On Fairness in Committee-Based Blockchains

Yackolley Amoussou-Guenou, Antonella Del Pozzo, Maria Potop-Butucaru, Sara Tucci-Piergiovanni

On Fairness in Committee-Based Blockchains

Yackolley Amoussou-Guenou
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

Antonella Del Pozzo
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

Maria Potop-Butucaru
Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

Sara Tucci-Piergiovanni
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

Abstract

Committee-based blockchains are among the most popular alternatives of *proof-of-work* based blockchains, such as Bitcoin. They provide strong consistency (no fork) under classical assumptions, and avoid using energy-consuming mechanisms to add new blocks in the blockchain. For each block, these blockchains use a committee that executes Byzantine-fault tolerant distributed consensus to decide the next block they will add in the blockchain. Unlike Bitcoin, where there is only one creator per block, in committee-based blockchain any block is cooperatively created. In order to incentivize committee members to participate in the creation of new blocks, rewarding schemes have to be designed. In this paper, we study the fairness of rewarding in committee-based blockchains and we provide necessary and sufficient conditions on the system communication under which it is possible to have a fair reward mechanism.

2012 ACM Subject Classification Computer systems organization → Dependable and fault-tolerant systems and networks

Keywords and phrases Blockchain, Consensus, Committee, Fairness, Proof-of-Stake, Reward, Selection

Digital Object Identifier 10.4230/OASIcs.Tokenomics.2020.4

Acknowledgements The authors thank Ludovic Desmeuzes for his work on the numerical examples.

1 Introduction

The blockchain technology is one of the most appealing technology since its introduction in the Bitcoin White Paper [31] in 2008. A blockchain is a distributed ledger, where information are stocked in blocks, and hashes link blocks in order to have a chain structure. Blockchain systems mostly use proof-of-work, where the first process that solves a crypto-puzzle can add a new block to the blockchain. First, this technique is highly energy consuming, and second, it does not ensure consistency, *i.e.* conserving the chain structure. Forks may happen and they lead to a tree structure. Some alternatives arisen to avoid at least one of these issues. For example, proof-of-stake based blockchains, where the probability of being able to add a block depends on the stake of a process; this alternative solves the energy consumption issue, but not the consistency one; they are also subject to the *nothing-at-stake* problem, where processes try to produce blocks on all the forks to ensure some rewards, and by doing so, do not resolve the forks. However, in [35], Saleh shows that the nothing-at-stake problem is mitigated. Other alternatives tackle both issues, for example, committee-based blockchains. In committee-based blockchains, for each height/block, a committee is selected and that committee uses a consensus algorithm to decide on the next block to append in the blockchain. By construction, committee-based blockchains are not subject to nothing-at-stake, since they ensure consistency (no fork).

To motivate processes to add and maintain the blockchain, rewarding mechanisms are in place. Because committee members can be faulty, rewarding mechanisms are inherently more complex to handle and their properties must be studied. Ad minimum, the rewarding mechanism must be fair, *i.e.* distributing the rewards in proportion to the merit of participants, where *merit* abstracts the notion of effort processes take for the construction of the blockchain [4], for instance it models the hashing power in Bitcoin.

Informally, we say that a blockchain protocol is fair if any *correct process* (a process that followed the protocol throughout the whole execution) that has α fraction of the total merit in the system will get at least α fraction of the total reward that is given in the system. Our fairness analysis, in line with Francez's definition of fairness [16], generally defines the fairness of protocols based on voting committees (e.g. Byzcoin [27], Hyperledger Fabric [11], PeerCensus [8], RedBelly [7], SBFT [18], Tendermint [2, 3, 5], etc.), actually studies fairness by separating the fairness of their *selection mechanism* and the fairness of their *reward mechanism*.

The selection mechanism is in charge of selecting the subset of processes that will participate to the agreement on the next block to append in the blockchain, while the reward mechanism defines the way rewards are distributed among processes that participate to the agreement. We propose a formal definition of fairness of selection mechanisms, and then we study the fairness of some selection mechanisms. The analysis of the reward mechanism allowed establishing the following fundamental result and necessary conditions with respect to the fairness of committee-based blockchains as follows:
There exists a(n) (eventual) fair reward mechanism for committee-based blockchains if and only if the system is (eventual) synchronous and faulty processes are detectable. (Theorems 11 and 13).

The rest of the paper is organized as follows. In Section 2, we compare the existing studies of fairness in blockchain systems; in Section 3, we define the system model; in Section 4, we give the basics of committee-based blockchain systems; in Section 5, we study the fairness in committee-based blockchain systems; in Section 6, we analyze some behaviors and the impact of the communication model on rewards; and in Section 7, we conclude.

2 Related Work

The closest work in blockchain systems to our fairness study (however very different in its scope) is the study of the *chain-quality*. In [17], Garay *et al.* define the notion of *chain-quality* as the proportion of blocks mined by honest miners in any given window; Garay *et al.* study the conditions under which during a given window of time, there is a bounded ratio of blocks in the chain that malicious players produced, over the total blocks in the blockchain. Kiayias *et al.* in [25] propose Ourobouros [25] and analyze its chain-quality property. In [33], Pass and Shi propose a notion of *fairness* which is an extension of the chain-quality property, they address one of the vulnerabilities of Bitcoin studied formally in [12, 13]. In [12, 13], Eyal and Sirer prove that if the adversary controls a coalition of miners holding even a minority fraction of the total computational power, this coalition can gain twice its share. Fruitchain [33] overcomes this problem by ensuring that no coalition controlling less than a majority of the computing power can gain more than a factor $1 + 3\delta$ by not respecting the protocol, where δ is a parameter of the protocol. We note that in their model, only one process creates a block in the blockchain, and that process has a reward for the created block. In [20], Guerraoui and Wang study the effect of the delays of message propagation in Bitcoin, and they show that in a system of two miners, one can take advantage of the delays and be

rewarded exponentially more than its share. We extend the definition of [33] for systems where each block is produced by a subset of processes. This is the case of Tendermint [5] or Hyperledger Fabric [11] for example, where for each block there is a subset of processes, *a committee* that produces that block.

