Full Analysis of Nakamoto Consensus in Bounded-Delay Networks

Juan Garay[1], Aggelos Kiayias[2], and Nikos Leonardos[3]

[1] Texas A&M University, garay@tamu.edu
[2] University of Edinburgh & IOHK, akiayias@inf.ed.ac.uk
[3] National and Kapodistrian University of Athens, nikos.leonardos@gmail.com

Abstract. Nakamoto consensus, arguably the most exciting development in distributed computing in the last few years, is in a sense a recasting of the traditional state-machine-replication problem in an unauthenticated setting, where furthermore parties come and go without warning. The protocol relies on a cryptographic primitive known as *proof of work* (PoW) which is used to throttle message passing with the PoW difficulty level being appropriately adjusted throughout the course of the protocol execution.

While the original formulation was only accompanied by rudimentary analysis, significant and steady progress has been made in abstracting the protocol's properties and providing a formal analysis under various restrictions, starting with the work by Garay, Kiayias and Leonardos [Eurocrypt '15], for a simplified version of the protocol which excluded PoW difficulty adjustment and assumed a fixed number of parties as well as synchronous communication rounds. These assumptions have since been relaxed, first by Pass, Seeman and Shelat [Eurocrypt '17] who also focused on the simplified version of the protocol but on the more realistic bounded-delay model of communication, and by Garay, Kiayias and Leonardos [Crypto '17] who looked into the full protocol including the PoW difficulty adjustment mechanism with a variable number of parties but again assuming synchronous communication and a predetermined schedule of participation. Despite the above progress, the full analysis of the protocol in the setting of bounded communication delays and dynamic participation has remained elusive.

This paper's main result is the proof that Nakamoto's protocol achieves, under suitable conditions, consistency and liveness in bounded-delay networks with adaptive (as opposed to predetermined) dynamic participation assuming, as in previous works, that the majority of the computational power favors the honest parties. While our techniques draw from previous analyses, our objective is significantly more challenging, demanding new insights and the introduction of new techniques in order to realize it.

1 Introduction

Nakamoto's blockchain protocol [18] is a protocol where parties engage in the collection and organization of transactions in a ledger without having any information about each other or even precise knowledge of the number of parties running the protocol at any given time. This is in contrast to classical models and results in consensus (aka Byzantine agreement) [20, 16] and other fundamental distributed computing tasks, where it is typically assumed that parties have pairwise authenticated communication channels or are initialized with the public keys of all participants. Instead, Nakamoto's blockchain protocol relies on the cryptographic primitive known as *proof-of-work* (PoW, aka cryptographic puzzles) [8, 22, 1, 13]), to throttle message transmission and stochastically create opportunities for unifying the parties' possibly diverging views, despite the presence of a subset of them acting adversarially.

Given that the original protocol was presented with only a rudimentary analysis focusing solely on the application context of fund transfers, a number of works have attempted to isolate the protocol's properties and provide a formal analysis. The first analysis, presented in [10], focused on a synchronous execution model, and assuming the probability of the parties to solve a PoW over a single message-passing round is suitably restricted, proved that the protocol satisfies consistency and liveness as long as the total computational power in the system is in favor of the honest parties. Two limitations of this first analysis were that the target recalculation mechanism of the blockchain protocol which adjusts the hardness of PoWs was excluded (and hence, necessarily, the total number parties was assumed to remain fixed throughout the execution), and that the execution model considered synchronous communication rounds.

Addressing the latter problem was undertaken in [19] (with further improvements in follow-up works [15, 21]), where the blockchain protocol was analyzed in the so-called *bounded-delay model* (cf. [7])[4], showing the protocol secure for a favorable choice of network delay Δ with respect to its hard-coded PoW hardness parameter, and its insecurity in the general case where Δ is chosen adversarially. Technically, the main challenge to address in transitioning to the bounded-delay model is the fact that the usefulness of a certain event in the protocol execution (e.g., the creation of a PoW at time t) is affected by events that are happening at times up to $t + \Delta$ forward in time (e.g., the creation of another PoW) and hence this dependence asks for additional care to be applied in the probabilistic analysis.

The problem of analyzing the target recalculation mechanism was addressed in [11], albeit again in the synchronous communication model, by introducing a setting where parties' participation is allowed to change round by round following a predetermined schedule that has a bounded rate of change. The main technical difficulty addressed in that setting was the fact that PoW successes are not independent events in the execution since the difficulty of the PoW primitive is determined by preceding execution events instead of being fixed throughout, as in [10, 19].

While the above works have significantly improved our understanding regarding the behavior of Nakamoto consensus in successively more refined theoretical models, the full analysis of the protocol has been elusive. Namely, the question is whether the protocol retains its properties in a bounded-delay network when the parties' number is dynamically changing without following a predetermined schedule, i.e., it is *adaptively* selected by the adversary, possibly even reacting to events that happen in the protocol execution, as long as the rate of change is bounded by a constant.

Our results. Our main result is the proof that Nakamoto's protocol achieves consistency and liveness in bounded-delay networks with adaptive dynamic participation. While our results draw from the previous analyses of [10, 19, 11], technically our objective is significantly more challenging and new techniques and insights are needed in order to realize it.

At the core of our analysis is the function $f(T, n)$, which determines the probability that n parties executing the protocol at a certain time find a PoW whose difficulty is determined by the "target" T (here, without loss of generality, n equates the number of parties with the number of CPUs of equal power running the protocol). Given that n is unknown and continuously changing, Nakamoto's protocol adjusts T at regular intervals called *epochs*. As we show, the protocol's resilience to attacks will stem from its ability to keep $f(T, n)$ close to a suitable value that is favorably positioned with respect to the, otherwise unknown, network delay Δ. Starting with the assumption that the protocol is initiated at an appropriate f value, the blockchain protocol will recalculate T to approximate that initial value by estimating the number of active parties n per epoch. The estimation is based on the observed production of PoWs as recorded in the blockchain itself and the relative timings of their production. A complication here is that for adversarially produced blocks, there is no guarantee that the timestamp they contain is correct. The only guarantee we have about timestamps is that in a valid blockchain they have to be monotonically increasing.

A major technical challenge is to work in a probabilistic setting where the random variables corresponding to the *cumulative difficulty* of PoWs (rather than their number) collected by the protocol participants capture the adaptive dynamic evolution of participants as well as the fact that some of these variables' values at a certain round may be affected by events in the future. The latter issue asks for a lower bound estimation of the aggregate difficulty (in terms of PoWs of different targets) collected by the honest parties over a period of time that is "isolated" from any future PoW event for a period of Δ rounds. At the same time, we need an upper bound on the aggregate difficulty amassed by adversarial parties while accounting for the fact that the adversary may choose to work on very difficult PoW instances for which it will be impossible to control their stochastic advantage via concentration bounds, due to high variance.

Our analysis culminates in two fundamental conditions, stated here at a high level (refer to Section 5.2 for the detailed specification), under which consistency and liveness of Nakamoto consensus can be shown: (C1) imposes a lower bound on the duration of an epoch, the period over which the estimation of observed

[4] In this model, formulated by Dwork, Lynch and Stockmeyer [7], there is an upper bound Δ, unknown to the protocol, in the delay that the adversary may inflict on the delivery of messages. See Section 2.

computational difficulty takes place; (C2) imposes a lower bound on the advantage that the honest computational power should have with respect to the adversary's power; finally, both conditions impose an upper bound on network delay Δ, as well as on the rate with which the number of parties dynamically change over time with respect to the "dampening" factor which is hard-coded in the Nakamoto protocol and tempers the PoW adjustment mechanism (cf. Def. 2).

The isolation of these two intuitive conditions is unique to our work, as no prior work was able to identify the sufficient conditions to prove Nakamoto consensus. Specifically, [10, 19, 15, 21] only identified variants of (C2) since they abstracted out the target recalculation mechanism, while [11] identified 7 interlocking requirements without accounting for network delay as in C2 above. Arguably, these conditions are also necessary for Nakamoto's consensus protocol. It is easy to see that the protocol will fail when there is not enough time to estimate the number of parties to adjust difficulty correctly (failure of C1); or when the honest parties' power is rivalled by the adversary (failure of C2).

Remark. The reader might wonder why the number of parties is allowed to adaptively dynamically evolve following even an exponential increase while the upper bound on the delay Δ is assumed fixed throughout the execution and not allowed to dynamically evolve as well. The reason is that Bitcoin was designed with exactly this setting in mind. (To see evidence of this consider that PoW production in the protocol is fixed to be at 10 min intervals, irrespectively of the computational hashing power available to the network which has increased significantly—and at periods of times even exponentially— since its initiation in 2009.) Designing a protocol that can absorb changes in Δ as well is an interesting, albeit rather theoretical, open question.

Organization of the paper. The remainder of the paper is organized as follows. In Section 2 we present the network, protocol execution and adversarial model; in particular, we define the *dynamic bounded-delay setting* where we our analysis is performed. In Section 3 we present the blockchain notation we will be using, Bitcoin's target recalculation function and desired properties the blockchain protocol should satisfy. Section 4 formally defines the Nakamoto consensus problem. Section 5 contains the bulk of our analysis, in particular the relevant protocol parameters and the conditions mentioned above under which Nakamoto consensus can be achieved, followed by characterizations of executions where the basic blockchain properties are satisfied, from which proofs of consistency and liveness can be derived. Mathematical facts and some of the proofs are presented in the appendix.

2 Model

We describe our protocols in the bounded-delay (aka "partially synchronous") model considered in [19] for the analysis of Nakamoto's blockchain protocol (albeit in the static setting with a fixed number of parties), in turn first formalized in [7], where there is an upper bound Δ in the delay (measured in number of rounds) that the adversary may inflict to the delivery of any message. The precise value of Δ will be unknown to the protocol (and in particular regular protocol participation will not rely on using Δ as a time-out parameter). Observe that "rounds" still exist in the model, but now these are not synchronization rounds where messages are supposed to be delivered to honest parties. We build on Canetti's formulation of "real world" notion of protocol execution [3–5] for multi-party protocols), adapting it to the dynamic setting with a varying number of parties and bounded delays.

Round structure and protocol execution. As in [9, 19], the protocol execution proceeds in "rounds" (note: these are not message-passing rounds) with inputs provided by an environment program denoted by $\mathcal{Z}$ to parties that execute the protocol Π. The adversary $\mathcal{A}$ is "adaptive," and allowed to take control of parties on the fly, as well as "rushing," meaning that in any given round the adversary gets to observe honest parties' actions before deciding how to react. Network and hash function access is bundled by a joint random oracle/diffusion ("gossiping") functionality. The diffusion functionality is similar to those in [9, 19]; it allows order of messages to be controlled by $\mathcal{A}$, i.e., there is no atomicity guarantees in message broadcast [12], and, furthermore, the adversary is allowed to spoof the source information on every message (i.e., communication is not authenticated). $\mathcal{A}$ can inject messages for selective delivery but cannot change the contents of the

honest parties' messages nor prevent them from being delivered beyond Δ rounds of delay — a functionality parameter.

