

# Do the Rich Get Richer?

## Fairness Analysis for Blockchain Incentives

Yuming Huang  
National University of Singapore  
huangyuming@u.nus.edu

Jing Tang\*

National University of Singapore  
isejtang@nus.edu.sg

Qianhao Cong  
National University of Singapore  
cong\_qianhao@u.nus.edu

Andrew Lim  
National University of Singapore  
isealim@nus.edu.sg

Jianliang Xu  
Hong Kong Baptist University  
xujl@comp.hkbu.edu.hk

### ABSTRACT

Proof-of-Work (PoW) is the most widely adopted incentive model in current blockchain systems, which unfortunately is energy inefficient. Proof-of-Stake (PoS) is then proposed to tackle the energy issue. The rich-get-richer concern of PoS has been heavily debated in the blockchain community. The debate is centered around the argument that whether rich miners possessing more stakes will obtain higher staking rewards and further increase their potential income in the future. In this paper, we define two types of fairness, i.e., *expectational* fairness and *robust* fairness, that are useful for answering this question. In particular, expectational fairness illustrates that the expected income of a miner is proportional to her initial investment, indicating that the expected return on investment is a constant. To better capture the uncertainty of mining outcomes, robust fairness is proposed to characterize whether the return on investment concentrates to a constant with high probability as time evolves. Our analysis shows that the classical PoW mechanism can always preserve both types of fairness as long as the mining game runs for a sufficiently long time. Furthermore, we observe that current PoS blockchains implement various incentive models and discuss three representatives, namely ML-PoS, SL-PoS and C-PoS. We find that (i) ML-PoS (e.g., Qtum and Blackcoin) preserves expectational fairness but may not achieve robust fairness, (ii) SL-PoS (e.g., NXT) does not protect any type of fairness, and (iii) C-PoS (e.g., Ethereum 2.0) outperforms ML-PoS in terms of robust fairness while still maintaining expectational fairness. Finally, massive experiments on real blockchain systems and extensive numerical simulations are performed to validate our analysis.

### ACM Reference Format:

Yuming Huang, Jing Tang, Qianhao Cong, Andrew Lim, and Jianliang Xu. 2021. Do the Rich Get Richer? Fairness Analysis for Blockchain Incentives. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21)*, June 20–25, 2021, Virtual Event, China. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3448016.3457285>

\*Corresponding author: Jing Tang.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGMOD '21, June 20–25, 2021, Virtual Event, China

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8343-1/21/06...\$15.00

<https://doi.org/10.1145/3448016.3457285>

### 1 INTRODUCTION

#### 1.1 Background

Since 2008, blockchain has attracted a plethora of interests from both academia and industry. Essentially, a blockchain is a decentralized public ledger that contains all current and historical updates in the system. Since blockchain can achieve community trust without third parties, it can be used in numerous real-world applications, such as cryptocurrency, smart contract, voting and bidding systems.

Particularly, Bitcoin [50] is the first and the most popular permission-less blockchain system, which stores user transactions within sequentially linked blocks. New transactions are confirmed when they are packed into newly generated blocks. The block generation process is called mining and a network node that potentially produces blocks is known as a miner. To generate new blocks, miners should find a solution to a cryptographic puzzle, referred to as Proof-of-Work (PoW) [35]. To incentivize the maintenance of the Bitcoin network, the miners who successfully contribute a valid proof will be rewarded. As a consequence, miners compete with one another by operating more computational devices to seek more incentives. Such a competition, unfortunately, incurs striking energy consumption. Presently, the total electricity consumption by Bitcoin miners amounts to 74 TWh, exceeding Austria (70 TWh), Switzerland (58 TWh) and Singapore (47 TWh) [2, 15]. For sustainable development, Proof-of-Stake (PoS) [74] is proposed as an alternative of PoW. Instead of competing for computation power, PoS miners are more likely to win if they possess more stakes (e.g., cryptocurrency). Therefore, the PoS protocol eliminates electricity consumption and significantly reduces waste of electricity.

However, the blockchain community raises concerns about the PoS protocol that it may make the rich become richer because rich miners possessing more stakes are likely to obtain more rewards and further increase their potential income in the future [11, 59]. Such a rich-get-richer phenomenon (also known as the Matthew effect) is obviously unfair to miners and eventually harms decentralization of the network when the majority of stakes are controlled by a few rich miners [9, 38, 42, 62]. In particular, resource accumulation may increase the risk of transactions rollback and data tampering, damaging to data reliability and integrity, e.g., double spending. Despite the young age of blockchain, resource accumulation has caused several severe accidents. For example, very recently in August 2020, the transactions in Ethereum Classic were rolled back because of a 51% attack, resulting in a loss of 1.68 million dollars [51]. To our knowledge, the rich-get-richer concern of the

PoS protocol has been scarcely studied though it has been heatedly debated. We aim to tackle this issue by leveraging the notion of *fairness*, which is one of the principles of fairness, accountability and transparency (FAT) of responsible data science [29].

## 1.2 Contributions

In this study, we focus on the fairness of incentive models for PoW and PoS protocols, which is the key to address the rich-get-richer concern of PoS. We formally define two types of fairness, i.e., *expectational* fairness and *robust* fairness. Specifically, expectational fairness characterizes the relation between the resource controlled by a miner and her expected reward. However, this naive definition might be insufficient to capture the uncertainty of reward in the real world. As an example, suppose that a miner initially possesses 20% of the entire stakes and consider two possible mining games. That is, the miner always receives 20% of the total rewards in the first game, whereas she wins all rewards with a probability of 20% and gets nothing with the remaining probability in the second game. Apparently, the expected rewards obtained from both games are exactly the same, i.e., both games are fair in expectation. However, the reward allocation in the second game is obviously more uncertain, thereby increasing the risk of her income. To reveal the uncertainty of reward, we further propose a novel concept of robust fairness that describes the relation between the actual rewards obtained by a miner from a random outcome with respect to the stochastic process of a mining game and her initial investment.

Based on our definitions of fairness, we conduct analysis on PoW and PoS incentives. In PoW, the selection of a block proposer is based on the hash power controlled by each miner, which is independent of previous mining outcomes. In particular, a miner proposes a block with a probability being proportional to the hash power controlled by her, and we find that this mechanism can ensure both types of fairness. Our analysis on the PoW incentive model is confirmed with experiments on Geth client (v1.9.11) [20].

In PoS, stakes serve as a competing resource and the probability of a miner proposing a new block is based on her current stakes which depend not only on her initial investment but also on the rewards received in previous mining outcomes. Moreover, there are various implementations of PoS incentives resulting in different types of behavior. In this paper, we analyze three representative PoS incentive models, namely multi-lottery PoS (ML-PoS) for Qtum [75] and Blackcoin [69], single-lottery PoS (SL-PoS) for NXT [72] and compound PoS (C-PoS) for Ethereum 2.0 [26], which cover the present popular blockchain systems.