In [22, 21], Gürcan *et al.* study the fairness from the point of view of the processes that do not participate to the construction of the blockchain. Herlihy and Moir do a similar work in [23] where the authors study users' fairness and consider as an example Tendermint. Herlihy and Moir discussed how processes with malicious behavior could violate fairness by choosing transactions, and then they propose modifications to the original Tendermint to make some violations detectable and accountable. In [29], Lev-Ari *et al.* study fairness on transactions in committee-based blockchains with synchronous assumptions by using a detectable communication abstraction allowing them to identify malicious participants. Our work does not study fairness in the point of view of users.

Recent works consider the distribution of rewards in proof-of-stake based blockchains. In [28], Lagaillardie *et al.* show that even if Tendermint is unfair, the presence of delegators helps in the growth of the system. In [14], Fanti *et al.* define equitability, which represents the evolution of the fraction of total stakes of nodes, in particular, they compute the effect of so-called *compounding* where rewards are directly re-invest in stakes. Karakostas *et al.* define in [24] egalitarianism. Egalitarianism means that each node, no matter its stake fraction, wins the election process to append a new block the same amount as everyone. In this work, we focus on fairness, where nodes with higher merit should get more rewards. Our notion of fairness is different that egalitarianism, since the goal of egalitarianism is to have all nodes rewarded the same, no matter their merit.

3 System Model

The system is composed of an infinite set $\Pi = \{p_1, p_2, \dots p_i, \dots\}$ of sequential processes; i is the *index* of p_i. *Sequential* means that a process executes one step at a time. This does not prevent it from executing several threads with an appropriate multiplexing. As local processing time are negligible with respect to message transfer delays, we consider it as being equal to zero.

Arrival model. We assume a *finite arrival model* [1], *i.e.* the system has infinitely many processes but each run has only finitely many. The size of the set $\Pi_\rho \subset \Pi$ of processes that participate in each system run is not a priori-known. We also consider a finite subset $V \subseteq \Pi_\rho$ of committee members. The set V may change during any system run and its size $|V| = n$ is a priori known. A process is promoted in V by a selection function. Such selection function can be based for instance on stakes in proof-of-stake blockchains, or computing power in proof-of-work blockchains.

Time assumptions on communication. The processes communicate by exchanging messages through an eventually synchronous network [10]. *Eventually Synchronous* means that after a finite unknown time τ there is an a priori unknown bound δ on the message transfer delay. When $\tau = 0$ and δ is known the network is *synchronous*.

Failure model. Some processes can exhibit a Byzantine behavior [34] in the system. A Byzantine process is a process that deviates arbitrarily from the given protocol. We do not assume any bound on their number in the system, but up to f committee members

can exhibit a Byzantine behavior at each point of the execution. A process that exhibits a Byzantine behavior is called a Byzantine or a *faulty* process. A process that follows the given protocol is called *correct*.

Communication primitives. In the following, we assume the presence of a broadcast primitive. The primitive broadcast() is a best effort broadcast, which means that when a correct process broadcasts a value, eventually all the correct processes deliver it. A process p_i receives a broadcast of a message by executing the primitive delivery(). Messages are created with a digital signature, and we assume that digital signatures are unforgeable, so when a process p_i delivers a message, it knows the process p_j that created the message.

4 Committee-based Blockchains

Any committee-based blockchain uses instances of consensus solving a form of repeated consensus. This way, each committee agrees on a single value to avoid forks. Anceaume *et al.* [4] proved that classical distributed consensus is required to avoid forks.

Each correct process outputs an infinite sequence of decisions called the *output* of the process. More formally, as described by Delporte-Gallet *et al.* [9], and generalized to Byzantine failures in [2], an algorithm implements a repeated consensus if and only if it satisfies the following properties: (i) *Termination:* Every correct process has an infinite output. (ii) *Agreement:* If the i^{th} value of the output of a correct process is B, and the i^{th} value of the output of another correct process is B', then $B = B'$. (iii) *Validity:* Each value in the output of any correct process is valid with respect to a predefined predicate.

Detailed Description of the Algorithm

We denote by $\mathbb{B}$ the set of all blocks. A block contains, among other things, a header and a list of transactions. Let $bc \in \mathbb{B}^*$, be a finite sequence of blocks. $|bc|$ is the length (the number of blocks) of bc. We say that bc is a *blockchain* if $\forall k \in \mathbb{N} : 0 < k \leq |bc|$, in the header of the block at position k in bc, there is the hash of the block at position $k-1$. If additionally, the list of transactions in each blocks in the blockchain is valid with respect to the given application, we say that bc is a *valid* blockchain. The block at position 0 is the *genesis block*. Each process has a non-negative *stake*, which is the total amount of token it has. $\forall h > 0$, let V_h be the set of committee members for the height h. $\forall h > 0$, we assume that $|V_h| = n$, the size of committee member is fixed and equal to n.

The genesis block initializes the blockchain, selects the committee that will produce the block at position 1, describes how rewards will be distributed among committee members (which we call the *reward mechanism*), gives the initial distribution of stakes, and describes how processes will be selected for being part of committees with respect to the state of the blockchain (the *selection mechanism*). These information should be public, and known by all processes such that with the history of the blockchain, all processes can always compute deterministically the sets of committee members. This preclude in this work to consider proof-of-work blockchains (as in Bitcoin) since the computing power of the processes is not known by all processes nor verifiable.