The environment program $\mathcal{Z}$ determines the protocol execution; it creates and interacts with other instances of programs at the discretion of a control program C. Following [4], $(\mathcal{Z}, C)$ forms of a *system of interactive Turing machines* (ITM's). The only instances allowed by C are those of the protocol program Π, an adversary $\mathcal{A}$. These are called ITI's (interactive Turing Machines Instances). We refer to [4] for further details on the mechanics of the model. The only additional feature that is relevant to our setting is that we assume each instance is initialized with a special Boolean flag denoted as `ready` which is set to false upon initialisation.

The joint hash function/network functionality. We next present the dual functionality that is available to all parties running the protocol and the adversary and abstracts the hash function and the network. We choose to define the two as a joint functionality to ensure that access to the hash function will be provided in a "fair" manner to all participants according to our bookkeeping convention.

- *The hash function functionality.* It accepts queries of the form (compute, x) and (verify, x, y). For the first type of query, assuming x was never queried before, a value y is sampled from $\{0,1\}^\kappa$ and it is entered to a table T_H. If x was queried before the pair (x, y) is recovered from T_H. The value y is provided as an answer. For the second type of query, a membership test is performed on the table. Honest parties are allowed to ask *one query per round* of the type compute and unlimited queries of the type verify (note: we exclude Denial of Service attacks from our modeling, where $\mathcal{A}$ depletes the running time of parties by sending them too many messages for verification). The adversary $\mathcal{A}$ is given a bounded number of compute queries per round and no verify queries (the adversary can easily simulate those locally). The bound for the adversary is determined as follows. Whenever a corrupted party is activated the bound is increased by 1; whenever a query is asked the bound is decreased by 1 (it is not necessary that the specific corrupted party makes the query).
- *The diffusion functionality.* Message passing and round bookkeeping is maintained by this functionality. A round variable *round* is initialized to 0. For each party a string denoted by Receive() is maintained and the party is allowed to fetch the contents of its corresponding Receive() at any time. The functionality records all messages of the form $(\mathsf{Diffuse}, m)$ it receives from the parties. Completion of a round for a party is indicated by sending a special message ($\mathsf{RoundComplete}$). The adversary $\mathcal{A}$ is allowed to receive all the currently recorded $\mathsf{Diffuse}$ messages at any time and messages to the Receive() strings as desired. The round is completed when the adversary submits its ($\mathsf{RoundComplete}$) message. In such case, the functionality inspects the contents of all Receive() strings and includes any messages m that were diffused by the parties Δ rounds ago but not contributed by the adversary to the Receive() tapes (in this way guaranteeing message delivery up to Δ rounds). It also flushes any diffuse records that are placed in the Receive() string of all parties. The variable *round* is then incremented and a new round begins.

The dynamic bounded-delay setting. Given the functionalities as described above observe that contrary to prior formalizations, the adversary can choose the termination of the round thus deciding on the spot how many honest parties were activated adaptively. (In all previous works [10, 19, 11] the adversary was restricted to a preset number of activations.) In each round, the number of parties that are active in the protocol is denoted by n_r and is equal to the total number of parties that have submitted the ($\mathsf{RoundComplete}$) indicator to the diffusion functionality and have their internal flag `ready` set to true. Determining n_r can only be done by examining the view of all honest parties and is not a quantity that is accessible to any of the honest parties individually. The number of "corrupt" parties controlled by $\mathcal{A}$ in a round r is similarly denoted by t_r.

Parties, when activated, are able to read their input tape Input() and communication tape Receive() from the diffusion functionality. If a party finds that its `ready` flag is false, it enters a "bootstrapping" mode where it will diffuse a discovery message and synchronize (in the case of Nakamoto consensus, the party will send a request for the latest blockchains, will collect all of them until a time-out parameter is reached

and then will pick the most difficult one to start mining).[5] When the synchronization phase terminates, the party will set its `ready` flag to true and after this point it will be counted among the honest parties. An honest party goes "offline" when it misses a round, i.e., the adversary issues a (RoundComplete) but that party misses the opportunity to complete its computation. To record this action, whenever this happens we assume that the party's `ready` flag is set to false (in particular this means that a party is aware that it went offline; note, however, that the party does not need to report it to anyone). Also observe that parties are unaware of the set of activated parties. As in previous works (e.g., [10]), we assume, without loss of generality, that each honest party has the same computational power.[6]

We will restrict the environment to fluctuate the number of parties in a certain limited fashion. Suppose $\mathcal{Z}, \mathcal{A}$ with fixed coins produces a sequence of parties n_r where r ranges over all rounds of the execution. We define the following property, which is a finite sequence version of a similar property introduced in [11] for infinite sequences.

Definition 1. *For $\gamma \in \mathbb{R}^+$ we call $(n_r)_{r\in[0,B)}$, where $B \in \mathbb{N}$,* (γ, s)-respecting *if for any set $S \subseteq [0, B)$ of at most s consecutive integers,* $\max_{r\in S} n_r \leq \gamma \cdot \min_{r\in S} n_r$.

We say that $\mathcal{Z}$ is (γ, s)-respecting if for all $\mathcal{A}$ and coins for $\mathcal{Z}$ and $\mathcal{A}$ the sequence of parties n_r is (γ, s)-respecting.

The term $\{\textsc{view}^P_{\Pi,\mathcal{A},\mathcal{Z}}(z)\}_{z\in\{0,1\}^*}$ denotes the random variable ensemble describing the view of party P after the completion of an execution running protocol Π with environment $\mathcal{Z}$ and adversary $\mathcal{A}$, on input $z \in \{0,1\}^*$. We consider a "standalone" execution without any auxiliary information and we will thus restrict ourselves to executions with $z = 1^\kappa$. For this reason we will simply refer to the ensemble by $\textsc{view}^P_{\Pi,\mathcal{A},\mathcal{Z}}$. The concatenation of the view of all parties ever activated in the execution is denoted by $\textsc{view}_{\Pi,\mathcal{A},\mathcal{Z}}$.

Properties of protocols. In our theorems we will be concerned with *properties* of protocols Π running in the above setting. Such properties will be defined as predicates over the random variable $\textsc{view}_{\Pi,\mathcal{A},\mathcal{Z}}$ by quantifying over all possible adversaries $\mathcal{A}$ and environments $\mathcal{Z}$. Note that all our protocols will only satisfy properties with a small probability of error in κ as well as in a parameter k that is selected from $\{1, \ldots, \kappa\}$ (with foresight we note that in practice would be able to choose k to be much smaller than κ, e.g., $k = 6$).

3 Blockchains of Variable Difficulty

First we introduce the blockchain data structure and related notation, following [11]. Let $G(\cdot)$ and $H(\cdot)$ be cryptographic hash functions with output in $\{0,1\}^\kappa$. A block with target $T \in \mathbb{N}$ is a quadruple of the form $B = \langle r, st, x, ctr\rangle$ where $st \in \{0,1\}^\kappa, x \in \{0,1\}^*$, and $r, ctr \in \mathbb{N}$ are such that they satisfy the predicate $\mathsf{validblock}^T_q(B)$ defined as

$$(H(ctr, G(r, st, x)) < T) \wedge (ctr < q).$$

The parameter $q \in \mathbb{N}$ is an arbitrary bound that determines the size of the register ctr; in the case of Bitcoin it is 2^{32}. Contrary to [9, 11], q is not significant in any way in the analysis and can be set even to 1 (even though in practice this would not make sense, as it would require re-factoring the block contents after a single hash query).

A *blockchain*, or simply a *chain* is a sequence of *blocks* where the rightmost block is the *head* of the chain, denoted $\mathrm{head}(\mathcal{C})$. Note that the empty string ε is also a chain; by convention we set $\mathrm{head}(\varepsilon) = \varepsilon$. A chain $\mathcal{C}$ with $\mathrm{head}(\mathcal{C}) = \langle r, st, x, ctr\rangle$ can be extended to a longer chain by appending a valid block $B = \langle r', st', x', ctr'\rangle$ that satisfies $st' = H(ctr, G(r, st, x))$ and $r' > r$, where r' is called the *timestamp* of block B. In case $\mathcal{C} = \varepsilon$, by convention any valid block of the form $\langle r', st', x', ctr'\rangle$ may extend it. In either case we have an extended chain $\mathcal{C}_{\mathsf{new}} = \mathcal{C}B$ that satisfies $\mathrm{head}(\mathcal{C}_{\mathsf{new}}) = B$. We note that in Bitcoin and variant implementations the first block is called the "genesis" block, it is hardcoded in the client and serves the purpose of a common reference string. This is not significant for our analysis however and for simplicity can be ignored.

[5] Refer to Section 5.2 (specifically, discussion after Condition (C1)) for an adequate time-out value depending on the epoch's length and probability of at least one honest party out of the initial number of parties solving a PoW.

[6] A real-world mining pool or party of a certain hashing power can be thought of as a set of flat-model parties.

The *length* of a chain $\mathrm{len}(\mathcal{C})$ is its number of blocks. Consider a chain $\mathcal{C}$ of length ℓ and any nonnegative integer k. We denote by $\mathcal{C}^{\lceil k}$ the chain resulting from "pruning" the k rightmost blocks. Note that for $k \geq \mathrm{len}(\mathcal{C})$, $\mathcal{C}^{\lceil k} = \varepsilon$. If $\mathcal{C}_1$ is a prefix of $\mathcal{C}_2$ we write $\mathcal{C}_1 \preceq \mathcal{C}_2$.

Given a chain $\mathcal{C}$ of length $\mathrm{len}(\mathcal{C}) = \ell$, we let $\mathbf{x}_{\mathcal{C}}$ denote the vector of ℓ values that is stored in $\mathcal{C}$ and starts with the value of the first block. Similarly, $\mathbf{r}_{\mathcal{C}}$ is the vector that contains the timestamps of the blockchain $\mathcal{C}$.

The target T is recalculated for each block based on the round timestamps of the previous blocks. Specifically, there is a function $D : \mathbb{Z}^* \to \mathbb{R}$ which receives an arbitrary vector of round timestamps and produces the next target. The value $D(\varepsilon)$, denoted by T_0, is the initial target of the system that the genesis block should conform to. The *difficulty* of each block is measured in terms of how many times the block is harder to obtain than a block of target T_0. In more detail, the difficulty of a block with target T is equal to T_0/T; without loss of generality we will adopt the simpler expression $1/T$. We will use $\mathrm{diff}(\mathcal{C})$ to denote the aggregate difficulty of a chain. This is equal to the sum of all block difficulties that comprise the chain.

The target calculation function. At a high level, the target calculation function $D(\cdot)$ attempts to evaluate the number of parties that exist in the system (recall we operate in the flat model where honest parties have the same computational power) and adjust the target so the block production remains roughly the same over time. We use two parameters $m \in \mathbb{N}$ and $f_0 \in (0, 1)$ and the goal is to have an expectation that m blocks will be produced in m/f_0 rounds. We will see in Section 5 that the probability $f(T, n)$ with which n parties produce a new block with target T is roughly $Tn2^{-\kappa}$ with $f_0 = f(T_0, n_0)$ the initial "block production rate" where T_0 is the initial target and n_0 the initial estimate in the number of parties (in the sequel, for brevity, we will abuse notation and write f instead of f_0).