Specifically, ML-PoS enables the probability of a miner proposing a new block being proportional to her current possessed stakes, which preserves expectational fairness. However, the return on investment of a miner under this protocol may not concentrate to a constant even after the mining game runs for a long time because of the accumulated effect of the Markov chain. Our experimental evaluations on real blockchain systems and numerical simulations reveal that allocating more initial stakes in the early stage of mining process and/or reducing the reward of each block are helpful to improve robust fairness. Meanwhile, SL-PoS is devised by leveraging a single-lottery scheme to determine a block proposer, unlike ML-PoS that uses a multi-lottery scheme. However, we find that SL-PoS can accomplish neither expectational fairness nor robust

fairness as rich miners have a higher return on investment. In fact, with the advantages accumulated during mining, the game will run to monopolization almost surely, incurring the Matthew effect. Recently, C-PoS introduces an additional inflation reward by distributing base incentives to every miner proportional to their possessed stakes. Our analysis shows that in addition to preserving expectational fairness, such an inflation reward is useful for reducing the uncertainty of mining income so that robust fairness is more likely to be achieved.

In summary, our main contributions are as follows.

- (1) We propose two types of fairness for blockchain incentives, i.e., expectational fairness and robust fairness, to characterize the relation between the resource controlled by a miner and her rewards obtained from a mining game.
- (2) We conduct a thorough theoretical analysis on fairness for the most widely adopted PoW incentive protocol and three popular PoS incentive protocols. In the context of fairness, these protocols are generally ranked in the following descending order: PoW, C-PoS, ML-PoS and SL-PoS.
- (3) We carry out extensive experiments on real blockchain systems and numerical simulations to evaluate the fairness of different incentive protocols. Experimental results confirm our theoretical findings and shed light on the future development of fair incentive protocols.

## 1.3 Organizations

Section 2 discusses the details of the PoW incentive protocol and three PoS incentive protocols. Section 3 analyzes expectational fairness and Section 4 studies robust fairness for the mentioned four incentive protocols. Section 5 carries out experimental evaluations and numerical simulations. Section 6 discusses the lessons we learned from fairness analysis. Section 7 reviews related work, and Section 8 concludes the paper.

## 2 INCENTIVE MODELS OF BLOCKCHAINS

Miners compete for proposing a valid block to append it to the current blockchain, which is incentivized by rewards. The chance of winning the competition usually depends on the resource controlled by miners, e.g., computation power and staking power. In this section, we introduce the incentive protocols of Proof-of-Work (PoW) and Proof-of-Stake (PoS) that are widely adopted by many popular blockchain systems.

### 2.1 PoW Incentive Model

A new block is accepted by a PoW network after a cryptographic puzzle is solved by miners [50]. Specifically, a PoW puzzle is to find a valid nonce such that  $\text{Hash}(\text{nonce}, \dots) < D$ ,<sup>1</sup> where  $D$  is a pre-defined mining difficulty. The  $\text{Hash}(\cdot)$  function maps a nonce to an integer uniformly distributed in the range  $[0, 2^{256} - 1]$ . As a result, the event that the hash value is less than  $D$  meets Bernoulli distribution with a success probability of  $\frac{D}{2^{256}}$ . While  $D$  is much less than  $2^{256}$  by design, it is almost impossible to solve the puzzle

<sup>1</sup> Apart from the argument nonce, the  $\text{Hash}(\cdot)$  function has some other arguments like the merkle root of packed transactions and the hash value of the previous block, e.g.,  $\text{Hash}(\text{nonce}, \text{merkle root}, \text{previous hash})$ .

in one single trial. The likelihood of a miner solving the puzzle depends on the number of nonce per unit time she can check. For instance, we assume that there are two miners  $A$  and  $B$  who can verify  $H_A$  and  $H_B$  hashes every day, respectively. Therefore, the number of valid blocks found by  $A$  (resp.  $B$ ) during one day is very well approximated to a Poisson random variable with mean  $\mu_A = \frac{DH_A}{2^{256}}$  (resp.  $\mu_B = \frac{DH_B}{2^{256}}$ ). Then, the probability that miner  $A$  (resp.  $B$ ) will win the next block can be computed as follows. Specifically, let  $T_A$  (resp.  $T_B$ ) be the arrival time of the next block for  $A$  (resp.  $B$ ). With respect to the Poisson process, it is well known that  $T_A$  and  $T_B$  follow negative exponential distributions with rate parameters  $\mu_A$  and  $\mu_B$  [63, Section 2.2], respectively, i.e., the probability density function of  $T_A$  is  $f(t_A; \mu_A) = \mu_A e^{-\mu_A t_A}$  for  $t_A \geq 0$ . If miner  $A$  wins the next block,  $A$  must find a valid block earlier than  $B$  such that  $T_A < T_B$ . Therefore, the probability of miner  $A$  winning the next block is

$$\begin{aligned} \Pr[T_A < T_B] &= \int_0^\infty \int_0^{t_B} \mu_A e^{-\mu_A t_A} \mu_B e^{-\mu_B t_B} dt_A dt_B \\ &= \frac{\mu_A}{\mu_A + \mu_B} = \frac{H_A}{H_A + H_B}. \end{aligned}$$

## 2.2 Multi-Lottery PoS Incentive Model

Despite of its popularity, PoW incurs massive energy consumption as the mining competition relies on computation power. To eradicate the waste of computation resource, PoS protocols are invented, where the competition depends on staking power instead. In the following, we introduce the multi-lottery PoS incentive model [69, 75], referred to as ML-PoS. Stakeholders of ML-PoS blockchains create a valid block if a candidate block satisfies the condition that  $\text{Hash}(\text{time}, \dots) < D \cdot \text{stake}$ , where  $\text{time}$  represents the timestamp when the candidate block is generated,  $D$  is a pre-determined mining difficulty and  $\text{stake}$  is the value of stakes possessed. Since  $\text{Hash}(\cdot)$  is uniformly distributed in the range  $[0, 2^{256} - 1]$ , the event that a candidate block becomes valid meets Bernoulli distribution with a success probability of  $\frac{D \cdot \text{stake}}{2^{256}}$ . Thus, if a miner possesses more stakes, she is more likely to create a new block successfully. Moreover, miners will try at different timestamps until a candidate block becomes valid. The trials are independent of timestamps following the same Bernoulli distribution. Again, we assume that there are two miners  $A$  and  $B$  possessing  $S_A$  and  $S_B$  stakes, respectively. We refer to  $T_A$  (resp.  $T_B$ ) as the number of timestamps miner  $A$  (resp.  $B$ ) has checked until  $A$  (resp.  $B$ ) meets the first success timestamp. It is easy to see that  $T_A$  and  $T_B$  follow geometric distributions with probability parameters  $p_A = \frac{DS_A}{2^{256}}$  and  $p_B = \frac{DS_B}{2^{256}}$ , respectively, i.e.,  $\Pr[T_A = t] = (1 - p_A)^{t-1} p_A$ . Furthermore, if  $A$  wins the next block,  $A$  finds a valid block with fewer timestamps than  $B$  such that  $T_A < T_B$  or  $A$  has a chance of 50% when  $T_A = T_B$  to break the tie. It is easy to get that

$$\begin{aligned} \Pr[T_A < T_B] &= \sum_{t_B=1}^\infty \sum_{t_A=1}^{t_B-1} (1 - p_A)^{t_A-1} p_A (1 - p_B)^{t_B-1} p_B \\ &= \frac{p_A - p_A p_B}{p_A + p_B - p_A p_B}. \end{aligned}$$