For a height, processes not in the corresponding committee just wait for the decision from the committee members. The committee members for that height execute the consensus algorithm to decide on the block for that height. Once a process decides on a block, it sends the decided block to the whole network, and moves to the next height. Non-committee members wait to collect enough times the same decided block from the committee members,

in order to be tolerant to failures, and then move to the next height. When moving to the next height, processes wait a certain amount of time to collect more messages from committee members. These messages are the ones used to reward the previous committees. Intuitively, if a process receives a decision message for the decided block by a committee member, then probably that committee member followed the protocol during that height[1]. This allows implementing the repeated consensus. In [2], the authors formalized and proved correct the Tendermint repeated consensus algorithm, an example of committee-based blockchain protocol.

In this paper, we analyze the fairness of committee-based blockchains as described in the following section.

5 Fairness of Committee-based Blockchains

Chain Quality and Committee-based Blockchains

In the blockchain literature, chain quality has been defined by [17], and extended by [33], to study fairness in Bitcoin-like systems. A blockchain system has the property of chain quality if the proportion of blocks produced by honest processes in any given window, is proportional to their relative mining power. Intuitively, chain quality ensures that malicious processes do not produce more blocks than their proportion of mining power. One of the main differences between Bitcoin-like system and the committee-based blockchains is that in the former, one process produces a block, whereas in committee-based blockchain, a committee (a set) of processes produces a block. A committee is not necessarily composed only of correct processes, but can contain Byzantine or correct processes (with a correctness hypothesis of having some majority of the members following the protocol). We cannot apply the definition of chain quality to committee-based blockchain. Instead of defining the fairness with the blocks, we will define it relative to the proportion of total rewards a process gets.

Informally, we say that a blockchain protocol is fair if any *correct process* (a process that followed the protocol) that has a fraction α of the total merit in the system will get at least α fraction of the total reward that is given in the system. In order to tackle the fairness of a committee-based blockchain protocol such as HotStuff [36], Hyperledger Fabric [11], Redbelly [7], SBFT [18] or Tendermint [2, 3, 5], we split the mechanism in two: the *selection mechanism* and the *reward mechanism*. We say that each process has a given merit, which represents the effort the process is putting to maintain the blockchain, for instance it can represent the mining power of a process in proof-of-work blockchains, or the stakes in proof-of-stake blockchains, etc. The *selection mechanism* selects for each new height the committee members (the processes that will run the consensus instance) for that height. The *reward mechanism* is in charge to distribute rewards to committee members that produce a new block. Informally, if the selection mechanism is fair, then each process will become committee member proportionally to its merit, and if the reward mechanism is fair then for each height, only the correct committee members get a reward. By combining the two mechanisms, a correct process is rewarded at least a number proportional to its merit parameter over the infinite execution of the system.

[1] That is not true in general, since Byzantine processes, for instance, can send the decided value at the end without doing anything during the protocol execution.

5.1 Selection Mechanism

5.1.1 Definition and Fairness of Selection Mechanisms

In a system where the size of the committee is strictly lower than the number of processes in the system, there should be a way to select the members of the committees. Always selecting the same processes is a way to centralize the system. That set of processes can exercise a power of oligarchy, and add in the blockchain only transactions they want.

Formally, a selection mechanism is the function $\mathsf{selection} : \mathbb{B}^+ \times \mathbb{N} \to \Pi^n \cup \emptyset$, where n is the size of committees, Π the set of processes, and $\mathbb{B}^+$ represents the set of non-empty blockchains such that if bc is a non-empty blockchain, then:

$$\mathsf{selection}(bc, h) = \begin{cases} V_h, & \text{if } |bc| \geq h - 1 \\ \emptyset, & \text{otherwise} \end{cases},$$

where we recall that $|bc|$ means the length of the blockchain bc, and V_h is the set of committee members for height h.

Some information can be computed and/or stored in the blockchain, for instance the number of time each process has been committee member, or the stakes of each process that represents their wealth. The selection mechanism can select, based on these information, for instance, processes with the highest stakes, or processes that were committee members less often, or always the same set of processes, etc.

To abstract the notion of effort of a process, we denote by $\alpha_i(t) \in [0, 1]$ the merit parameter of p_i at time t proportionally to the total merit at time t, such that $\forall t, \sum_{p_i \in \Pi_\rho} \alpha_i(t) = 1$.

If $\forall t \in \mathbb{N}, \forall i, \alpha_i(t) = \alpha_i(0)$, we denote by α_i the merit of the process p_i. That means that the merits do not depend on the evolution of the blockchain, nor on its contents. Let v_i be the number of times p_i becomes committee member, proportionally to the number of blocks, so $v_i \in [0, 1]$. We propose the following definition of fairness of selection mechanisms where merits are fixed and do not change with time. The definition allows all processes with positive merit to be member of committees infinitely often, and with respect to their merit.

▶ **Definition 1** (Fairness of Selection Mechanism)**.** *Assume that the blockchain is built infinitely, so $\forall h \geq 0$, there is a block at position h. We say that a selection mechanism is fair if it respects the following properties:*

1. *If $\alpha_i \neq 0$ then $v_i \neq 0$; or equivalently, $\alpha_i \neq 0 \implies \forall h \geq 0, \exists h' \geq h : p_i \in V_{h'}$*
2. *If $\alpha_i \geq \alpha_j$ then $v_i \geq v_j$.*

Informally, Condition 1 means that each process with a positive merit parameter should become a committee member infinitely often. Condition 2 means that a process with a low merit cannot be selected more than a process with a higher merit. Note that this definition depends only on the merit and not on the behavior of the processes (correct or Byzantine).

A definition of fairness of a generic selection mechanism (that does change over time) is still an open question, but such definition should encapsulate the Definition 1 as special case.