It follows that, if n is known, it is easy to adjust T so that $Tn2^{-\kappa}$ matches f. Just set T to be $T_0 n_0/n$. In order to facilitate the estimation of n, consider the last m blocks of a chain $\mathcal{C}$ where computed for target T and it took Λ rounds to produce them. We define the quantity

$$n(T, \Lambda) = 2^{\kappa} m/T\Lambda$$

Observe that $n(T_0, m/f) = 2^{\kappa} T_0 n_0 2^{-\kappa}/T_0 = n_0$. The above suggest a simple way to define the target recalculation formula: $T_0 n(T, \Lambda)/n_0$. This is flawed however as it is subject to attack, see [2, 11]. With surprising foresight, Bitcoin set its target recalculation together using a "dampening filter"-like adjustment. The formal definition is as follows.

Definition 2 (adapted from [11]). *For fixed constants $\kappa, \tau, m, n_0, T_0$, the target calculation function $D : \mathbb{Z}^* \to \mathbb{R}$ is defined as*

$$D(\varepsilon) = T_0 \quad \textit{and} \quad D(r_1, \ldots, r_v) = \begin{cases} \frac{1}{\tau} \cdot T & \textit{if } \frac{n_0}{n(T,\Lambda)} \cdot T_0 < \frac{1}{\tau} \cdot T; \\ \tau \cdot T & \textit{if } \frac{n_0}{n(T,\Lambda)} \cdot T_0 > \tau \cdot T; \\ \frac{n_0}{n(T,\Lambda)} \cdot T_0 & \textit{otherwise,} \end{cases}$$

where $n(T, \Lambda) = 2^{\kappa} m/T\Lambda$, with $\Lambda = r_{m'} - r_{m'-m}$, $T = D(r_1, \ldots, r_{m'-1})$, and $m' = m \cdot \lfloor v/m \rfloor$.

In the definition, $(r_1, \ldots, r_v)$ corresponds to a chain of v blocks with r_i the timestamp of the ith block; m', Λ, and T correspond to the last block, duration, and target of the last completed epoch, respectively.

Blockchain properties. The blockchain data structure's two fundamental properties, adapted from [9, 11], are related to the Nakamoto consensus properties of Consistency and Liveness (see below).

The *common prefix* property, parameterized by a value $k \in \mathbb{N}$, considers an arbitrary environment and adversary, and holds as long as any two parties' chains at two rounds have the earlier one subsumed in the former as long as k blocks are removed.

Definition 3 (Common Prefix). *The* common prefix *property Q_{cp} with parameter $k \in \mathbb{N}$ states that for any two players P_1, P_2 holding chains $\mathcal{C}_1, \mathcal{C}_2$ at rounds r_1, r_2, with $r_1 \leq r_2$, it holds that $\mathcal{C}_1^{\lceil k} \preceq \mathcal{C}_2$.*

The second property, called *chain quality*, quantifies the honest-party contributions that are contained in a sufficiently long and continuous part of a party's chain. Because we consider chains of variable difficulty, it is more convenient to think of parties' contributions in terms of the total difficulty they add to the chain as opposed to the number of blocks they add (as done in [10]). The property states that adversarial parties are bounded in the amount of difficulty they can contribute to any sufficiently long segment of the chain.

Definition 4 (Chain Quality). *The* chain quality *property Q_{cq}, with parameters $\mu \in \mathbb{R}$ and $\ell \in \mathbb{N}$, states that for any party P with chain $\mathcal{C}$ in $\textsc{view}_{\Pi,\mathcal{A},\mathcal{Z}}$, and any segment of that chain of difficulty d such that the timestamp of the first block of the segment is at least ℓ smaller than the timestamp of the last block, the blocks the honest parties have contributed in the segment have total difficulty at least $\mu \cdot d$.*

4 Nakamoto Consensus and its Properties

As mentioned in Section 1, *Nakamoto consensus* (aka "ledger consensus") is the problem where a set of servers (nodes, parties) operate continuosly accepting inputs ("transactions") and incorporate them in a public data structure called the *ledger*. More specifically, the problem is to maintain a ledger of transactions serialized in the form of a transaction sequence $\mathcal{L}$; satisfying the following two properties [10, 11]. Below we make the distinction between $\mathcal{L}$ and $\widetilde{\mathcal{L}}$, with the first denoting the settled ledger in the view of the party, and the second denoting the settled ledger with a sequence of transactions appended that are still not settled in the view of the party. In the context of Nakamoto's Bitcoin protocol, we note that $\widetilde{\mathcal{L}}$ will be the sequence of transactions defined by the chain $\mathcal{C}$ held by the party, while $\mathcal{L}$ will be the sequence of transactions defined by the prefix $\mathcal{C}^{\lceil k}$, where k is a security parameter.

- **Consistency** (cf. **Persistence** [10]): For any two honest parties P_1, P_2, reporting $\mathcal{L}_1, \mathcal{L}_2$ at rounds $r_1 \leq r_2$, respectively, it holds that $\mathcal{L}_1$ is a prefix of $\widetilde{\mathcal{L}}_2$.
- **Liveness** (parameterized by $u \in \mathbb{N}$, the "wait time" parameter): If a transaction tx is provided to all honest parties for u consecutive rounds, then it holds that for any player P, tx will be in $\mathcal{L}$.

We remark that the problem is a variant of the *state machine replication* problem [23].

5 Nakamoto Consensus in Bounded-Delay Networks and Dynamic Environments

In this section we present the full analysis and proofs of Nakamoto's consensus protocol in the originally envisioned dynamic environment where parties come and go, resulting in the adjusment of blocks' difficulty values. In particular, we formally specify (Section 5.2) the two conditions mentioned in Section 1 under which consistency and liveness of Nakamoto consensus can be shown.

5.1 Additional notation, definitions, and preliminary propositions

Our probability space is over all executions of length at most some polynomial in κ. Formally, the set of elementary outcomes can be defined as a set of strings that encode every variable of every party during each round of a polynomially bounded execution. (We will not delve into such formalism and leave further details unspecified.) We will denote by $\mathbf{Pr}$ the probability measure of this space. Define also the random variable $\mathcal{E}$ taking values on this space and with a distribution induced by the random coins of all entities (adversary, environment, parties) and the random oracle.

Suppose at round r exactly n parties query the oracle for target T. The probability at least one of them will succeed is

$$f(T, n) = 1 - (1 - pT)^n \leq pTn, \quad \text{where } p = 1/2^\kappa.$$

For the initial target T_0 and the initial estimate of the number of parties n_0, we denote $f_0 = f(T_0, n_0)$. Looking ahead, the objective of the target recalculation mechanism would be to maintain a target T for each

party such that $f(T, n_r) \approx f_0$ for all rounds r. For this reason, we will drop the subscript from f_0 and simply refer to it as f; to avoid confusion, whenever we refer to the function $f(\cdot, \cdot)$, we will specify its two operands.

We will next present some definitions which will allow us to introduce a few ("good") properties. These properties are an intermediate step towards proving common prefix and chain quality, but are also interesting in their own. The next two definitions are about the notions of "good chain" and "good round." The underlying notion of "goodness" is concerned with the targets that the honest parties are querying the random oracle for. At a round r of an execution the n_r honest parties might be querying the random oracle for various targets. We denote by $T_r^{\min}$ and $T_r^{\max}$ the minimum and maximum of those targets.

With respect to parameters that appear as "free" in the following definitions (such as γ, Δ, ℓ), please refer to the next subsection.

Definition 5. *Round r is* good *if $f/2\gamma^2 \leq pn_r T_r^{\min}$ and $pn_r T_r^{\max} \leq (1+\delta)\gamma^2 f$.*

Definition 6. *Round r is a* target-recalculation point *of a chain $\mathcal{C}$, if $\mathcal{C}$ has a block with timestamp r and height a multiple of m. A target-recalculation point r is* good *if the target T for the next block satisfies $f/2\gamma \leq pn_r T \leq (1+\delta)\gamma f$. A chain is* good *if all its target-recalculation points are* good.

We now define notions related to timestamps of the blocks, such as "accuracy," "epoch," and "duration."

Definition 7. *A block created at round u is* accurate *if it has a timestamp v such that $|u - v| \leq \ell + 2\Delta$. A chain is* accurate *if all its blocks are accurate. A chain is* stale *if for some $u \geq \ell + 2\Delta$ it does not contain an honest block with timestamp $v \geq u - \ell - 2\Delta$.*

Definition 8. *The blocks between two consecutive target recalculation points u and v on a chain $\mathcal{C}$ are an* epoch *of $\mathcal{C}$. The* duration *of the epoch is $u - v$.*

At a certain round of an execution, we would like to prove that the chain of every honest party has several desirable properties (along the notions just defined). This, however, entails a stronger statement in the following sense. At any given round there might exist chains which do not belong to any honest party (perhaps because the adversary kept them private), but have the potential to be adopted by one (i.e., have sufficient difficulty). With that in mind we define the following set of chains of a round r.

$$\mathcal{S}_r = \left\{ \mathcal{C} \in E_r \,\middle|\, \begin{array}{l} (\mathcal{C} \text{ belongs to an honest party}) \text{ or } (\exists \mathcal{C}' \in E_r \text{ that belongs to an honest party and either} \\ \text{diff}(\mathcal{C}) > \text{diff}(\mathcal{C}') \text{ or } \text{diff}(\mathcal{C}) = \text{diff}(\mathcal{C}') \text{ and } \text{head}(\mathcal{C}) \text{ was computed no later than } \text{head}(\mathcal{C}')) \end{array} \right\},$$

where $\mathcal{C} \in E_r$ means that $\mathcal{C}$ exists and is valid at round r.

Next, we define a series of useful predicates with respect to such set of chains.

Definition 9. *For a round r, let:*

- $\textsc{GoodChains}(r) \triangleq$ *"For all $u \leq r$, every chain in $\mathcal{S}_u$ is good."*
- $\textsc{GoodRounds}(r) \triangleq$ *"All rounds $u \leq r$ are good."*
- $\textsc{NoStaleChains}(r) \triangleq$ *"For all $u \leq r$, the are no stale chains in $\mathcal{S}_u$."*
- $\textsc{Accurate}(r) \triangleq$ *"For all $u \leq r$, all chains in $\mathcal{S}_u$ are accurate."*
- $\textsc{Duration}(r) \triangleq$ *"For all $u \leq r$ and duration Λ of any epoch in $\mathcal{C} \in \mathcal{S}_u$, $\frac{1}{2(1+\delta)\gamma^2} \cdot \frac{m}{f} \leq \Lambda \leq 2(1+\delta)\gamma^2 \cdot \frac{m}{f}$."*

Our goal is to show that, with high probability, an execution satisfies the common prefix and chain quality properties. To fulfill this goal we will first focus on showing that the execution satisfies the predicates we defined above. In particular, we will argue first that none of these predicates can fail, assuming proper initialization. Towards that goal, we first specify a number of random variables.