Similarly,  $\Pr[T_A = T_B] = \frac{p_A p_B}{p_A + p_B - p_A p_B}$ . Therefore, the probability of  $A$  winning the next block is

$$\Pr[T_A < T_B] + \frac{1}{2} \cdot \Pr[T_A = T_B] = \frac{p_A - p_A p_B / 2}{p_A + p_B - p_A p_B}.$$

Moreover, the time interval between two blocks is around 5–10 minutes by design. Thus,  $p_A$  and  $p_B$  are sufficiently small (e.g.,  $1/1200$ ), which indicates that  $p_A p_B$  is negligible. As a result,  $A$  wins the next block with a probability of  $\frac{p_A}{p_A + p_B} = \frac{S_A}{S_A + S_B}$ .

**Remark.** In ML-PoS, the  $\text{Hash}(\cdot)$  function depends on time (i.e., timestamp) instead of nonce as in PoW. Using time ensures that each miner has exactly one trial at each timestamp. Hence, the number of trials depends only on staking power so that the mining completion is independent of computation power. However, if nonce is applied, miners might try different nonces at each timestamp. As a result, the number of trials would rely on computation power as well.

## 2.3 Single-Lottery PoS Incentive Model

Another variant of PoS incentive model uses the single-lottery protocol [72], referred to as SL-PoS. Unlike ML-PoS where multiple trials are involved at different timestamps when miners compete for a block, SL-PoS only allows a single trial for each block. Specifically, each miner is assigned a lottery ticket represented by time, which is given by  $\text{time} = \text{basetime} \cdot \text{Hash}(\text{pk}, \dots) / \text{stake}$ , where  $\text{basetime}$  is a pre-determined constant,  $\text{pk}$  denotes the miner's public key, and  $\text{stake}$  refers to the miner's staking power. The protocol works as follows: (i) time determines when the candidate block will become valid, and (ii) the first valid block (i.e., the one with the smallest value of time) will be accepted by the current blockchain whereas the other candidates will be discarded. Again, the miner who possesses more stakes has a better chance to get a smaller value of time and hence she is more likely to be selected as the proposer of the next block. Consider the two-miner scenario where miner  $A$  and miner  $B$  control  $S_A$  and  $S_B$  stakes, respectively. Without loss of generality, we assume that  $S_A \leq S_B$ . Let  $T_A$  (resp.  $T_B$ ) denote the waiting time of  $A$ 's (resp.  $B$ 's) candidate block becoming valid, i.e.,  $T_A = \text{basetime} \cdot \text{Hash}(\text{pk}_A, \dots) / S_A$  where  $\text{pk}_A$  is  $A$ 's public key and  $\text{Hash}(\text{pk}_A, \dots)$  is a random number uniformly distributed in the range  $[0, 2^{256} - 1]$  with respect to  $\text{pk}_A$ . If  $A$  wins the next block, the waiting time of  $A$  should be smaller than that of  $B$  such that  $T_A < T_B$  or  $A$  has a chance of 50% when  $T_A = T_B$ . Therefore, the probability of miner  $A$  winning the next block is

$$\begin{aligned} \Pr[T_A < T_B] &+ \frac{1}{2} \cdot \Pr[T_A = T_B] \\ &= \sum_{h_B=0}^{2^{256}-1} \sum_{h_A=0}^{\frac{S_A h_B}{S_B}-1} \frac{1}{2^{256} \cdot 2} + \frac{1}{2} \sum_{h_B=0}^{2^{256}-1} \frac{1}{2^{256} \cdot 2} \\ &= \frac{S_A}{2S_B} \cdot \frac{2^{256} - 1}{2^{256}} + \frac{1}{2 \cdot 2^{256}} \approx \frac{S_A}{2S_B}, \end{aligned} \quad (1)$$

where  $h_A = \text{Hash}(\text{pk}_A, \dots)$  and  $h_B = \text{Hash}(\text{pk}_B, \dots)$ .

**Discussion.** In PoW, ML-PoS and SL-PoS, the reward is determined by the likelihood of a miner proposing a new block. However, unlike PoW and ML-PoS, we find that the success probability of a miner proposing a new block in SL-PoS is not proportional to her staking power in general. In particular, the above analysis shows that the probability of miner  $A$  winning the next block is  $\frac{S_A}{2S_B} < \frac{S_A}{S_A + S_B}$  when  $S_A < S_B$ , e.g.,  $\frac{S_A}{2S_B} \approx \frac{1}{2} \cdot \frac{S_A}{S_A + S_B}$  when  $S_A \ll S_B$ .

## 2.4 Compound PoS Incentive Model

Recently, a compound PoS incentive protocol, referred to as C-PoS, is deployed by the next generation Ethereum 2.0 [26]. Miners will be rewarded as (i) proposers who propose a new block and (ii) attesters who verify the validity of a block. Specifically, a miner will be assigned one identity for every 32 Ethers deposited in the smart contract. These identities are randomly and disjointly partitioned to 32 shards as attesters to verify transactions in parallel. In addition, one identity will be selected uniformly at random from every shard as the block proposer. During each mining epoch, a miner can receive  $3 \cdot \text{base} \cdot \text{vote}$  incentives for each attester identity she controls, where base is a pre-determined constant, vote is the percentage of attesters that stay online and actively submit votes (which is usually close to 100%). Meanwhile, a total of  $\frac{1}{8} \cdot \text{base} \cdot N$  incentives are provided for block proposers because of their contributions to their newly proposed blocks in each epoch, where  $N$  is the total number of identities assigned to all miners. Consider a two-miner scenario in a generalized C-PoS, where miners  $A$  and  $B$  possess  $S_A$  and  $S_B$  stakes, respectively. Let  $v$  and  $w$  denote the total rewards for attesters and proposers, respectively. Assume that there are  $P$  shards. Then, miner  $A$  will obtain  $v \cdot \frac{S_A}{S_A+S_B}$  stakes as attesters and  $w \cdot \frac{X}{P}$  stakes as proposers, where  $X$  is the number of blocks proposed by  $A$  in the epoch following binomial distribution  $\text{Bin}(P, \frac{S_A}{S_A+S_B})$ . Therefore, the total reward for  $A$  is  $v \cdot \frac{S_A}{S_A+S_B} + w \cdot \frac{X}{P}$ , where  $X \sim \text{Bin}(P, \frac{S_A}{S_A+S_B})$ .