▶ Remark 2. If the total number of processes in the system is equal to the size of committees, then all processes are always selected, so the selection mechanism in that case is trivially fair, although asking a huge set of processes to run the consensus is not scalable.

5.1.2 Examples of Selection Mechanisms

In this section, we briefly study different possible selection mechanisms. Let us assume that there are $N > n$ processes during the whole execution of the system and processes cannot enter nor exit. Let us also assume that merits do not change over time, $\forall t \in \mathbb{N}, \forall i, \alpha_i(t) = \alpha_i(0)$ and that all processes have the same merit, $\forall i, j \in \Pi_\rho, \alpha_j = \alpha_j > 0$.

In the examples below, we consider selection mechanisms that depends on the stakes of processes, while the merit is fixed and does not depends on the (evolution of) stakes. All processes are correct, and a committee member is rewarded when the committee it is part of produces a block. The reward increases the stake of the process. For our analysis, we further consider that processes are ordered by their stake and their id (public address for instance). Let us assume that at the beginning of the execution, all processes have the same amount of stake. Without loss of generality, and up to renaming, consider also that during the execution, if $\exists i, j : i < j$ such that p_i and p_j have the same amount of stakes, then p_i is selected before p_j.

Select the processes with the highest stake

This selection mechanism works as follows: for any height h, with respect to the blockchain up to height $h-1$, the n processes having the biggest amount of stakes are selected to be part of the committee.

This mechanism lead to the situation where only the n processes selected first will always be selected, and the other processes will never be. This mechanism is not fair, since Condition 1 is not satisfied. The processes not selected have, by assumption, a positive stake, and so they should be selected infinitely often.

Note that in the special case where there are at most n processes with a positive stake, and they are all selected, then fairness of selection is satisfied.

Select the processes with lowest stake

This selection mechanism works as follows: for any height h, with respect to the blockchain up to height $h-1$, the n processes having the lowest amount of stakes are selected to be part of the committee.

If all N processes are part of the system since the beginning, the number of times each process has been selected after l blocks is on average $l * n/N$ selections. This mechanism is fair according to the Definition 1.

Let us remark that this mechanism is fair with the model considered in this example, since processes cannot exit nor enter during the execution. If that assumption is removed, the following can happen. If processes could enter or leave, once a process is selected and rewarded, it knows that it would not be selected before a long period of time, on average after n/N blocks, one incentive could be to create new sub addresses (processes) such that they will have low stakes and it will be selected more often. Another similar scenario is that if a process has a big amount of stakes, it might not be allowed to participate in committees for a long time until all the process with small stakes caught up. The process might want to split into lot of stakeholders with small stakes. In such a way, the new stakeholders will always have the smallest stakes and be selected, if it continues to do so, it might block other processes to be committee members. Therefore, selecting the processes with the lowest stakes is not stable in an open setting.

▶ Remark 3. An unfair selection mechanism can lead to a centralization of the system, by always letting the same processes decide on the next block. Although the assumption on the bound of Byzantine processes does not depend on the selection mechanism, we note that when a selection mechanism selected the processes with the lowest amount of stakes, and only correct processes in committees are rewarded, at some point processes that were non-correct will have the least stake, and thus be selected.

5.2 Reward Mechanism

5.2.1 Definition of Reward Mechanism

In blockchain systems, processes that produce and add blocks to the blockchain are rewarded. In a committee-based blockchain, a committee is the producer of the block. Within that committee, some processes may not behave as prescribed. There may be different ways of rewarding members of committees. To do so, we define the *reward mechanism*. A reward mechanism consists of the reward function defined as follows: $\mathsf{reward} : \mathbb{B}^+ \times \mathcal{P}(\mathit{Information}) \times \mathbb{N} \to \mathbb{R}^{|\Pi|} \cup (\bot, \ldots, \bot)$, where $\mathcal{P}(\mathit{Information})$ is the power set of all messages, and we recall that Π is the set of processes, and $\mathbb{B}^+$ represents the set of non-empty blockchains.

If bc is a blockchain and $|bc| < h$, $\mathsf{reward}(bc, M, h) = (\bot, \ldots, \bot)$. Otherwise, it assigns to each process a given reward. In $\mathsf{reward}(bc, M, h)$, M represents the set of messages received, h the height of the blockchain bc where the reward of committee member is computed.

A reward is considered allocated if it is written in the blockchain. The second part of the reward mechanism is to choose when to allocate the rewards corresponding for a given height. If a reward has been allocated at a height h, the process can use it after a certain number of blocks defined in the genesis block (e.g. [17, 13]). We consider that for each production of block its rewards are allocated in exactly one block and not over different blocks, such that after the allocation of rewards a process knows if it has been rewarded or not. Note that once rewards are allocated, they cannot change anymore. Some blockchain systems add punishment mechanism, called *slashing*, to afflict, afterwards, some costs to a process if there is a proof of some misbehavior, as described in [32].

5.2.2 Fairness of Reward Mechanism

Let p_i be a process, and let T be a fragment of p_i's execution. If at the beginning of the fragment T, the internal state of p_i is correct and p_i follows the given protocol during the fragment T, then we say that p_i is T-correct. A correct internal state is a state of the process that can be the result of p_i following the protocol. A correct process is T-correct $\forall T$. For example, p_i is considered h-correct if during its execution of height h it followed the protocol.

We define the following properties for characterizing the fairness of a reward mechanism. Let h be a height. Each committee member has a boolean variable r_i^h, which we call *reward parameter* defined as follows:

1. If p_i is a not a committee members for h, then $r_i^h = 0$,
2. **h-completeness.** If p_i is a committee members for h and p_i is h-correct, then $r_i^h = 1$,
3. **h-accuracy.** If p_i is a committee members for h and p_i is not h-correct, then $r_i^h = 0$.