Random variables. In our analysis, we will be interested in estimating the difficulty acquired by honest parties during a sequence of rounds. Their number at a round r is denoted n_r and define the real random variable D_r equal to the sum of the difficulties of all blocks computed by honest parties at round r. Also, define Y_r to equal the maximum difficulty among all blocks computed by honest parties at round r, and

Q_r to equal Y_r when $D_u = 0$ for all $r < u < r + \Delta$ and 0 otherwise. We call a round r such that $D_r > 0$ *successful* and one wherein $Q_r > 0$ *isolated successful*. Regarding the adversary, let t_r denote the number of parties he controls at round r (equivalently, the number of random-oracle queries he can make at round r). Note that n_r and t_r are determined by the environment at the beginning of round r and should conform to the (γ, s)-respecting definition (Definition 1). We wish to upper bound the difficulty he can acquire during a set J of queries. Looking ahead, to obtain a good upper bound that holds with high probability, we will need some upper bound on the difficulty of a single block. However, the adversary may query the oracle for arbitrarily low targets and may obtain blocks of arbitrarily high difficulty. The following definition will allow us to work around these technical obstacles.

Consider a set of consecutive adversarial queries J, and note that the execution up to the first query in J determines the target associated with it. We denote this target by $T(J)$ and say that $T(J)$ is *associated* with J. We define $A(J)$ and $B(J)$ to be equal to the sum of the difficulties of all blocks computed by the adversary during queries in J for target at least $T(J)/\tau$ and $T(J)$, respectively. That is, queries in J for targets less than $T(J)/\tau$ (resp. $T(J)$) do not contribute to $A(J)$ (resp. $B(J)$). While considering consecutive epochs of a particular chain, the target can either increase by at most τ (and $B(J)$ will be appropriate), or decrease by at most τ (and $A(J)$ will be useful). For a set of rounds S or queries J we write $n(S) = \sum_{r \in S} n_r$ and similarly $t(S)$, $D(S)$, $Q(S)$, $A(J)$, $B(J)$.

Let $\mathcal{E}_{r-1}$ fix the execution just before round r. In particular, a value E_{r-1} of $\mathcal{E}_{r-1}$ determines the adversarial strategy and so determines the targets against which every party will query the oracle at round r and the number of parties n_r and t_r, but it does not determine D_r or Q_r. For an adversarial query j we will write $\mathcal{E}_{j-1}$ for the execution just before this query.

Proposition 1. *For any round r, $[1 - f(T_r^{\max}, n_r)]pn_r \leq \mathbf{E}[Y_r|\mathcal{E}_{r-1} = E_{r-1}] \leq \mathbf{E}[D_r|\mathcal{E}_{r-1} = E_{r-1}] = pn_r$, $\mathbf{E}[Y_r^2|\mathcal{E}_{r-1} = E_{r-1}] \leq pn_r/T_r^{\min}$, and $\mathbf{var}[D_r|\mathcal{E}_{r-1} = E_{r-1}] \leq pn_r/T_r^{\min}$.*

Proof. Let us drop the subscript r for convenience. Suppose that the n honest parties at round r query for targets $T_1, \ldots, T_n$. Observe that all these variables are determined by E_{r-1}. We have

$$\begin{aligned}\mathbf{E}[Y_r|\mathcal{E}_{r-1} = E_{r-1}] &= \sum_{i \in [n]} \frac{1}{T_i} \cdot \frac{T_i}{2^\kappa} \prod_{i<j} \big[1 - f(T_j, 1)\big] \geq \sum_{i \in [n]} p \prod_{j \in [n]} \big[1 - f(T_j, 1)\big] \\ &\geq \sum_{i \in [n]} p \prod_{j \in [n]} \big[1 - f(T^{\max}, 1)\big] = \sum_{i \in [n]} p[1 - f(T^{\max}, n)] = pn[1 - f(T^{\max}, n)],\end{aligned}$$

where the third inequality holds because $f(T, n)$ is increasing in T. For the upper bound on variance,

$$\mathbf{var}[D_r|\mathcal{E}_{r-1} = E_{r-1}] \leq \sum_{i \in [n]} \frac{1}{T_i^2} \cdot \frac{T_i}{2^\kappa} = \sum_{i \in [n]} \frac{p}{T_i} \leq \frac{pn}{T^{\min}}$$

and $\mathbf{E}[Y_r^2|\mathcal{E}_{r-1} = E_{r-1}]$ is bounded alike.

The following proposition collects a few useful inequalities that hold in a (γ, s)-respecting environment.

Proposition 2. *Let U be a set of at most s consecutive rounds in a (γ, s)-respecting environment and $S \subseteq U$.*

- *For any $n \in \{n_r : r \in U\}$, $\frac{n}{\gamma} \leq \frac{n(S)}{|S|} \leq \gamma n$.*
- $n(U) \leq \big(1 + \frac{\gamma|U \setminus S|}{|S|}\big) n(S)$.
- $|S| \sum_{r \in S} (pn_r)^2 \leq \gamma (\sum_{r \in S} pn_r)^2$.

Proof. The first part is proved in [11] and is a direct consequence of the definition. For the second, note that $n(U) = n(S) + n(U \setminus S)$ and $n(U \setminus S) \leq |U \setminus S|\gamma n(S)/|S|$. For the third, note that $pn/\gamma \leq pn_r \leq \gamma pn$ for any $r \in S$. The inequality follows from Theorem 8, since $A/G \leq (\gamma + 1/\gamma)/2 \leq \gamma$.

5.2 Parameters and Their Conditions

Our formalization of the protocol involves a number of parameters that have already appeared in the text above. With respect to the environment,

- δ: Advantage of honest parties[7] ($t_r < (1-\delta)n_r$ for all r);
- (γ, s) : It restricts the fluctuation of the number of parties across rounds (Definition 1); we set $s = \tau m/f$.

An important parameter, which is a function of the protocol's initialization parameters n_0 and T_0, is

- $f \in (0,1)$: The probability at least one honest party out of n_0 computes a block for target T_0; i.e., $f(T_0, n_0)$.

The protocol strives to maintain the probability of a successful round as close f as possible. This is the task of the target recalculation function. Recall Definition 2 and the significance of the involved parameters

- $m \in \mathbb{N}$: The length of an epoch in number of blocks;
- $\tau \geq 1$: The dampening filter.

The value $\tau m/f$ is the longest an epoch might take to complete and $\gamma \geq 1$ is an estimate of how much the number of parties can change in such a time interval.

Our two main security parameters are κ and λ. The first determines the probability of a collision and the second the probability desirable properties fail to hold.

- κ : The length of the range of the hash function;
- λ : Related to the properties of the protocol.

To achieve security, we will argue concentration of several random variables. Furthermore, in any exponentially long (in the security parameters) execution bad events are bound to happen.

- ϵ: Quality of concentration of random variables;
- L : The total number of rounds in the execution.

For $L = \mathsf{poly}(\kappa, \lambda)$, our statements will fail with probability $\mathsf{poly}(\kappa, \lambda)(e^{-\kappa} + e^{-\lambda})$.

To obtain meaningful concentration, we should be considering a sufficiently long sequence of at least

$$\ell = \frac{4(1+3\epsilon)}{\epsilon^2 f[1-(1+\delta)\gamma^2 f]^{\Delta+1}} \cdot \max\{\Delta, \tau\} \cdot \gamma^3 \cdot \lambda. \tag{1}$$

consecutive rounds.

For our analysis to go through, the above parameters should satisfy certain **conditions** which we now discuss. First, we will require that the number ℓ defined above is appropriately small compared to the duration of an epoch.

$$2\ell + 6\Delta \leq \frac{\epsilon m}{2(1+\delta)\gamma^2 f}. \tag{C1}$$

The above condition also suggests a way to set the time-out parameter for bootstrapping, since (along with the expression for ℓ above) it provides an upper bound on Δ. Second, the advantage of the honest parties δ should be large enough to absorb error factors such as $(1-\epsilon)$ and $[1-(1+\delta)\gamma^2 f]^{\Delta}$. To that end, we require the following inequalities.

$$[1-(1+\delta)\gamma^2 f]^{\Delta} \geq 1-\epsilon \quad \text{and} \quad \epsilon \leq \delta/8 \leq 1/8. \tag{C2}$$

5.3 Chain Growth Lemma

We now prove the Chain Growth Lemma. This lemma appears already in [10] in a model with fixed difficulty and fixed number of parties. In [14] the name "chain growth" appears for the first time, and where the authors explicitly state a Chain Growth Property. In [11], the lemma is proved in a synchronous model allowing variable difficulty and varying number of parties. Here we give a different proof that works in the dynamic bounded-delay model. The lemma provides a lower bound on the progress of the honest parties, which holds irrespective of any adversary.

[7] Note that we denote the number of honest parties at round r by n_r and the number of parties controlled by the adversary by t_r, so that the total number of parties is $n_r + t_r$. Although this is not standard, it simplifies several expressions and is also in agreement with notation in [11].

Lemma 1 (Chain Growth). *Suppose that at round u of an execution E an honest party broadcasts a chain of difficulty d. Then, by round v, every honest party has received a chain of difficulty at least $d+Q(S)$, where $S = \{r : u + \Delta \leq r \leq v - \Delta\}$.*

Proof. If two blocks are obtained at rounds which are at distance at least Δ, then we are certain that the later block increased the accumulated difficulty. To be precise, assume $S^* \subseteq S$ is such that, for all $i, j \in S^*$, $|i - j| \geq \Delta$ and $Y_i > 0$. We argue that, by round v, every honest party has a chain of difficulty at least

$$d + \sum_{r \in S^*} Y_r \geq d + \sum_{r \in S} Q_r.$$

Observe first that every honest party will receive the chain of difficulty d by round $u + \Delta$ and so the first block obtained in S^* extends a chain of weight at least d. Next, note that if a block obtained in S^* is the head of a chain of weight at least d', then the next block in S^* extends a chain of weight at least d'.

5.4 Typical Executions: Definition and Related Proofs

We now define formally our notion of *typical* executions. Intuitively, the idea that this definition captures is as follows. Suppose that we examine a certain execution E. Note that at each round of E the parties perform Bernoulli trials with success probabilities possibly affected by the adversary. Given the execution, these trials are determined and we may calculate the expected progress the parties make given the corresponding probabilities. We then compare this value to the actual progress and if the difference is reasonable we declare E *typical*. Note, however, that considering this difference by itself will not always suffice, because the variance of the process might be too high. Our definition, in view of Theorem 7, says that either the variance is high with respect to the set of rounds we are considering, or the parties have made progress during these rounds as expected.

Beyond the behavior of random variables described above, a typical execution will also be characterized by the absence of a number of bad events about the underlying hash function $H(\cdot)$ which is used in proofs of work and is modeled as a random oracle. The bad events that are of concern to us are defined as follows (recall that a block's creation time is the round that it has been successfully produced by a query to the random oracle either by the adversary or an honest party).

Definition 10. *An* insertion *occurs when, given a chain $\mathcal{C}$ with two consecutive blocks B and B', a block B^* created after B' is such that B, B^*, B' form three consecutive blocks of a valid chain. A* copy *occurs if the same block exists in two different positions. A* prediction *occurs when a block extends one with later creation time.*

Given the above we are now ready to specify what is a typical execution.