**Remark.** To maintain the total reward for attesters (or proposers) in each epoch stable, Ethereum 2.0 slowly decreases the value of base as additional identities are rewarded after each epoch. In particular,  $v + w$  is around 1 Ether and  $v$  is  $\sim 20$  times of  $w$  in Ethereum 2.0 [26].

## 3 EXPECTATIONAL FAIRNESS

Fairness is one of the most important concerns for the design of incentive mechanisms. Intuitively, in a blockchain system with a fair incentive mechanism, the reward of a miner should be proportional to the amount of resource (e.g., computation power in PoW and staking power in PoS) that she obtains. That is, the return on investment is identical for every miner. In this section, we first introduce our assumptions and the definition of expectational fairness and then analyze such fairness for the aforementioned four incentive protocols, namely PoW, ML-PoS, SL-PoS and C-PoS.

### 3.1 Assumption and Definition

We consider permission-less blockchains for all the protocols analyzed and make the following assumptions.

- (1) Only two miners, i.e., miner  $A$  and miner  $B$ , are competing for proposing blocks in the network.
- (2) Initially, the resource share of miner  $A$  (resp.  $B$ ) is  $a$  (resp.  $b$ ). Without loss of generality,  $a$  and  $b$  are normalized such that  $a + b = 1$ .
- (3) The reward of each mining epoch remains the same, i.e.,  $w$  proposer reward and  $v$  inflation reward (e.g., attester reward in C-PoS).
- (4) Both miners  $A$  and  $B$  do not perform additional action after a mining game starts.

Without loss of generality, we focus on studying the relationship between the original mining share of miner  $A$  and the total rewards obtained by  $A$ . The simple two-miner model in Assumption 1 is for the sake of brevity. In Section 6.1, we elaborate how to generalize it to the scenario with multiple miners, e.g., considering  $B$  as a set of miners in PoW [22, 40]. In Assumption 2, the resource share represents the hash power ratio controlled by miners in PoW and the proportion of the initial amount of stakes possessed in PoS. As to Assumption 3, blockchain systems may change rewards when time evolves. For example, Bitcoin halves its block reward every 210,000 blocks. However, the current halving period of Bitcoin is around four years, which is a long time that can asymptotically support the assumption that the reward of each block remains unchanged [28, 49]. Also note that the values of  $w$  and  $v$  reflect the relative relation between an initial resource and a reward per mining epoch, since  $a$  and  $b$  are normalized. According to Assumption 4, we assume that both PoW and PoS miners passively participate in the mining game and do not perform any additional action like withdrawal or top-up [34, 39].

Our aim is to study whether a miner with a fraction of the total resource can finally obtain the same fraction of reward in expectation. To achieve this goal, we leverage the concept of expectational fairness which is formally defined as follows.

*Definition 3.1 (Expectational Fairness).* An incentive mechanism preserves expectational fairness for miner  $A$  possessing a fraction  $a$  of the total resource if  $A$  receives a fraction  $\lambda_A$  of the total reward satisfying  $\mathbb{E}[\lambda_A] = a$ .

### 3.2 Expectational Fairness for PoW

Initially, miner  $A$  (resp. miner  $B$ ) controls a fraction  $a$  (resp.  $b = 1 - a$ ) of the total hash power (i.e., mining rigs). The miner who successfully creates a new block will be rewarded an incentive  $w$ . Let  $\lambda_A$  represent the fraction of rewards received by miner  $A$  after a total of  $n$  blocks are appended to the blockchain. The total number  $n\lambda_A$  of blocks proposed by  $A$  follows a binomial distribution  $\text{Bin}(n, a)$ . Therefore, the expected reward of miner  $A$  is  $nwa$  and hence the expectation of the reward fraction is always equal to  $a$ . This indicates that PoW achieves expectational fairness, i.e., the expected reward of miner  $A$  is proportional to her initial computation power.

**THEOREM 3.2.** *PoW achieves expectational fairness.*

### 3.3 Expectational Fairness for ML-PoS

At the beginning, miner  $A$  (resp. miner  $B$ ) owns a fraction  $a$  (resp.  $b = 1 - a$ ) of the total stakes. Each block gives a reward of  $w$  stakes. Unlike PoW, the chance that  $A$  can win a block not only depends on the initial staking power (i.e.,  $a$ ) but also relies on previous mining outcomes. Specifically, the probability of  $A$  proposing a new block is determined by  $A$ 's current staking power, including the initial stakes and the earned stakes. For example, if a miner is "lucky" to mine some blocks, her expected rewards will be improved in the future as the volume of her stakes increases. In the following, we show that ML-PoS still preserves expectational fairness.

**THEOREM 3.3.** *ML-PoS achieves expectational fairness.*

Intuitively, the rationality behind Theorem 3.3 is from the fact that ML-PoS enables the probability of a miner proposing a new

block being proportional to her currently possessed stakes. That is, the conditional expected reward for miner  $A$  is proportional to her currently possessed stakes for each block. Taking expectation over the randomness of her possessed stakes, we can obtain that the expected reward for miner  $A$  is proportional to her expected possessed stakes for each block, which concludes the theorem. Due to space limitations, we omit all proofs, and interested readers are referred to the appendix in our technical report [6] for details.

### 3.4 Expectational Fairness for SL-PoS

As we discussed in Section 2.3, different from ML-PoS, the probability that  $A$  wins a block under SL-PoS is  $\frac{a}{2b} \leq a$  when  $a \leq b$ . As a result, unless  $a = b$ , the expected reward of  $A$  is not guaranteed to be proportional to her initial resource share  $a$  and hence SL-PoS does not preserve expectational fairness.

**THEOREM 3.4.** *SL-PoS does not ensure expectational fairness even after an infinity number of blocks are proposed.*

### 3.5 Expectational Fairness for C-PoS

Different from ML-PoS, in each mining epoch of C-PoS, the network randomly selects  $P$  block proposers and each proposer will receive a proposer reward of  $\frac{w}{P}$  (e.g.,  $P = 32$  in Ethereum 2.0 [26]). In addition, the system also provides a total of  $v$  inflation reward (e.g., attester reward in Ethereum 2.0 [26]) to all miners. Both the probability of proposer selection and the allocation of inflation reward are proportional to miners' present staking power. Therefore, analogous to ML-PoS, C-PoS still preserves expectational fairness although it is more complicated.

**THEOREM 3.5.** *C-PoS achieves expectational fairness.*

## 4 ROBUST FAIRNESS

In the previous section, we analyze the stochastic process of the mining procedure leveraging the concept of expectational fairness. However, people usually care more about fairness in every possible outcome rather than a simple expectation. Expectational fairness, unfortunately, cannot provide such analysis. To tackle this issue, in this section, we propose a new concept of robust fairness, which can better characterize the relation between the initial investment and the reward distribution.