If $r_i^h = 0$, it means that p_i is not rewarded for height h, and if $r_i^h = 1$, p_i has been rewarded for h. The properties are inspired by classical properties of failure detectors [6].

▶ Remark 4. Note that we do not reward non-committee members. In this article, we do not consider delegations. When a process delegates to a committee member, once the committee member is rewarded, all of its delegates are rewarded proportionally to what they delegated. In future works, we may consider the case of committee-based blockchains with delegation. To do so, r_i^h must contain more information and not just be a boolean variable.

▶ **Definition 5** (Complete Fairness of a Reward Mechanism)**.** *Let $\mathcal{R}$ be a reward mechanism. If $\forall h > 0$, $\mathcal{R}$ satisfies Conditions 1 and h-completeness (Condition 2), we say that $\mathcal{R}$ satisfies complete fairness.*

If a reward mechanism satisfies complete fairness, it means that for all height $h > 0$, all h-correct committee members are rewarded, and non-committee members are not.

▶ **Proposition 6.** *There exists at least one reward mechanism satisfying complete fairness.*

Once a block is in the chain, rewarding all committee members, in the next block, for that block and only them, satisfy Conditions 1 and 2. Condition 1 is satisfied since for all height, non-committee members are not rewarded. Condition 2 also holds, for any given height h, all committee members of h are rewarded, in particular all h-correct committee members.

▶ **Definition 7** (Accurate Fairness of a Reward Mechanism)**.** *Let $\mathcal{R}$ be a reward mechanism. If $\forall h > 0$, $\mathcal{R}$ satisfies conditions 1 and h-accuracy (Condition 3), we say that $\mathcal{R}$ satisfies accurate fairness.*

If a reward mechanism satisfies accurate fairness, it means that for all height $h > 0$, all non h-correct committee members are not rewarded.

▶ **Proposition 8.** *There exists at least one reward mechanism satisfying accurate fairness.*

Never allocating rewards satisfies Conditions 1 and 3. Condition 1 is satisfied since non-committee members are not rewarded. Condition 3 holds, since no process is rewarded. In particular for any given height $h > 0$, no non h-correct committee members is rewarded.

Although simple and trivial to satisfy either complete fairness or accurate fairness, satisfying both at the same time is more complex and not always possible.

▶ **Definition 9** (Fairness of a Reward Mechanism)**.** *Let $\mathcal{R}$ be a reward mechanism. If $\forall h > 0$, $\mathcal{R}$ satisfies Conditions 1, h-completeness (Condition 2) and h-accuracy (Condition 3), we say that $\mathcal{R}$ is fair.*

We say that a reward mechanism is fair when at each height h, all and only h-correct committee members are rewarded.

▶ Remark 10. $\forall h > 0$, if $|V_h| > 1$, then for a reward mechanism to be (eventually) fair, rewards cannot be allocated directly in the block. For any height $h > 0$, the set of h-correct committee members cannot be known in advance. If $\forall h > 0, |V_h| = 1$, the reward can be directly given to the only committee member, so in the block at height h.

▶ **Theorem 11.** *There exists a fair reward mechanism in a committee-based blockchain protocol iff the system is synchronous.*

Proof. We prove this theorem by double implication.

- If there exists a fair reward mechanism, then the system is synchronous.
  Let $\mathcal{R}$ be a reward mechanism. By contradiction, we assume that $\mathcal{R}$ is fair and that the system is not synchronous.
  V_h is the set of committee members for the height h. Let $k > 0$ be the number of blocks to wait before distributing the rewards for V_h. The reward is allocated by the committee $V_{h'}$, where $h' = h + k$. Since the system is not synchronous, the committee members of height h', $V_{h'}$, may not receive all messages from V_h before allocating the rewards.
  By Conditions 1, and 2, since the reward mechanism is fair, by Conditions 1 - 3, all and only the h-correct committee members of the height h have a reward parameter equal to 1. That means that the h'-correct committee members of $V_{h'}$ know exactly who were the h-correct committee members in V_h, so they got all the messages before giving the reward. Contradiction, so the system is synchronous.

- If the system is synchronous, then there exists a fair reward mechanism.
  We assume that the system is synchronous and $\forall h > 1$, all messages sent by h-correct processes at height h are delivered before the block at height $h+1$. Let $\mathcal{R}$ be the following reward mechanism: let h be a height. Rewards for a block at height h are allocated at height $h+1$ by the committee V_{h+1}.
  - If a process is not a committee member for height h, set its reward parameter to 0, this is known since processes are already at height $h+1$.
  - By combining the messages from committee member of h processes, since the communication system is synchronous, it is possible to differentiate between h-correct and non h-correct committee members, and so set the reward parameter of h-correct committee members of h to 1; and set the reward parameter of non h-correct committee members of h to 0.

  By construction, the committee members in $h+1$ allocates rewards to all and only h-correct committee members, so $\mathcal{R}$ is fair, it satisfies all fairness Conditions 1 - 3.

$\square_{Theorem\ 11}$

If there is no synchrony, there cannot be a fair reward mechanism for committee-based blockchains. Our definition of fairness states that for any height Conditions 1-3 are satisfied. Processes should always receive all rewards they deserve. This definition can be weakened.

▶ **Definition 12** (Eventual Fairness of a Reward Mechanism). *Let $\mathcal{R}$ be a reward mechanism. If $\exists h_0 > 0 : \forall h \geq h_0$, $\mathcal{R}$ satisfies Conditions 1, h-completeness (Condition 2) and h-accuracy (Condition 3), we say that $\mathcal{R}$ is eventually fair.*

A reward mechanism is eventually fair if after an unknown time, the rewards are allocated to and only to correct committee members.

We note that if a reward mechanism is fair, then it is eventually fair but the reverse (reciprocal) is not necessarily true.