Definition 11 (Typical execution). *An execution E is* typical *if the following hold.*

(a) For any set S of at least ℓ consecutive good rounds,

$$(1-\epsilon)[1-(1+\delta)\gamma^2 f]^{\Delta} pn(S) < Q(S) \leq D(S) < (1+\epsilon)pn(S).$$

(b) For any set J of consecutive adversarial queries and $\alpha(J) = 2(\frac{1}{\epsilon} + \frac{1}{3})\lambda/T(J)$,

$$A(J) < p|J| + \max\{\epsilon p|J|, \tau\alpha(J)\} \quad \textit{and} \quad B(J) < p|J| + \max\{\epsilon p|J|, \alpha(J)\}.$$

(c) No insertions, no copies, and no predictions occurred in E.

We will be interested in comparing the computational power of the adversary against that of the honest parties in a set of consecutive rounds S. However, in the bounded-delay model with delay Δ, the adversary can mute the honest parties for the final Δ rounds. The calculation summarized in the following lemma will be used repeatedly.

Lemma 2. *Consider a typical execution in a (γ, s)-respecting environment. Let $S = \{r : u \leq r \leq v\}$ be a set of at least ℓ consecutive good rounds and J the set of adversarial queries in $U = \{r : u - \Delta \leq r \leq v + \Delta\}$.*

(a) $(1+\epsilon)p|J| \leq Q(S) \leq D(U) < (1+5\epsilon)Q(S)$.
(b) $T(J)A(J) < \epsilon m/4(1+\delta)$ *or* $A(J) < (1+\epsilon)p|J|$ *and* $\tau T(J)B(J) < \epsilon m/4(1+\delta)$ *or* $B(J) < (1+\epsilon)p|J|$.
(c) If w is a good round such that $|w - r| \leq s$ for any $r \in S$, then $Q(S) > (1-\epsilon)[1-(1+\delta)\gamma^2 f]^{\Delta}|S|pn_w/\gamma$. If in addition $T(J) \geq T_w^{\min}$, then $A(J) < (1-\delta+3\epsilon)Q(S)$.

The main result of this section is that almost all polynomially bounded in κ and λ executions are typical.

Theorem 1. *Assuming the ITM system $(\mathcal{Z}, C)$ runs for L steps, the probability of the event "$\mathcal{E}$ is not typical" is bounded by $O(L^2)(e^{-\lambda} + 2^{-\kappa})$.*

5.5 Technical Overview

Next, we present an overview of the flow of the analysis, establishing the validity of the predicates of Definition 9 over the space of typical executions in a (γ, s)-respecting environment. This constitutes the bulk of the analysis, which in turn leads to the establishment of the common prefix and chain quality properties, which in turn are used to show that Nakamoto consensus' properties hold. The detailed presentation can be found in the subsequent subsections.

First, we show that typical executions are good and accurate. Our first lemma says that the adversary cannot maintain a chain by himself for too long. The reason is that the honest parties will progress faster and his blocks will be orphaned. This implies accurate timestamps.

Lemma 3. $\textsc{GoodRounds}(r-1) \implies \textsc{NoStaleChains}(r)$.

Corollary 1. $\textsc{GoodRounds}(r-1) \implies \textsc{Accurate}(r)$.

Lemma 4. $\textsc{GoodRounds}(r-1) \wedge \textsc{GoodChains}(r-1) \implies \textsc{Duration}(r)$.

Lemma 5. $\textsc{GoodRounds}(r-1) \implies \textsc{GoodChains}(r)$.

Corollary 2. $\textsc{GoodRounds}(r-1) \implies \textsc{GoodRounds}(r)$.

Which allows us to conclude that, under certain conditions, all the predicates hold:

Theorem 2. *Consider a typical execution in a $(\gamma, 2(1+\delta)\gamma^2 m/f)$-respecting environment. If the Conditions C1 and C2 are satisfied, then all predicates of Definition 9 hold.*

Next, we establish that the common prefix and chain quality properties hold.

Common Prefix and Chain Quality. The following lemma says that two "longest" chains cannot diverge for too long.

Lemma 6. *For any round r of a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment and any two chains $\mathcal{C}$ and $\mathcal{C}'$ in $\mathcal{S}_r$, the timestamp of* $\mathrm{head}(\mathcal{C} \cap \mathcal{C}')$ *is at least $r - 2\ell - 4\Delta$.*

Theorem 3 (Common Prefix). *For a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment, the common-prefix property holds for parameter ϵm.*

Theorem 4 (Chain Quality). *For a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment, the chain-quality property holds with parameters $\ell + 2\Delta$ and $\mu = \delta - 3\epsilon$.*

Nakamoto Consensus: Consistency and Liveness. Consistency and Liveness can now be shown easily as in previous work, such as [11, 19, 10].

Theorem 5 (Consistency). *For a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment, Consistency is satisfied by setting the settled transactions to be those reported more than ϵm blocks deep.*

The proof follows directly from the Common Prefix property (Theorem 3).

Theorem 6 (Liveness). *For a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment, Liveness is satisfied for depth ϵm with wait-time $(4\gamma^2+1)\epsilon m/f$.*

We refer to the remainder of the section for the full analysis.

5.6 Typical Executions are Good and Accurate

In this subsection we study in detail the validity of the predicates of Definition 9 over the space of typical executions in a (γ, s)-respecting environment. All statements in this subsection assume a typical execution and a (γ, s)-respecting environment, for $s \geq 2(1+\delta)\gamma^2 m/f$. Furthermore, the Conditions are assumed to hold for the initialization parameters n_0 and T_0.

Our first lemma says that the adversary cannot maintain a chain by himself for too long. The reason is that the honest parties will progress faster and his blocks will be orphaned. This implies accurate timestamps.

Lemma 3. $\textsc{GoodRounds}(r-1) \implies \textsc{NoStaleChains}(r)$.

Proof. Suppose—towards a contradiction—$\mathcal{C} \in \mathcal{S}_r$ and has not been extended by an honest party for at least $\ell + 2\Delta$ rounds and r is the least round with this property. Let B be the last block of $\mathcal{C}$ computed by honest parties (possibly the genesis) and let w be its timestamp. Set $S = \{u : w + \Delta \leq u \leq r - \Delta\}$ and $U = \{u : w \leq u \leq r\}$. Note that by our assumption $|S| \geq \ell$. Suppose that the blocks of $\mathcal{C}$ after B span k epochs with corresponding targets $T_1, \ldots, T_k$. For $i \in [k]$ let m_i be the number of blocks with target T_i and set $M = m_1 + \cdots + m_k$ and $d = m_1/T_1 + \cdots + m_k/T_k$. Our plan is to contradict the assumption that $\mathcal{C} \in \mathcal{S}_r$ by showing that all chains in $\mathcal{S}_r$ have more difficulty than $\mathcal{C}$. By Chain-Growth (Lemma 1), all the honest parties have advanced (in difficulty) during the rounds in U by $Q(S)$. Therefore, to reach a contradiction it suffices to show that $d < Q(S)$.

When $k > 2$ we may partition these M blocks into $k-1$ parts so that each part has the following properties: (1) it contains at most one target-recalculation point, and (2) it contains at least $m/2$ blocks. For each $i \in \{1, 2, \ldots, k-1\}$, let $j_i \in J$ be the index of the query during which the first block of the i-th part was computed and set $J_i = \{j : j_i \leq j < j_{i+1}\}$ (Definition 11(c) assures $j_i < j_{i+1}$). We claim

$$d = \sum_{i=1}^{k} \frac{m_i}{T_i} < \sum_{i=1}^{k-1} (1+\epsilon)|J_i| \leq (1+\epsilon)p|J| \leq Q(S).$$

For the first inequality, consider part i. We have $T_i = T(J_i)$ and—because of the first property of the partition—two possible cases for T_{i+1}: either $T_i \leq T_{i+1} \leq \tau T_i$ or $T_i/\tau \leq T_{i+1} \leq T_i$. In the first case, the difficulty of the blocks acquired in J_i is at most $B(J_i)$ and their number at most $\tau T_i B(J_i)$. In the second case, the difficulty of the blocks acquired in J_i is at most $A(J_i)$ and their number at most $T_i A(J_i)$. In either case, since the adversary acquired at least $m/2$ blocks in J_i, the desired bound follows from Lemma 2(b). The final inequality is Lemma 2(a).

If $k \leq 2$, let J denote the queries in U starting from the first adversarial query attempting to extend B. Then, $T_1 = T(J)$ and $T_2 \geq T(J)/\tau$; thus, $d \leq A(J)$. If $A(J) < (1+\epsilon)p|J|$, then $A(J) < Q(S)$ is obtained by Lemma 2(a). Otherwise, $A(J) < (\frac{1}{\epsilon}+1)\tau\alpha(J) = 2(\frac{1}{\epsilon}+1)(\frac{1}{\epsilon}+\frac{\epsilon}{3})\tau\lambda/T(J)$. However, by considering only the first ℓ rounds in S, $n(S) \geq n_u \ell/\gamma$. We have

$$Q(S) > (1-\epsilon)[1-(1+\delta)\gamma^2 f]^{\Delta} \cdot \frac{p n_u \ell T_1}{\gamma T_1} > \frac{(1-\epsilon)[1-(1+\delta)\gamma^2 f]^{\Delta} f\ell}{2\gamma^3 T(J)} \geq \frac{2(1-\epsilon)(1+3\epsilon)\tau\lambda}{\epsilon^2 T(J)} \geq A(J).$$

Corollary 1. $\textsc{GoodRounds}(r-1) \implies \textsc{Accurate}(r)$.

Proof. Suppose—towards a contradiction—that, for some $w \leq r$, $\mathcal{C} \in \mathcal{S}_w$ contains a block which is not accurate and let $u \leq w$ be the timestamp of this block and v its creation time. If $u - v > \ell + 2\Delta$, then every honest party would consider $\mathcal{C}$ to be invalid during rounds $v, v+1, \ldots, u$. If $v - u > \ell + 2\Delta$, then in order for $\mathcal{C}$ to be valid it should not contain any honest block with timestamp in $u, u+1, \ldots, v$. (Note that we are using Definition 11(c) here as a block could be inserted later.) In either case $\textsc{NoStaleChains}(r)$ is false contradicting the previous lemma.

Lemma 4. $\textsc{GoodRounds}(r-1) \wedge \textsc{GoodChains}(r-1) \implies \textsc{Duration}(r)$.

Proof. Assume—towards a contradiction—that $\textsc{Duration}(r)$ is false. Then, there exists a $w \le r$ and a chain $\mathcal{C} \in \mathcal{S}_w$ with an epoch of target T and duration Λ that does not satisfy

$$\frac{1}{2(1+\delta)\gamma^2} \cdot \frac{m}{f} \le \Lambda \le 2(1+\delta)\gamma^2 \cdot \frac{m}{f}.$$

We consider the earliest epoch with this property.