### 4.1 Definition of Robust Fairness

Robust fairness, intuitively, implies that the random outcome of a miner's reward is concentrated to its initial investment with a high probability. To capture the concept of robust fairness, we define  $(\epsilon, \delta)$ -fairness as follows.

**Definition 4.1** ( $(\epsilon, \delta)$ -Fairness). For any given pair of parameters  $(\epsilon, \delta)$  such that  $\epsilon \geq 0$  and  $0 \leq \delta \leq 1$ , an incentive mechanism preserves an  $(\epsilon, \delta)$ -fairness for miner  $A$  possessing a fraction  $a$  of the total resource if  $A$  receives a fraction  $\lambda_A$  of the total reward satisfying

$$\Pr[(1 - \epsilon)a \leq \lambda_A \leq (1 + \epsilon)a] \geq 1 - \delta.$$

Definition 4.1 defines bicriteria fairness. Note that the definition of  $(\epsilon, \delta)$ -fairness does not explicitly include the total number  $n$

of blocks (or epochs for C-PoS) for competing.<sup>2</sup> Usually,  $\lambda_A$  will gradually converge as long as  $n$  increases. Thus, our analysis of  $(\epsilon, \delta)$ -fairness is carried out on a large value of  $n$ , including a special case where  $n$  goes to infinity. In the rest of the paper, we also say that an  $\epsilon$ -fairness is achieved with a probability at least  $1 - \delta$ , which exactly means that  $(\epsilon, \delta)$ -fairness is preserved. According to Definition 4.1, smaller values of  $\epsilon$  and/or  $\delta$  indicate a higher level of fairness. In particular, an incentive mechanism preserving  $(0, 0)$ -fairness, which is absolutely fair, is an ideal protocol.

### 4.2 Robust Fairness for PoW

For the PoW incentive protocol, let  $F(k; n, a)$  be the cumulative distribution function of the random variable  $n\lambda_A$ , i.e.,

$$F(k; n, a) := \Pr[n\lambda_A \leq k] = \sum_{i=0}^k \binom{n}{i} a^i (1-a)^{n-i}.$$

In addition, let  $\Delta(\epsilon; n, a)$  be a function of  $\epsilon, n, a$  such that

$$\Delta(\epsilon; n, a) := F(\lfloor n(1 + \epsilon)a \rfloor; n, a) - F(\lceil n(1 - \epsilon)a \rceil; n, a).$$

Thus,  $\Delta(\epsilon; n, a) = \Pr[(1 - \epsilon)a \leq \lambda_A \leq (1 + \epsilon)a]$ . Therefore, to achieve an  $(\epsilon, \delta)$ -fairness for miner  $A$ , the incentive parameters  $n$  and  $a$  must satisfy that  $\Delta(\epsilon; n, a) \geq 1 - \delta$ . However,  $\Delta(\epsilon; n, a)$  is complicated, which cannot clearly and explicitly reveal the requirements. To tackle this issue, we make use of Hoeffding inequality [33], which provides a *neat* expression of an upper bound on the probability that the sum of bounded independent random variables deviates from its expected value by more than a certain amount. In what follows, we give a sufficient condition (but not necessary) required by PoW for preserving an  $(\epsilon, \delta)$ -fairness.

**THEOREM 4.2.** *PoW preserves an  $(\epsilon, \delta)$ -fairness for miner  $A$  with the computation power of  $a$  if the total number  $n$  of blocks for competing satisfies  $n \geq \frac{\ln(\frac{2}{\delta})}{2a^2\epsilon^2}$ .*

By Theorem 4.2, we know that if miner  $A$  competes for more blocks and/or possesses more hash power, she will feel fairer. Moreover, it is easy to get that for PoW,  $\lambda_A$  converges to  $a$  almost surely when  $n \rightarrow \infty$ , i.e.,  $\Pr[\lim_{n \rightarrow \infty} \lambda_A = a] = 1$ . Thus, when  $n \rightarrow \infty$ , PoW is absolutely fair by achieving the  $(0, 0)$ -fairness.

### 4.3 Robust Fairness for ML-PoS

The mining process of ML-PoS can be modeled by a classical Pólya Urn such that the fraction  $\lambda_A$  of blocks proposed by  $A$  will finally converge to a beta distribution  $\text{Beta}(\frac{a}{w}, \frac{b}{w})$  almost surely [45, Theorem 3.2]. In particular, it is sufficient to achieve an  $(\epsilon, \delta)$ -fairness if  $I_{(1+\epsilon)a}(\frac{a}{w}, \frac{b}{w}) - I_{(1-\epsilon)a}(\frac{a}{w}, \frac{b}{w}) \geq 1 - \delta$ , where  $I$  is the regularized incomplete beta function. However, the relation is again not explicitly revealed. In the following, we derive a simple sufficient requirement by ML-PoS for achieving an  $(\epsilon, \delta)$ -fairness. Unlike PoW where the mining outcomes are independent and identically distributed random variables, the mining competition for ML-PoS is a Markov chain process. To tackle this issue, we leverage Azuma inequality [7] for martingales [18], which supports certain weakly dependent random variables.

<sup>2</sup>To reveal  $n$  explicitly, we may use  $\lambda_{A,n}$  to indicate the mining outcome of  $n$  blocks. For brevity,  $\lambda_A$  refers to  $\lambda_{A,n}$  in this paper unless specified otherwise.

**THEOREM 4.3.** *ML-PoS preserves an  $(\epsilon, \delta)$ -fairness for miner A with staking power of  $a$  if the total number  $n$  of blocks for competing and the block reward  $w$  satisfy  $1/n + w \leq \frac{2a^2\epsilon^2}{\ln \frac{2}{\delta}}$ .*

If miner A competes more blocks and/or possesses more staking power, it is easier for ML-PoS to achieve an  $(\epsilon, \delta)$ -fairness. However, unlike PoW, ML-PoS is sensitive to the reward  $w$ , e.g., a small reward for each block is more likely to be fair.

#### 4.4 Robust Fairness for SL-PoS

According to Section 2.3, for SL-PoS, the probability that miners win a block is not proportional to their staking powers. Specifically, the return on investment of a miner increases along with her staking power, which shows a clear unfairness of rich-get-richer. In the following, we study the robust fairness for SL-PoS by exploring the reward distribution.

Our analysis utilizes the techniques of Stochastic Approximation (SA) [60, 61]. We first introduce some useful definitions and lemmas of SA in the following.