Detectable Byzantine Processes

In synchronous systems, it is always possible to detect Byzantine processes, for example using the broadcast abstraction detectable all-to-all (DA2A) defined in [29]. Detecting Byzantine processes allows to not reward them, and then to satisfy Condition 3. On the other hand, in eventual synchronous systems, the problem is more difficult. As in this work, Kihlstrom et al., in [26] distinguish between detectable and non-detectable Byzantine. The detectable Byzantine are the processes whose behavior can be detected, for instance by doing omission or commissions failures. Non-detectable Byzantine are Byzantine processes whose fault cannot be detected, for example processes that alter their internal state. In [19], Greve et al. extend that approach and propose a failure detector for detectable Byzantine in dynamic networks.

Note that although Kihlstrom *et al.* proposed in [26] a failure detector for solving consensus, our problem is not the same, and we cannot apply their failure detector as it is. In [26], once a process has a detectable Byzantine behavior, it should be suspected forever. In our model, we do not want to punish indefinitely Byzantine processes. In fact, we want for any height h to not reward only processes that were not h-correct. For example, let p_i be a process such that it is part of committees h and h', such that $h' > h$. Suppose also that during height h, p_i sends contradictory messages (and so is Byzantine), but then recovered before the beginning of height h' and follows the protocol during all h'. Even if p_i is not h-correct, it recovered before h' and is h'-correct. If p_i has been detected and not rewarded for height h, that should be taken into consideration of its work during height h', and since it follows the protocol, it should be rewarded for height h'. The failure detector proposed by Kihlstrom *et al.*, once a process has been suspected, marks such process as Byzantine forever.

▶ **Theorem 13.** *There exists an eventual fair reward mechanism in a committee-based blockchain protocol iff the system is (eventually) synchronous and Byzantine processes are detectable.*

Proof. We prove this theorem by double implication.

- If there exists an eventual fair reward mechanism, then the system is eventually synchronous or synchronous and Byzantine processes are detectable.
  Let $\mathcal{R}$ be a reward mechanism. We assume that $\mathcal{R}$ is eventually fair.
  If $\mathcal{R}$ is fair, by Theorem 11, the communication is synchronous, which ends the proof. Otherwise, since $\mathcal{R}$ is eventually fair, that means that there is a point in time h from which all the rewards are correctly allocated, so for any height $h' \geq h$, h'-correct committee members of committees at height h' are able to distinguish between non-correct processes during the height they are distributing the rewards, the Byzantine are then detectable. If we consider h as the beginning of the execution, then we have that $\mathcal{R}$ is fair, and by Theorem 11, the message delay is upper bounded. We have that after h, the message delay is upper bounded, so the communication is eventually synchronous. Therefore, the Byzantine are then detectable, and the communication is synchronous or eventually synchronous.
- If the system is eventually synchronous or synchronous, and Byzantine processes are detectable, then there exists an eventual fair reward mechanism.
  If the system is synchronous, the proof follows directly from Theorem 11. Consider that the system is eventually synchronous, but not synchronous. Let $\mathcal{R}$ be the following mechanism: Let h be a height. Rewards for a block at height h are allocated at height $h+1$ by the committee V_{h+1}.
  - If a process is not a committee member for height h, set its reward parameter to 0, this is known since processes are already at height $h+1$.
  - By combining the messages from committee member of h processes, if there is not sufficient information to detect the behavior of processes, reward only those detected as h-correct and in V_h, and the process proposing the distribution of reward increases the duration to wait before starting the next height. If there is enough information to detect the behavior of all processes in V_h, then reward the h-correct processes in V_h and do not reward non h-correct processes in V_h.

  $\mathcal{R}$ is eventually fair.

□*Theorem* 13

▶ **Corollary 14.** *In an asynchronous system, there is no (eventual) fair reward mechanism in a committee-based blockchain tolerating Byzantine processes.*

Proof. Assume that the system is asynchronous, where there are good periods such that consensus can be reached. By contradiction, let $\mathcal{R}$ be an eventual fair reward mechanism.

- If there are non-detectable Byzantine processes in the system, $\mathcal{R}$ is not fair (Theorem 13);
- If all Byzantine processes are detectable, then by Theorem 13, the system must be synchronous, or eventually synchronous.

We have a contradiction, since the system is asynchronous. It is not possible to have an (eventual) fair reward mechanism in an asynchronous system. □*Corollary* 14

Note that this result holds even if all the processes are correct but not known in advance, and the protocol tolerates Byzantine faults. In fact, it is different from the FLP impossibility result of consensus in an asynchronous system with one faulty process [15].

6 Numerical Examples

In this section, we examine the impact of different communication models on the fairness of reward mechanisms through several numerical examples that confirm the results on the fairness of reward mechanisms from Section 5.2.

Execution. In our analyses, processes run a committee-based blockchain protocol as described in Section 4, and rewards for a block produced at a height h are allocated in the block at height $h+1$. Note that the consensus module is Byzantine fault tolerant. We highlight the environment's important characteristics: the communication system, the total number of processes in the system, the size of each committee, the different type of processes and their number at a given height, the rewarding mechanism, and the selection mechanism. We must choose the value of these parameters before launching the execution. We consider different communication systems, and rewards are allocated by the next committee by using messages they delivered from the previous height – use the combination of all messages and check if they correspond to the correct time and a possible value to send according to the state.

We consider a system where all processes are part of all consensus instances. For clarity, and without loss of generality, we consider a system with $n \geq 4$ processes where all are selected. As stated in Remark 2, selecting all processes is a fair selection mechanism. We can now focus on the impact of the network on rewards. For any height h, there can be at most $\lfloor (n-1)/3 \rfloor$ non h-correct processes in each committee. For a committee, a quorum of $\lceil 2n/3 \rceil$ is needed for any decisions. In the case where there are for any height h some non h-correct processes, we assume that processes have enough information to detect it when distributing rewards. In particular, and for the experiment the Byzantine are specially tagged, and that tag is used only for allocating rewards. When an h-correct process receives a message from a non h-correct, it suspects it, and broadcasts the information. When an h-correct process delivers at least $2\lfloor n/3 \rfloor + 1$ suspicions for a process, it considers it as non h-correct, and does not propose to reward it.