For the upper bound, Lemma 3 implies the existence of two honest blocks in this epoch with timestamps that differ at least $\Lambda - 2\ell + 4\Delta$. Let u and v be these timestamps and define $S = \{i : u + \Delta \le i \le v - \Delta\}$. Assuming $\Lambda > 2(1+\delta)\gamma^2 m/f$, Condition C1 implies $|S| \ge \Lambda - 2\ell - 6\Delta \ge 2(1+\delta)(1-\epsilon)\gamma^2 m/f$. We have

$$Q(S) > (1-\epsilon)[1-(1+\delta)\gamma^2 f]^{\Delta} \cdot \frac{f|S|}{2\gamma^2 T} \ge (1+\delta)(1-\epsilon)^3 \cdot \frac{m}{T} > \frac{m}{T}.$$

For the first inequality we used that the beginning of the epoch was a good target-recalculation point; the lower bound for $|S|$ for the next and Condition C2 for the last inequality. This contradicts Chain Growth, since the honest parties at round v already have more than m/T difficulty on top of u.

To prove the lower bound we are going to argue that even if the honest parties and the adversary join forces they still cannot obtain m blocks. Let u be the timestamp of the first block in the epoch and $v = u + \Lambda$. Set $S = \{u, \ldots, v\}$, $S' = \{u - \ell - 2\Delta, \ldots, v + \ell + 2\Delta\}$,[8] and J the set of queries available to the adversary during the rounds in S' starting with the first query for target T (so that $T(J) = T$). Since $\textsc{Accurate}(r)$ holds (Corollary 1), the adversarial queries that contributed to the epoch are all in J. Since u is a good target-recalculation point[9] and $n_r \le \gamma n_u$ for all $r \in S'$, it follows that $pn(S) \le p\gamma n_u |S| \le (1+\delta)\gamma^2 f|S|/T$. Thus,

$$D(S) < (1+\epsilon)pn(S) \le (1+\epsilon)(1+\delta) \cdot \frac{\gamma^2 f|S|}{T} \le (1+\epsilon) \cdot \frac{m}{2T}.$$

With respect to the adversary, if $\tau TB(J) < \epsilon m/4$, then the total number of blocks is less than m and we are done. Otherwise, by Lemma 2(b),

$$\begin{aligned} B(J) &< (1+\epsilon)p|J| \le (1+\epsilon)(1-\delta)pn(S') \le (1+\epsilon)(1-\delta)p\gamma n_u|S'| \le (1+\epsilon)(1-\delta) \cdot \frac{p\gamma n_u T}{T} \cdot |S'| \\ &\le (1+\epsilon)(1-\delta) \cdot \frac{(1+\delta)\gamma^2 f}{T} \cdot |S'| \le (1-\delta^2)(1+\epsilon)\Big(1 + \frac{2\ell + 4\Delta}{|S|}\Big)|S| \cdot \frac{\gamma^2 f}{T} \le (1-\delta)(1+\epsilon)^2 \cdot \frac{m}{2T}. \end{aligned}$$

For the last inequality we used Condition C1 and $|S| \le m/2(1+\delta)\gamma^2 f$. Since $\epsilon \le \delta/4$ (Condition C2), the total count of blocks is again less than m.

Lemma 5. $\textsc{GoodRounds}(r-1) \implies \textsc{GoodChains}(r)$.

Proof. Note that it is our assumption that the first round (the genesis) is a good target-recalculation point. Therefore, it suffices to show that if a recalculation point u in a chain $\mathcal{C} \in \mathcal{S}_r$ is good, then the next one at $v = u + \Lambda \le r$ is also good. Let T be the target of the epoch starting at u and T' the target of the next one. We wish to show that $f/2\gamma \le pn_v T' \le (1+\delta)\gamma f$.

We prove first the lower bound. If $\Lambda \ge \gamma m/f$, then $T' \ge \gamma T$ (using $\gamma \le \tau$) and so $pn_v T' \ge pn_u T'/\gamma \ge pn_u T \ge f/2\gamma$, because u is assumed to be a good target-recalculation point. We assume now $\Lambda < \gamma m/f$, which implies $\Lambda \le (T'/T)(m/f)$. Define $S = \{u, \ldots, v\}$, $S' = \big\{u - \ell - 2\Delta, \ldots, v + \ell + 2\Delta\big\}$,[8] and J the set of queries available to the adversary in S'. Clearly, all blocks were computed during honest queries in S or adversarial ones in J. Note that, by Condition C1 and Lemma 4, $|S'| = \Lambda + 2\ell + 4\Delta \le (1+\epsilon)\Lambda$. We have

$$B(J) < (1-\delta)(1+\epsilon)pn(S') \le (1-\delta)(1+\epsilon)p\gamma n_v|S'| \le (1-\delta)(1+\epsilon)^2 p\gamma n_v \Lambda.$$

[8] To be more precise we should adjust the endpoints of S' to make sure $S' \subseteq \{0, ..., r\}$.

[9] The alert reader might spot the annoying case $u = v = r$ (which basically amounts to the adversary completing a whole epoch by himself). Since we are considering the earliest epoch in violation, the epoch ending at u didn't last longer than s rounds. The proof of Lemma 5 implies $pn_u T \le 2\gamma f$ holds.

Similarly, $D(S) < (1+\epsilon)pn(S) \leq (1+\epsilon)p\gamma n_v \Lambda$. Assuming $pn_v T' < f/2\gamma$ we obtain the contradiction

$$2\gamma p n_v \Lambda \leq 2\gamma p n_v \cdot \frac{T'}{T} \cdot \frac{m}{f} < \frac{m}{T} \leq D(S) + B(J) < (2 + 4\epsilon - \delta)p\gamma n_v \Lambda \leq 2\gamma p n_v \Lambda.$$

For the upper bound, let $S = \{u + \ell + 3\Delta, \ldots, v - \ell + 3\Delta\}$. Note first that if $\Lambda \leq m/\gamma f$, then $T' \leq T/\gamma$ and so $pn_v T' \leq p\gamma n_u T' \leq pn_u T \leq (1+\delta)\gamma f$, where we used that u is a good target-recalculation point. Thus, we may assume $\Lambda > m/\gamma f$, which implies $\Lambda \geq (T'/T)(m/f)$ and $|S| = |\Lambda - 2\ell - 6\Delta| \geq (1-\epsilon)\Lambda$. Assuming $pn_v T' > (1+\delta)\gamma f$, we obtain the following contradiction.

$$\frac{pn_v \Lambda}{(1+\delta)\gamma} \geq \frac{pn_v}{(1+\delta)\gamma} \cdot \frac{T'}{T} \cdot \frac{m}{f} > \frac{m}{T} \geq Q(S) > (1-\epsilon)[1 - (1+\delta)\gamma^2 f]^{\Delta} \cdot \frac{pn_v |S|}{\gamma} \geq \frac{pn_v \Lambda}{(1+\delta)\gamma}.$$

The first two inequalities have been discussed above. For the third one, note that since $\mathcal{C} \in \mathcal{S}_r$, by Lemma 3 there is a block computed by an honest party among the first and the last $\ell + 2\Delta$ rounds of the epoch; the inequality follows by Chain Growth. The next one follows from Definition 11, since $|S| \leq s$ due to Lemma 4, and so $n(S) \geq n_v|S|/\gamma$. The last inequality is a consequence of Condition C2.

Corollary 2. $\textsc{GoodRounds}(r-1) \implies \textsc{GoodRounds}(r)$.

Proof. Consider any $\mathcal{C} \in \mathcal{S}_r$ and let u be its last recalculation point before r and T the associated target. Note that if r is a recalculation point, it follows directly by Lemma 5 that it is good. Otherwise, we need to show that $f/2\gamma^2 \leq pn_r T \leq (1+\delta)\gamma^2 f$. By Lemma 5, $f/2\gamma \leq pn_u T \leq (1+\delta)\gamma f$. By Lemma 4, $n_u/\gamma \leq n_r \leq \gamma n_u$. Combining these two bounds we obtain the desired inequality.

Theorem 2. *Consider a typical execution in a $(\gamma, 2(1+\delta)\gamma^2 m/f)$-respecting environment. If the Conditions C1 and C2 are satisfied, then all predicates of Definition 9 hold.*

Proof. We only need to verify that the predicates hold for the first $\ell + 2\Delta$ rounds, assuming they hold at the first round. Note that if no epoch has been completed, all honest parties query for target T_0 and are at most γn_0. Thus, we only need to verify $\textsc{Duration}(\ell + 2\Delta)$. The lower bound of Lemma 4 holds unless $\textsc{GoodRounds}(r)$ fails for some $r < \ell + 2\Delta$, which does not happen by Corollary 2.

5.7 Common Prefix and Chain Quality

The following lemma says that two "longest" chains cannot diverge for too long. We say below that $d \in \mathbb{R}$ is *contained* in a block B and write $d \in B$, when B extends a chain $\mathcal{C}$ and $\mathrm{diff}(\mathcal{C}) < d \leq \mathrm{diff}(\mathcal{C}B)$.

Lemma 6. *For any round r of a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment and any two chains $\mathcal{C}$ and $\mathcal{C}'$ in $\mathcal{S}_r$, the timestamp of* $\mathrm{head}(\mathcal{C} \cap \mathcal{C}')$ *is at least $r - 2\ell - 4\Delta$.*

Proof. Let v be the timestamp of $\mathrm{head}(\mathcal{C} \cap \mathcal{C}')$ and $u \leq v$ the greatest timestamp among those blocks on $\mathcal{C} \cap \mathcal{C}'$ that was computed by an honest party. Let $U = \{i : u < i \leq r\}$, $S = \{i : u + \Delta \leq i \leq r - \Delta\}$, and let J denote the adversarial queries that correspond to the rounds in U. We claim that, if $r - v \geq \ell + 2\Delta$, then

$$2Q(S) \leq D(U) + A(J).$$

This contradicts Lemma 2 for $\delta \geq 8\epsilon$, which implies $D(U) < (1+5\epsilon)Q(S)$ and $A(J) < (1 - \delta + 3\epsilon)Q(S)$.

Towards proving the claim, associate with each $r \in S$ such that $Q_r > 0$ an arbitrary honest block that is computed at round r for difficulty Q_r. Let $\mathcal{B}$ be the set of these blocks and note that their difficulties sum to $Q(S)$. We argue the existence of a set of blocks $\mathcal{B}'$ computed in U such that $\mathcal{B} \cap \mathcal{B}' = \emptyset$ and $\{d \in B : B \in \mathcal{B}\} \subseteq \{d \in B : B \in \mathcal{B}'\}$. This suffices, because each block in $\mathcal{B}'$ contributes either to $D(U) - Q(S)$ or to $A(J)$ and so $Q(S) \leq D(U) - Q(S) + A(J)$.