**Definition 4.4 (Stochastic Approximation [60]).** A stochastic approximation algorithm  $\{Z_n\}$  is a stochastic process taking value in  $[0, 1]$ , adapted to the filtration  $\mathcal{F}_n$ , that satisfies,

$$Z_{n+1} - Z_n = \gamma_{n+1}(f(Z_n) + U_{n+1}),$$

where  $\gamma_n, U_n \in \mathcal{F}_n$ ,  $f: [0, 1] \mapsto \mathbb{R}$  and the following conditions hold almost surely

- (i)  $c_l/n \leq \gamma_n \leq c_u/n$ ,
- (ii)  $|U_n| \leq K_u$ ,
- (iii)  $|f(Z_n)| \leq K_f$ , and
- (iv)  $\mathbb{E}[\gamma_{n+1}U_{n+1} | \mathcal{F}_n] \leq K_e\gamma_n^2$ ,

where  $c_l, c_u, K_u, K_f, K_e$  are finite positive real numbers.

The stochastic approximation algorithm is originally used for root-finding problems. Specifically,  $\{Z_n\}$  is a stochastic process with an initial value of  $Z_0$ ,  $\gamma_n$  denotes a moving step size gradually decreasing along with  $n$  and  $\gamma_n U_n$  is a random noise with expectation tending to zero quickly. In a nutshell,  $Z_n$  moves towards one of the zero points of  $f(\cdot)$  and finally converges as long as the update process iterates a sufficiently large number of steps.

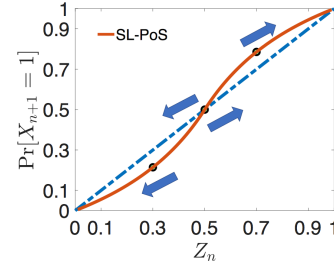
**LEMMA 4.5 (ZERO POINT OF SA [60]).** *If  $f$  is continuous then  $\lim_{n \rightarrow \infty} Z_n$  exists almost surely and is in  $Q_f = \{x: f(x) = 0\}$ .*

Note that  $Z_n$  may not converge to every zero point in  $Q_f$ . That is, if a zero point  $q$  is stable,  $Z_n$  converges to  $q$  when  $n \rightarrow \infty$  with a positive probability. Otherwise, if  $q$  is an unstable point,  $Z_n$  converges to  $q$  with zero probability. The following lemmas characterize the properties of stable and unstable points of SA.

**Definition 4.6 (Attainability [60]).** A subset  $I$  is attainable if for every fixed  $N \geq 0$ , there exists an  $n \geq N$  such that  $\Pr[Z_n \in I] > 0$ .

**LEMMA 4.7 (STABLE ZERO POINT OF SA [60]).** *Suppose  $q \in Q_f$  is stable, i.e.,  $f(x)(x - q) < 0$  whenever  $x \neq q$  is close to  $q$ . If every neighborhood of  $q$  is attainable then  $\Pr[Z_n \rightarrow q] > 0$ .*

**LEMMA 4.8 (UNSTABLE ZERO POINT OF SA [60]).** *Assume that there exists an unstable point  $q$  in  $Q_f$ , i.e., such that  $f(x)(x - q) \geq 0$  locally, and that  $\mathbb{E}[U_{n+1}^2 | \mathcal{F}_n] \geq K_L$  holds, for some  $K_L > 0$ , whenever  $Z_n$  is close to  $q$ . Then,  $\Pr[Z_n \rightarrow q] = 0$ .*



**Figure 1: Probability of winning the next block for SL-PoS.**

Now, we are ready to analyze the robust fairness for SL-PoS. Specifically, we denote  $Z_n$  as the fraction of staking power possessed by miner A after  $n$  blocks are competed, e.g.,  $Z_0 = a$ . We show that  $\{Z_n\}$  is a stochastic approximation algorithm. In particular, the update of  $Z_n$  is directed by the probability that the miner wins the next block characterized by  $f(\cdot)$ . Moreover, the update step size of  $Z_n$  also decreases along with  $n$  because the more stakes are issued during the mining game, the less one block outcome can affect  $Z_n$ . Then, we apply the stochastic approximation algorithm to study the asymptotic behavior of  $Z_n$ . Interestingly, we find that  $Z_n$  will finally converge to 0 or 1 almost surely, which indicates that SL-PoS cannot achieve robust fairness.

**THEOREM 4.9.** *For SL-PoS, the proportion reward  $\lambda_A$  of miner A converges to either 0 or 1 almost surely when  $n \rightarrow \infty$ . This indicates that SL-PoS cannot achieve robust fairness.*

Theorem 4.9 states that  $\lambda_A$  will finally converge to either  $\lambda_A = 0$  or  $\lambda_A = 1$ , no matter how much staking power is initially controlled by miner A. In other words, the mining game ends with the fact that one monopoly miner acquires almost 100% staking power. Figure 1 illustrates how the fraction of staking power evolves during the mining game. When  $Z_n = 0.3$ , the probability miner A can win another block is less than 30% thus her expected fraction of stakes decreases as the mining game proceeds. Finally,  $Z_n$  tends to 0. Vice versa, when  $Z_n = 0.7$ , the miner will win a block with a probability higher than 70%. As a consequence, her fraction of stakes tends to 1. Specially, when  $Z_n$  initiates on 0.5, the fraction of staking power possessed by the miner gradually leaves  $Z_n = 1/2$  towards either the left side or the right side and then converges to 0 or 1 with a fifty-fifty chance.

#### 4.5 Robust Fairness for C-PoS

C-PoS provides both inflation and proposer rewards. The income uncertainty comes from the proposer reward, which is reduced by the inflation reward that is distributed proportionally to miners' shares of stakes. Therefore, compared with ML-PoS, C-PoS is more likely to achieve robust fairness.

**THEOREM 4.10.** *C-PoS preserves an  $(\epsilon, \delta)$ -fairness for miner A with the staking power of  $a$  if the total number  $n$  of epochs for competing, the shard size  $P$  of each epoch, the proposer reward  $w$  and inflation reward  $v$  for each mining epoch satisfy  $\frac{w^2(1/n+w+v)}{(w+v)^2P} \leq \frac{2a^2\epsilon^2}{\ln \frac{2}{\delta}}$ .*

Theorem 4.10 states that C-PoS is more likely to achieve an  $(\epsilon, \delta)$ -fairness than ML-PoS because  $\frac{w^2(1/n+w+v)}{(w+v)^2P}$  is notably smaller than  $1/n + w$  due to the inflation reward  $v$  and  $P$  shards in each epoch.