We use MATLAB [30] for our analysis. We analyze three different communication models. First, a synchronous communication, where there is no delay. Then we consider the two following semi-synchronous communication models (i) the system alternates between good and bad periods, where during good periods message delays are upper bounded, and (ii) from an unknown time, message delays are upper bounded (eventually synchronous model). We note that in all these models, consensus can be reached. In the good/bad model, progress for consensus instances are guaranteed during the good periods. Note that in the eventually synchronous model, once the global stabilization time (GST) happens all message delays are upper bounded. If for a process, the GST happens during height h, then for all height $h' > h$, the message delays are upper bounded.

In each configuration of the communication model, we ran the experiment 50 times, and took the mean. 0 represents if a process did not receive a reward, and 1 if the process received a reward for the corresponding height. Due to the space limitation, we only present the experiment of the eventual synchronous model.

Eventually Synchronous Model

We consider an eventual synchronous model where all processes are correct, and part of each committee instances. Recall that eventual synchronous is a system where after a finite but unknown time (the GST), message delay is upper bounded for the rest of the execution. In our examples, we consider that the GST happens during height 10.

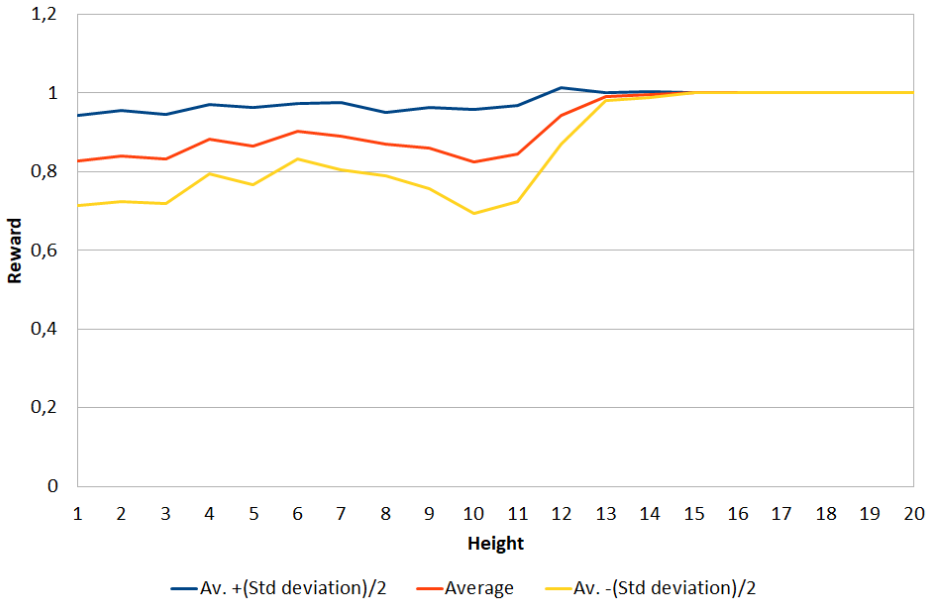

Figure 1 Evolution of Rewards in an Eventual Synchronous System, where Global Stabilization Time happens during height 10.

On Fig. 1, we present the evolution of reward for each height. We draw the mean of the average reward of each participant (red curve). The blue and yellow curves represent the standard deviation. We can see the set in which the processes are rewarded. When the blue and yellow curves converge, it means that all processes have on average the same reward. We notice that from height 10, the evolution of the reward is increasing. Approximately, from height 14, all processes are rewarded. Before height 10, there is a fluctuation in the evolution of rewards allocated because of the asynchronous periods; processes are not necessarily rewarded even if they participate. Their messages were not received on time. Once the GST happens, message delays become upper bounded but some processes still have a time-out shorter than the bound. These processes still increase their time-out until they receive all messages or detect incorrect behavior. When all processes deliver messages during their corresponding rounds, they allocate the rewards to all and only correct processes. It means that the reward mechanism is eventually fair.

7 Conclusions and Future Works

The originality of our contribution is the study of the impact of network conditions on the fairness of the rewarding in committee-based blockchains prone to Byzantine behavior. We proved that the reward mechanism is (eventually) fair iff the system communication is (eventually) synchronous and Byzantine processes are detectable. Our study opens interesting future research directions in particular the extension to other types of behaviors such as rational or amnesic. Furthermore, we are interested in studying the impact of network attacks on the fairness of rewarding. Another interesting direction is the design of self-adaptive fair rewarding schemes.

References

1 Marcos K Aguilera. A pleasant stroll through the land of infinitely many creatures. *ACM Sigact News*, 35(2):36–59, 2004.

2 Yackolley Amoussou-Guenou, Antonella Del Pozzo, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. Correctness of Tendermint-Core Blockchains. In *OPODIS 2018, December 17-19, 2018, Hong Kong, China*, pages 16:1–16:16, 2018.

3 Yackolley Amoussou-Guenou, Antonella Del Pozzo, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. Dissecting tendermint. In *Networked Systems - 7th International Conference, NETYS 2019, Marrakech, Morocco, June 19-21, 2019*, pages 166–182, 2019.

4 Emmanuelle Anceaume, Antonella Del Pozzo, Romaric Ludinard, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. Blockchain Abstract Data Type. In *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2019, Phoenix, AZ, USA, June 22-24.*, pages 349–358, 2019.