Consider, then, a block $B \in \mathcal{B}$ extending a chain $\mathcal{C}^*$ and let $d = \mathrm{diff}(\mathcal{C}^* B)$. If $d \leq \mathrm{diff}(\mathcal{C} \cap \mathcal{C}')$ (note that $u < v$ in this case and $\mathrm{head}(\mathcal{C} \cap \mathcal{C}')$ is adversarial), let B' be the block of $\mathcal{C} \cap \mathcal{C}'$ containing d. Such a

block clearly exists and has a timestamp greater than u. Furthermore, $B' \notin \mathcal{B}$, since B' was computed by the adversary. If $d > \mathrm{diff}(\mathcal{C} \cap \mathcal{C}')$, note that there is a unique $B \in \mathcal{B}$ such that $d \in B$ (recall the argument in Chain Growth Lemma 1). Since B cannot simultaneously be on $\mathcal{C}$ and $\mathcal{C}'$, there is a $B' \notin \mathcal{B}$ either on $\mathcal{C}$ or on $\mathcal{C}'$ that contains d.

Theorem 3 (Common Prefix). *For a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment, the common-prefix property holds for parameter ϵm.*

Proof. Suppose common prefix fails for two chains $\mathcal{C}_1$ and $\mathcal{C}_2$ at rounds $r_1 \leq r_2$. It is not hard to see that in such a case there was a round $r \leq r_2$ and two chains $\mathcal{C}$ and $\mathcal{C}'$ in $\mathcal{S}_r$, such that each had at least k blocks after $\mathrm{head}(\mathcal{C} \cap \mathcal{C}')$. By Lemmas 4 and 6, at least $\epsilon m/2$ belong to one epoch. In view of Lemma 6, it suffices to show that that these were computed in at least $\ell + 2\Delta$ rounds. Let T be the target of these blocks and suppose the honest parties query the oracle for target T during a set of rounds S of size $\ell + 2\Delta$. By Condition C1, $|S| \leq \epsilon m/4(1+\delta)\gamma^2 f$. Furthermore, by Theorem 2, $pn_r T \leq (1+\delta)\gamma^2 f$ holds for each $r \in S$. Putting these together, the number of such blocks that the honest parties computed are less than

$$T \cdot D(S) < (1+\epsilon) \sum_{r \in S} pn_r T \leq (1+\epsilon)\epsilon m/4.$$

By Lemma 2 the adversary contributed less than $(1+\epsilon)\epsilon m/4(1+\delta)$ blocks, for a total of less than $\epsilon m/2$.

Theorem 4 (Chain Quality). *For a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment, the chain-quality property holds with parameters $\ell + 2\Delta$ and $\mu = \delta - 3\epsilon$.*

Proof. Let us denote by B_i the i-th block of $\mathcal{C}$ so that $\mathcal{C} = B_1 \dots B_{\mathrm{len}(\mathcal{C})}$ and consider K consecutive blocks $B_u, \dots, B_v$. Define K' as the least number of consecutive blocks $B_{u'}, \dots, B_{v'}$ that include the K given ones (i.e., $u' \leq u$ and $v \leq v'$) and have the properties (1) that the block $B_{u'}$ was computed by an honest party at some round r or is B_1 in case such block does not exist ($r = 0$), and (2) that there exists a round r' such that $B_1 \dots B_{v'} \in \mathcal{S}_{r'}$. Denote by d' the total difficulty of these K' blocks. Define $U = \{r, \dots, r'\}$, $S = \{r + \Delta, \dots, r' - \Delta\}$, and J the adversarial queries in U starting with the first to obtain one of the K' blocks. Let x denote the total difficulty of all the blocks from honest parties that are included in the K blocks and—towards a contradiction—assume $x < \mu d'$. In a typical execution, all the K' blocks $\{B_j : u' \leq j \leq v'\}$ have been computed in U. But then we have the following contradiction to Lemma 2(c).

$$A(J) \geq d' - x > (1-\mu)d' \geq (1-\mu)Q(S) = (1 - \delta + 3\epsilon)Q(S).$$

The first two inequalities follow from the definitions of x and d' and the assumed relation between them. It is not hard to see that the last inequality follows from Chain-Growth Lemma. Finally, to verify that this is indeed a contradiction, note that if $U > (1+\delta)\gamma^2 m/f$ we may use Lemma 3 to partition U appropriately (using blocks computed by honest parties as pivot points) and apply Lemma 2(c) to each part. This is valid, since a block computed by an honest party provides both properties (1) and (2) required for K'.

5.8 Nakamoto Consensus: Consistency and Liveness

For parameters that satisfy Conditions C1 and C2, we can show that a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment enjoys Consistency and Liveness. Their proofs follow along the lines of previous work, such as [11, 19, 10]. In particular, Consistency will follow directly from the Common-Prefix property, that we show to hold in the above circumstances.

Theorem 5 (Consistency). *For a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment, Consistency is satisfied by setting the settled transactions to be those reported more than ϵm blocks deep.*

Theorem 6 (Liveness). *For a typical execution in a $(\gamma, (1+\delta)\gamma^2 m/f)$-respecting environment, Liveness is satisfied for depth ϵm with wait-time $(4\gamma^2 + 1)\epsilon m/f$.*

Proof. We claim that the chain $\mathcal{C}$ of any honest party has at least ϵm blocks that where computed in the last $4\epsilon\gamma^2 m/(1-2\epsilon)f + 4\Delta$ rounds. Indeed, $\mathcal{C}$ must have a segment that lies in a single epoch—say of target T—and was computed in a set U of at least $2\epsilon\gamma^2 m/(1-2\epsilon)f + 2\Delta$ consecutive rounds. If S is its subset without the first and last Δ rounds, by Chain-Growth Lemma 1, the length of this segment is at least

$$T \cdot Q(S) > (1-\epsilon)[1-(1+\delta)f]^{\Delta} \sum_{r \in S} p n_r T \geq \frac{(1-2\epsilon)f|S|}{2\gamma^2} \geq \epsilon m.$$

Furthermore, if a transaction tx is included in any block computed by an honest party for the first $\ell + 2\Delta$ rounds, by Lemma 3, the chain $\mathcal{C}$ of any honest party contains tx in a block B. The total wait-time amounts to

$$\ell + 6\Delta + \frac{4\epsilon\gamma^2 m}{(1-2\epsilon)f} \leq \frac{\epsilon m}{2(1+\delta)f} + \frac{4\epsilon\gamma^2 m}{(1-2\epsilon)f} \leq (4\gamma^2+1) \cdot \frac{\epsilon m}{f}.$$

References

1. Adam Back. Hashcash. http://www.cypherspace.org/hashcash, 1997.
2. Lear Bahack. Theoretical bitcoin attacks with less than half of the computational power (draft). *IACR Cryptology ePrint Archive*, 2013:868, 2013.
3. Ran Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 13(1):143–202, 2000.
4. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2000. http://eprint.iacr.org/2000/067.
5. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 136–145. IEEE Computer Society, 2001.
6. Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
7. Cynthia Dwork, Nancy A. Lynch, and Larry J. Stockmeyer. Consensus in the presence of partial synchrony. *J. ACM*, 35(2):288–323, 1988.
8. Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In Ernest F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 139–147. Springer, 1992.
9. Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications. *IACR Cryptology ePrint Archive*, 2014:765, 2014.
10. Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 281–310. Springer, 2015.
11. Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol with chains of variable difficulty. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 291–323. Springer, 2017.
12. Vassos Hadzilacos and Sam Toueg. A modular approach to fault-tolerant broadcasts and related problems. Technical report, 1994.
13. Ari Juels and John G. Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *NDSS*. The Internet Society, 1999.
14. Aggelos Kiayias and Giorgos Panagiotakos. Speed-security tradeoffs in blockchain protocols. *IACR Cryptology ePrint Archive*, 2015:1019, 2015.
15. Lucianna Kiffer, Rajmohan Rajaraman, and Abhi Shelat. A better method to analyze blockchain consistency. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 729–744. ACM, 2018.
16. Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

17. Colin McDiarmid. *Probabilistic Methods for Algorithmic Discrete Mathematics*, chapter Concentration, pages 195–248. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.
18. Satoshi Nakamoto. Bitcoin open source implementation of p2p currency. http://p2pfoundation.ning.com/forum/topics/bitcoin-open-source, February 2009.
19. Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 643–673, 2017.
20. Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.
21. Ling Ren. Analysis of nakamoto consensus. *IACR Cryptology ePrint Archive*, 2019:943, 2019.
22. R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, Cambridge, MA, USA, 1996.
23. Fred B. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Comput. Surv.*, 22(4):299–319, 1990.
24. J.M. Steele and Mathematical Association of America. *The Cauchy-Schwarz Master Class: An Introduction to the Art of Mathematical Inequalities*. MAA problem books series. Cambridge University Press, 2004.

A Mathematical Facts

All the following definitions and statements assume finite probability spaces and random variables with finite means.

Definition 12. *[6, Definition 5.3] A sequence of random variables $(X_0, X_1, \dots)$ is a martingale with respect to the sequence $(Y_0, Y_1, \dots)$, if, for all $n \geq 0$, X_n is determined by $Y_0, \dots, Y_n$ and $\mathbf{E}[X_{n+1}|Y_0, \dots, Y_n] = X_n$.*

The following is closer to Theorem 3.15 in [17], but see also Theorems 8.1 and 8.2 in [6].

Theorem 7. *Let $(X_0, X_1, \dots)$ be a martingale with respect to the sequence $(Y_0, Y_1, \dots)$. Suppose an event G implies*

$$X_k - X_{k-1} \leq b \text{ (for all } k) \quad \text{and} \quad V = \textstyle\sum_k \mathbf{var}[X_k - X_{k-1}|Y_1, \dots, Y_{k-1}] \leq v,$$

Then, for non-negative n and t,

$$\mathbf{Pr}\big[X_n \geq X_0 + t \wedge G\big] \leq \exp\Big\{-\frac{t^2}{2v + 2bt/3}\Big\}.$$

The following Cauchy-Schwarz converse (see [24]) will be of use.

Theorem 8. *For all non-negative real numbers $a_k, b_k, k = 1, 2, \dots, n$ that satisfy $m \leq a_k/b_k \leq M$ for some constants $0 < m \leq M < \infty$,*

$$\sqrt{\sum_{k=1}^{n} a_k^2}\sqrt{\sum_{k=1}^{n} b_k^2} \leq \frac{A}{G}\sum_{k=1}^{n} a_k b_k,$$

where $A = (m + M)/2$ and $G = \sqrt{mM}$.

B Proof of (All Executions are Typical) Theorem 1 and the Related Lemma 2

Proof. Since the length L of the execution is fixed we will prove the stated bound for a fixed set of consecutive rounds S—or, with respect to the adversary, a fixed set of consecutive queries J—and then apply a union bound over all such sets in the length of the execution. Furthermore, we may assume $|S| \leq s$. This is because $\ell \leq s/2$ and we may partition S in parts such that each part has size between ℓ and s. We then sum over all parts to obtain the desired bound. Let us also fix an execution E_0 just before the beginning of S (or J).

We will prove that the statements fail with exponentially small probability for an arbitrary E_0. Note that E_0 determines the number of parties n_0 and t_0 at the beginning of S (or J) and the target $T(J)$ associated with the first query in J.