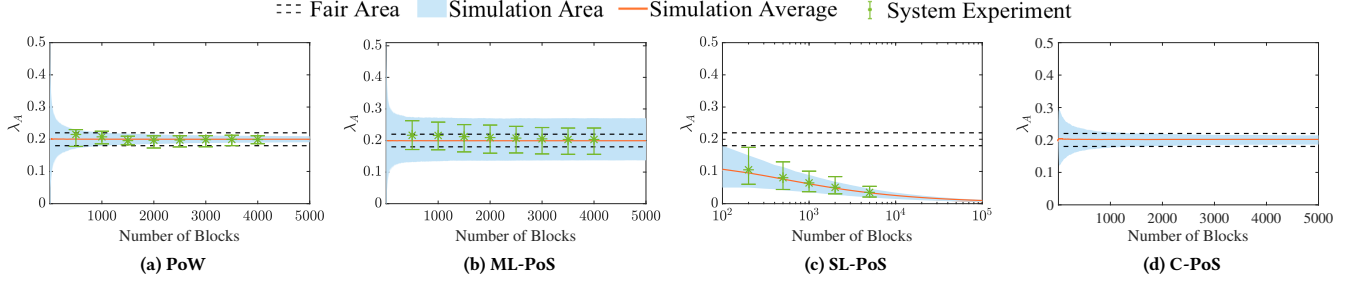


Figure 2: Evolution of  $\lambda_A$  along with the number  $n$  of blocks under  $a = 0.2$ ,  $w = 0.01$  and  $v = 0.1$ .

In particular, to achieve robust fairness for C-PoS, we can increase inflation reward  $v$  and shard size  $P$ , and meanwhile reduce proposer reward  $w$ . Note also that Theorem 4.10 degenerates to Theorem 4.3 if no inflation reward is provided and there is one shard in each epoch, i.e.,  $v = 0$  and  $P = 1$ .

## 5 EXPERIMENTAL EVALUATION

In this section, we evaluate the fairness for PoW and three PoS protocols (i.e., ML-PoS, SL-PoS and C-PoS) with both real system experiments and numerical simulations.

### 5.1 Experimental Setup

In the real system deployment, we select Geth client (v1.9.11) [20], Qtum core (v0.19.0.1) [58] and NXT client (v1.12.2) [36] as representatives of PoW, ML-PoS and SL-PoS mechanisms, respectively. Note that C-PoS is proposed by Ethereum 2.0 [26] which is still under development, due to which we cannot evaluate with real system experiments. All experiments are conducted on Amazon AWS EC2 Services. Specifically, PoW experiments are deployed on M5.4xlarge instances each with a 16-core Intel Xeon Platinum 8175M CPU and 32GB RAM. Since PoS protocols are computational insensitive, ML-PoS and SL-PoS experiments are conducted on M5.large instances each with a 2-core Intel Xeon Platinum 8175M CPU and 4GB RAM. In each test case, we implement a two-miner network and each miner is deployed on an individual EC2 instance. We repeat the experiments 10 times for PoW and 500 times for PoS, and report the statistical results.

Numerical simulations are also carried out to supplement the experiments especially when real system evaluations are limited in computational resource or mining time. In particular, we validate the fairness for C-PoS completely based on numerical simulations due to the lack of the real system. We repeat the simulations 10,000 times and report the statistical results.

For the evaluations of robust fairness, we set  $\varepsilon = 0.1$  and  $\delta = 10\%$  by default. That is, with a probability of at least 90%, the return on investment of a miner in a random outcome is in the range of  $[0.9, 1.1]$  of the average over all miners. For convenience, we refer to the range  $[(1 - \varepsilon)a, (1 + \varepsilon)a]$  as *fair area* and the range  $[0, (1 - \varepsilon)a) \cup ((1 + \varepsilon)a, 1]$  as *unfair area*. We measure the likelihood of  $\lambda_A$  locating in the fair area to reveal the robust fairness, i.e.,  $(\varepsilon, \delta)$ -fairness is achieved if such a likelihood is no less than  $1 - \delta$ .

### 5.2 Fairness Results

Figure 2 illustrates the evolution of  $\lambda_A$  with the number of blocks  $n$  that miners compete, which can capture both expectational fairness

and robust fairness. Initially, miner A controls  $a = 20\%$  of the total mining power for all the tested cases. For ML-PoS and SL-PoS, the reward of each block is set as  $w = 0.01$  (which is normalized against the total initial stakes). For C-PoS, the proposer and inflation rewards of each mining epoch are set as  $w = 0.01$  and  $v = 0.1$  where the inflation reward is 10 times of the proposer reward like the settings in Ethereum 2.0. In Figure 2, the area between two black dash lines is the fair area, i.e.,  $[(1 - \varepsilon)a, (1 + \varepsilon)a]$ . Meanwhile, the orange line represents the sample average of all the simulation results, and the bottom and top edges of the blue area indicate the 5th and 95th percentiles, respectively. That is, expectational fairness is achieved if the orange line matches the horizontal line with a value of 0.2, while robust fairness is achieved if the blue area locates within the fair area. Similarly, the green bar shows the result of the real system experiments, where the central mark indicates the mean, and the bottom and top edges indicate the 5th and 95th percentiles.

Figure 2(a) reports the evolution of  $\lambda_A$  for PoW. As can be seen, the average of both the experiment and the simulation is very close to 0.2, which confirms that PoW achieves expectational fairness as stated in Theorem 3.2. The result also shows that  $\lambda_A$  of both the system and the simulation gradually converges to the fair area as  $n$  increases. Specifically, when  $n < 100$ , there are a noticeable fraction of cases locating in the unfair area. On the other hand, when  $n > 1,000$ , almost all cases locate in the fair area, which confirms Theorem 4.2. In the Ethereum mining protocol, the average time interval between two blocks is around 15 seconds and thus the total reward obtained by a miner will concentrate to the expectation with a high probability after around 4.1 hours.

Figure 2(b) shows the result for ML-PoS. Again, the average of  $\lambda_A$  of both the experiment and the simulation is close to 0.2, which demonstrates the expectational fairness of ML-PoS (i.e., Theorem 3.3). However, unlike PoW, we find that there are a large number of cases locating in the unfair area even though miners have competed for a great number of blocks, e.g.,  $n = 5,000$  or equivalently 8.6 days in our experiment. That is, for certain settings of block reward  $w$ , e.g.,  $w = 0.01$ , miner A is likely to feel unfair no matter how long A participates the mining game, which is not robustly fair. Recall that Theorem 4.3 shows that if  $1/n + w \leq \frac{2a^2\varepsilon^2}{\ln(2/\delta)}$ , ML-PoS can achieve an  $(\varepsilon, \delta)$ -fairness, which is consistent with our observation. However, in our evaluation,  $\frac{2a^2\varepsilon^2}{\ln(2/\delta)} \approx 0.00027 \ll w = 0.01$ , which does not satisfy the requirement of robust fairness for ML-PoS.