5 E. Buchman, J. Kwon, and Z. Milosevic. The latest gossip on BFT consensus. *CoRR*, abs/1807.04938v1, July 2018. URL: `https://arxiv.org/abs/1807.04938v1`.

6 Tushar Deepak Chandra and Sam Toueg. Unreliable failure detectors for reliable distributed systems. *J. ACM*, 43(2):225–267, 1996.

7 Tyler Crain, Vincent Gramoli, Mikel Larrea, and Michel Raynal. (Leader / Randomization / Signature)-free Byzantine Consensus for Consortium Blockchains, 2017.

8 Christian Decker, Jochen Seidel, and Roger Wattenhofer. Bitcoin meets strong consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking, Singapore, January 4-7, 2016*, pages 13:1–13:10, 2016.

9 Carole Delporte-Gallet, Stéphane Devismes, Hugues Fauconnier, Franck Petit, and Sam Toueg. With finite memory consensus is easier than reliable broadcast. In *Principles of Distributed Systems, 12th International Conference, OPODIS 2008, Luxor, Egypt, December 15-18, 2008. Proceedings*, pages 41–57, 2008.

10 Cynthia Dwork, Nancy A. Lynch, and Larry J. Stockmeyer. Consensus in the presence of partial synchrony. *J. ACM*, 35(2):288–323, 1988.

11 Elli Androulaki *et al.* Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In *Proceedings of the Thirteenth EuroSys Conference, EuroSys 2018, Porto, Portugal, April 23-26, 2018*, pages 30:1–30:15, 2018.

12 Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014*, pages 436–454, 2014.

13 Ittay Eyal and Emin Gün Sirer. Majority is not enough: bitcoin mining is vulnerable. *Commun. ACM*, 61(7):95–102, 2018.

14 Giulia C. Fanti, Leonid Kogan, Sewoong Oh, Kathleen Ruan, Pramod Viswanath, and Gerui Wang. Compounding of wealth in proof-of-stake cryptocurrencies. In Ian Goldberg and Tyler Moore, editors, *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019*, pages 42–61, 2019.

15 M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM*, 32(2), April 1985.

16 Nissim Francez. *Fairness.* Texts and Monographs in Computer Science. Springer, 1986.

17 J. A. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Proc. of the EUROCRYPT International Conference*, 2015.

18 Guy Golan-Gueta, Ittai Abraham, Shelly Grossman, Dahlia Malkhi, Benny Pinkas, Michael K. Reiter, Dragos-Adrian Seredinschi, Orr Tamir, and Alin Tomescu. SBFT: A scalable and decentralized trust infrastructure. In *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019, Portland, OR, USA, June 24-27, 2019*, pages 568–580, 2019.

19 Fabíola Greve, Murilo Santos de Lima, Luciana Arantes, and Pierre Sens. A time-free byzantine failure detector for dynamic networks. In *2012 Ninth European Dependable Computing Conference, Sibiu, Romania, May 8-11, 2012*, pages 191–202, 2012.

20 Rachid Guerraoui and Jingjing Wang. On the unfairness of blockchain. In *Networked Systems - 6th International Conference, NETYS 2018, Essaouira, Morocco, May 9-11, 2018*, pages 36–50, 2018.

21 Önder Gürcan, Alejandro Ranchal Pedrosa, and Sara Tucci-Piergiovanni. On cancellation of transactions in bitcoin-like blockchains. In *On the Move to Meaningful Internet Systems. OTM 2018 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018*, pages 516–533, 2018.

22 Önder Gürcan, Antonella Del Pozzo, and Sara Tucci-Piergiovanni. On the bitcoin limitations to deliver fairness to users. In *On the Move to Meaningful Internet Systems. OTM 2017 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE 2017, Rhodes, Greece, October 23-27, 2017*, pages 589–606, 2017.

23 Maurice Herlihy and Mark Moir. Enhancing accountability and trust in distributed ledgers. *CoRR*, abs/1606.07490, 2016.

24 Dimitris Karakostas, Aggelos Kiayias, Christos Nasikas, and Dionysis Zindros. Cryptocurrency Egalitarianism: A Quantitative Approach. In *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019), Paris, France, May 06-07*, 2019.

25 Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 357–388, 2017.

26 Kim Potter Kihlstrom, Louise E. Moser, and P. M. Melliar-Smith. Byzantine fault detectors for solving consensus. *Comput. J.*, 46(1):16–35, 2003.

27 E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Security Symposium*, 2016.

28 Nicolas Lagaillardie, Mohamed Aimen Djari, and Önder Gürcan. A Computational Study on Fairness of the Tendermint Blockchain Protocol. *Information*, 10(12):378, 2019.

29 Kfir Lev-Ari, Alexander Spiegelman, Idit Keidar, and Dahlia Malkhi. Fairledger: A fair blockchain protocol for financial institutions. In *23rd International Conference on Principles of Distributed Systems, OPODIS 2019, December 17-19, 2019, Neuchâtel, Switzerland*, pages 4:1–4:17, 2019.

30 MATLAB. *version 9.6 (R2019a)*. The MathWorks Inc., Natick, Massachusetts, 2019.

31 S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. `https://bitcoin.org/bitcoin.pdf` (visited on 2019-08-15), 2008.

32 Dev Ojha and Christopher Goes. F1 Fee Distribution. In *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019), Paris, France, May 06-07*, 2019.

33 Rafael Pass and Elaine Shi. The sleepy model of consensus. In *ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, pages 380–409, 2017.

34 M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, April 1980.

35 Fahad Saleh. Blockchain Without Waste: Proof-of-Stake. SSRN Scholarly Paper ID 3183935, Social Science Research Network, Rochester, NY, January 2019.

36 Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan-Gueta, and Ittai Abraham. Hotstuff: BFT consensus with linearity and responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019.*, pages 347–356, 2019.