For each round $i \in S$, define a Boolean random variable F_i equal to 1 exactly when all n_i hash values that were returned to the queries of the honest parties were above $\min\{T : f(T, n_i) \geq (1+\delta)\gamma^2 f\}$; define $Z_i = Y_i \cdot F_{i+1} \cdots F_{i+\Delta-1}$. Let G denote the event that the rounds in S are good. Given G, for any $i \in S$, $(F_i = 1) \implies (D_i = 0)$ and so $Q_i \geq Z_i$. Thus, for any d,

$$\mathbf{Pr}\Big[G \wedge \sum_{i\in[k]} Q_i \leq d\Big] \leq \mathbf{Pr}\Big[G \wedge \sum_{i\in[k]} Z_i \leq d\Big],$$

and we may focus on the term on the right-hand side. Identify S with $\{1, \ldots, |S|\}$ and partition it with sets of the form $S_j = \{j, j+\Delta, j+2\Delta, \ldots\}$ for $j \in \{0, 1, \ldots, \Delta-1\}$. We will show that, for each part S_j,

$$\mathbf{Pr}\Big[G \wedge \sum_{i\in S_j} Z_i \leq (1-\epsilon)[1-(1+\delta)\gamma^2 f]^{\Delta} p \sum_{i\in S_j} n_i\Big] \leq e^{-\lambda}.$$

Let us fix such a set $S_j = \{s_1, s_2, \ldots, s_\nu\}$, with $\nu \geq \lfloor |S|/\Delta \rfloor$, and define the event G_t as the conjunction of the events G and $t = \epsilon(1-2\gamma^2 f)^{\Delta} pn(S_j)$. Note that $n(S_j) \leq L$ and so t ranges over a discrete set of size at most L and we can afford a union bound over it. Thus, it is sufficient to show that for any such t,

$$\mathbf{Pr}\Big[G_t \wedge \sum_{i\in S_j} Z_i \leq [1-(1+\delta)\gamma^2 f]^{\Delta} p \sum_{i\in S_j} n_i - t\Big] \leq e^{-\lambda}.$$

To that end, consider the sequence of random variables

$$X_0 = 0;\ \ X_k = \sum_{i\in[k]} Z_{s_i} - \sum_{i\in[k]} \mathbf{E}[Z_{s_i}|\mathcal{E}_{s_i-1}],\ \ k \in [\nu].$$

This is a martingale with respect to the sequence $\mathcal{E}_{s_1-1}(\mathcal{E}_0 = E_0), \ldots, \mathcal{E}_{s_\nu-1}, \mathcal{E}$, because (recalling basic properties of conditional expectation [17]),

$$\mathbf{E}[X_k|\mathcal{E}_{s_k-1}] = \mathbf{E}\big[Z_{s_k} - \mathbf{E}[Z_{s_k}|\mathcal{E}_{s_k-1}]\big|\mathcal{E}_{s_k-1}\big] + \mathbf{E}[X_{k-1}|\mathcal{E}_{s_k-1}] = X_{k-1}.$$

Specifically, the above follows from linearity of conditional expectation and the fact that X_{k-1} is a deterministic function of $\mathcal{E}_{s_{k-1}+\Delta-1} = \mathcal{E}_{s_k-1}$. Furthermore, given an execution E satisfying G_t,

$$\epsilon \sum_{i\in S_j} \mathbf{E}[Z_i|\mathcal{E}_{s_k-1} = E_{s_k-1}] \geq \epsilon \sum_{i\in S_j} [1-(1+\delta)\gamma^2 f]^{\Delta} pn_i = t.$$

Thus, our goal is to show $\mathbf{Pr}[-X_\nu \geq t \wedge G_t] \leq e^{-\lambda}$.

We now provide the details relevant to Theorem 7. Consider an execution E satisfying G_t and let B denote the event $\mathcal{E}_{s_k-1} = E_{s_k-1}$. Note that $Z_{s_k}^2 = Y_{s_k}^2 \cdot F_{s_k+1} \cdots F_{s_k+\Delta-1}$ and all these random variables are independent given B. Since $X_k - X_{k-1} = Z_{s_k} - \mathbf{E}[Z_{s_k}|\mathcal{E}_{s_k-1}]$ and

$$Z_{s_k} - \mathbf{E}[Z_{s_k}|B] \leq \frac{1}{T_{s_k}^{\min}} = \frac{pn_{s_k}}{pn_{s_k} T_{s_k}^{\min}} \leq \frac{\gamma pn(S_j)}{pn_{s_k} T_{s_k}^{\min}|S_j|} \leq \frac{\gamma pn(S_j)}{\nu f/(2\gamma^2)} \leq \frac{2\gamma^3 t}{\epsilon(1-2\gamma^2)^{\Delta} f\nu} \stackrel{\text{def}}{=} b, \tag{2}$$

we see that the event G implies $X_k - X_{k-1} \leq b$. With respect to $V = \sum_k \mathbf{var}[X_k - X_{k-1}|\mathcal{E}_{s_k-1}] \leq \sum_k \mathbf{E}[Z_{s_k}^2|\mathcal{E}_{s_k-1}]$, using the independence of the random variables and Proposition 1,

$$\sum_{k\in[\nu]} \mathbf{E}[Z_{s_k}^2|B] \leq [1-(1+\delta)\gamma^2 f]^{\Delta-1} \sum_{k\in[\nu]} \frac{(pn_{s_k})^2}{pn_{s_k} T_{s_k}^{\min}} \leq \frac{[1-(1+\delta)\gamma^2 f]^{\Delta-1}}{f/(2\gamma^2)} \cdot \sum_{k\in[\nu]} (pn_{s_k})^2.$$

Applying Proposition 2 on this bound, we see that event G_t implies

$$V \le \frac{2\gamma^3[1-(1+\delta)\gamma^2 f]^{\Delta-1}}{f|S_j|} \cdot \Big(\sum_{k\in[\nu]} pn_{s_k}\Big)^2 \le \frac{2\gamma^3 t^2}{\epsilon^2 f(1-2\gamma^2 f)^{\Delta+1}\nu} \stackrel{\text{def}}{=} v. \tag{3}$$

In view of these bounds (note that $bt < \epsilon v$), by Theorem 7,

$$\mathbf{Pr}[-X_\nu \ge t \wedge G_t] \le \exp\Big\{-\frac{t^2}{2v(1+\frac{\epsilon}{3})}\Big\} \le \exp\Big\{-\frac{\epsilon^2 f[1-(1+\delta)\gamma^2 f]^{\Delta+1}\nu}{4\gamma^3(1+\frac{\epsilon}{3})}\Big\} \le e^{-\lambda},$$

where for the last inequality we used Condition 1 (recall that $\nu \ge \ell/\Delta$).

For the bound on $D(S)$ it will be convenient to work per query. Let J denote the queries in S, $\nu = |J|$, and Z_i the difficulty of any block obtained from query $i \in J$. Define the martingale sequence

$$X_0 = 0;\ \ X_k = \sum_{i\in[k]} Z_i + \sum_{i\in[k]} \mathbf{E}[Z_i|\mathcal{E}_{i-1}],\ \ k \in [\nu].$$

With similar calculations as above we obtain that G_t (with $t = \epsilon p\nu$) implies

$$X_k - X_{k-1} \le \frac{2\gamma^3 t}{\epsilon f|S|} \stackrel{\text{def}}{=} b \quad \text{and} \quad V \le \frac{2\gamma^3 t^2}{\epsilon^2 f|S|} \stackrel{\text{def}}{=} v.$$

Applying Theorem 7 we obtain

$$\mathbf{Pr}\big[X_\nu \ge t \wedge G_t\big] \le \exp\Big\{-\frac{\epsilon t}{2b(1+\frac{\epsilon}{3})}\Big\} \le e^{-\lambda}.$$

We next focus on part (b). For each $j \in J$, let A_j be equal to the difficulty of the block obtained with the j-th query as long as the target was at least $T(J)/\tau$; thus, $A(J) = \sum_{j\in J} A_j$. If $|J| = \nu$, identify J with $[\nu]$ and define the martingale

$$X_0 = 0;\ \ X_k = \sum_{j\in[k]} A_j - \sum_{j\in[k]} \mathbf{E}[A_j|\mathcal{E}_{j-1}],\ \ k \in [\nu].$$

For all $k \in [\nu]$ we have $X_k - X_{k-1} \le \tau/T(J)$, $\mathbf{var}[X_k - X_{k-1}|\mathcal{E}_{k-1}] \le p\tau/T(J)$, and $\mathbf{E}[A_j|\mathcal{E}_{j-1}] \le p$. We may apply Theorem 7 with $b = \tau/T(J)$, $v = bp\nu \le bt/\epsilon$, and $t = \max\big\{\epsilon p\nu,\ 2(\frac{1}{\epsilon}+\frac{1}{3})b\lambda\big\}$. We obtain

$$\mathbf{Pr}\Big[\sum_{j\in J} A_j \ge p\nu + t\Big] \le \exp\Big\{-\frac{t}{2b(\frac{1}{3}+\frac{1}{\epsilon})}\Big\} \le e^{-\lambda}.$$

For part (c), as in [11], it can be shown that an insertion or a copy imply a collision, which can be shown to occur with probability at most $\binom{L}{2}2^{-\kappa}$. Also, since there can be at most L predicted blocks, the probability a prediction occurs is at most $L^2 2^{-\kappa}$.

Proof (Proof of Lemma 2). (a) The middle inequality follows directly from the definition of the random variables. For the other two note that $n(U \setminus S) \le \gamma \cdot \frac{n(S)}{|S|} \cdot |U \setminus S| \le \gamma \cdot \frac{n(S)}{\ell} \cdot 2\Delta$. Since $|J| \le (1-\delta)n(U)$ and

$$n(U) = n(S) + n(U \setminus S) \le \Big(1 + \frac{2\gamma\Delta}{\ell}\Big)n(S) < \Big(1 + \frac{\epsilon^2}{2}\Big)n(S),$$

the inequalities follow from Definition 11 and Condition C2.

(b) Either $\epsilon p|J| \ge \tau\alpha(J)$ and Definition 11 applies directly, or $p|J| < \tau\alpha(J)/\epsilon$ and by Equation (1) and Condition C1,

$$T(J) \cdot A(J) < \frac{2}{\epsilon^2} \cdot (1+\epsilon)\Big(1+\frac{\epsilon}{3}\Big)\tau\lambda < \frac{f\ell}{2\gamma^3} < \frac{\epsilon m}{4(1+\delta)\gamma^5}.$$

(c) In a (γ, s)-respecting environment, $\gamma n(S) \geq n_w|S|$. Incorporating this in Definition 11 we obtain the first bound. For the second one, since w is good, $pn_wT(J) \geq pn_wT_w^{\min} \geq f/2\gamma^2$. Using $\epsilon < 1/6$,

$$\epsilon(1-2\epsilon)pn(S) \geq \epsilon(1-2\epsilon)|S| \cdot \frac{pn_wT_w^{\min}}{\gamma T(J)} \geq \frac{\epsilon(1-2\epsilon)f\ell}{2\gamma^3 T(J)} > \tau\alpha(J).$$

As in (a), $p|J| \leq (1-\delta+\epsilon^2/2)pn(S)$. We obtain $A(J) \leq (1-\delta+\epsilon)pn(S)$ and use Condition C2.