pp. 68–80, 1989.

[LLS89] D. Lichtenstein, N. Linial, M. Saks. Some Extremal Problems Arising from Discrete Control Processes. *Combinatorica*, 9:269–287, 1989.

[ORV94] R. Ostrovsky, S. Rajagopalan, U. Vazirani. Simple and Efficient Leader Election in the Full Information Model. In *Proc. of* 26*th STOC*, pp. 234–242, 1994.

[RSZ99] A. Russell, M. Saks, D. Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. In *Proc. of* 31*st STOC*, pp. 339–347, 1999.

[RZ98] A. Russell, D. Zuckerman. Perfect information leader election in $\log^* n + O(1)$ rounds. In *Proc. of* 39*th FOCS*, pp. 576–583, 1998.

[S89] M. Saks. A robust noncryptographic protocol for collective coin flipping. *SIAM J. Discrete Math.*, 2(2):240–244, 1989.

[SV86] M. Santha, U. Vazirani. Generating Quasi-Random Sequences from Semi-Random Sources. *Journal of Computer and System Sciences*, 33(1):75–87, 1986.

[V84] L. Valiant. Short Monotone Formulae for the Majority Function. *J. Algorithms*, 5:363–366, 1984.