Figure 2(c) plots the evolution of  $\lambda_A$  for SL-PoS. We observe that different from the three other protocols,  $\lambda_A$  continuously decreases as the mining game proceeds. Specifically, the average of  $\lambda_A$  for the

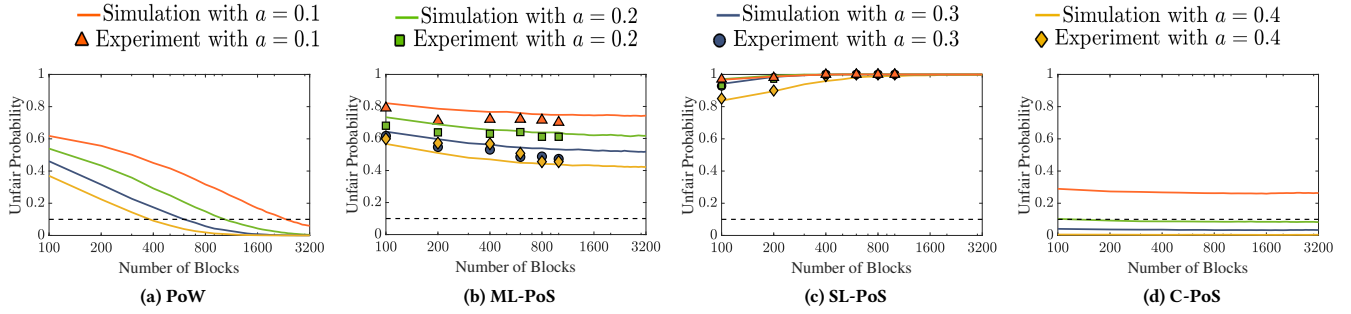


Figure 3: Unfair probabilities for PoW, ML-PoS, SL-PoS and C-PoS under  $w = 0.01$ ,  $v = 0.1$  and different settings of  $a$ .

first block is  $0.2/(2 \times 0.8) = 12.5\%$  (Section 2.3), and it decreases to 2% quickly after  $10^4$  new blocks are proposed (around 9.2 days on NXT). Furthermore,  $\lambda_A$  even approaches 0 when  $n$  reaches  $10^5$  (around 92 days on NXT). This indicates that when  $a = 0.2$  and  $w = 0.01$ , after the mining game operates for a period of time, the poor miner (i.e., miner A) will completely lose her stake share and the rich miner (i.e., miner B) will monopolize the future generation of new blocks. This phenomenon exhibits a clear unfairness. These observations confirm our analysis in Theorem 3.4 and Theorem 4.9.

Figure 2(d) gives the result for C-PoS where there are  $P = 32$  shards. We observe that the average of  $\lambda_A$ , as expected by Theorem 3.5, is almost 0.2. Compared with ML-PoS, the distribution of  $\lambda_A$  for C-PoS has a significantly narrower range. In fact, C-PoS is superior to ML-PoS, leveraging the advantages of sharding and inflation reward to reduce the uncertainty of reward allocation, as analyzed in Theorem 4.10. Therefore, we conclude that C-PoS is the best PoS protocol among the three tested in terms of fairness.

### 5.3 Study on Expectational Fairness for SL-PoS

In Figure 2, we show that among the four examined protocols, only SL-PoS does not ensure expectational fairness. In this section, we further study some factors, including initial stake allocation  $a$  and block reward  $w$ , that may affect the expectational fairness for SL-PoS. The result is given in Figure 4, where markers and lines represent the experimental and simulation results, respectively.

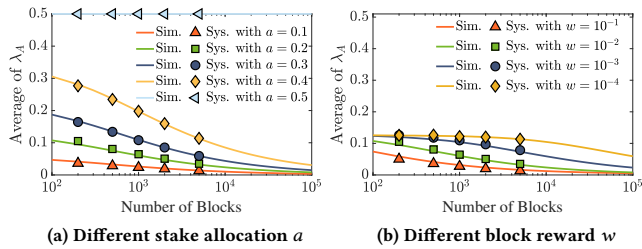


Figure 4: Average of reward proportion  $\lambda_A$  for SL-PoS.

**5.3.1 Impact of Initial Stake Allocation.** Figure 4(a) reports the average of the reward proportion  $\lambda_A$  under  $w = 0.01$  and different staking power  $a$  of miner A with values in  $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ . Interestingly, the average reward proportion of A reduces to 0 for all settings except the one at  $a = 0.5$ . This indicates that no matter how much staking power (once  $a < 0.5$ ) miner A controls initially, the miner will own zero staking power finally. We also observe that the average of  $\lambda_A$  increases along with  $a$ , which implies that it

takes longer time to completely lose competitiveness for the miner with a larger initial staking power.

**5.3.2 Impact of Block Reward.** Figure 4(b) shows the average of the reward proportion  $\lambda_A$  under  $a = 0.2$  and different block reward  $w$  with values in  $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ . We observe that the average of  $\lambda_A$  decreases along with both (i) the number  $n$  of blocks and (ii) block reward  $w$ . The reason is that if the block reward  $w$  is smaller, the fraction of staking power controlled by A reduces slower, so as to the average of  $\lambda_A$ .

### 5.4 Study on Robust Fairness

We further explore the effects of initial resource allocation  $a$ , block proposer reward  $w$  and inflation reward  $v$  on robust fairness. We measure the likelihood of  $\lambda_A$  locating in the unfair area, referred to as *unfair probability*, i.e.,  $\Pr[\lambda_A < (1 - \epsilon)a \vee \lambda_A > (1 + \epsilon)a]$ . This metric reveals that  $(\epsilon, \delta)$ -fairness is achieved only if the unfair probability is no more than  $\delta$ . Figure 3 and Figure 5 show the results, where the experimental and simulation results are indicated by markers and lines, respectively. Note that we only show experimental results for ML-PoS and SL-PoS, as repeating PoW experiments 10 times is insufficient to calculate unfair probability and C-PoS is under the development of Ethereum 2.0.

**5.4.1 Impact of Initial Resource Allocation.** Figure 3(a) shows the unfair probability for PoW under various  $a$ , where the black dash line represents the probability threshold  $\delta = 0.1$  for achieving the  $(\epsilon, \delta)$ -fairness. We observe that the unfair probability under all settings reduces along with the number  $n$  in general, which again implies that an  $(\epsilon, \delta)$ -fairness is always achievable by PoW when  $n$  is sufficiently large. We also find that PoW achieves an  $(\epsilon, \delta)$ -fairness faster for larger value of  $a$ . As an example, the number  $n$  of blocks required for preserving robust fairness for a medium miner with  $a = 0.3$  is less than 800 while that for a tiny miner with  $a = 0.1$  is more than 2,000. This indicates that if the majority of miners in a PoW-based blockchain system control a small fraction of the total hash power (which is the usual case), it requires a relatively long time period to preserve an  $(\epsilon, \delta)$ -fairness for every miner.

Figure 3(b) plots the unfair probability for ML-PoS. We observe that at the beginning, the unfair probability decreases along with the number  $n$  of blocks. However, when  $n$  reaches some thresholds, e.g.,  $n = 1,000$ , the unfair probability converges to certain constants that are likely to be larger than the threshold of  $\delta$ . This indicates that a long period of competing time does not suffice the requirement of  $(\epsilon, \delta)$ -fairness. Meanwhile, we observe that the unfair probability



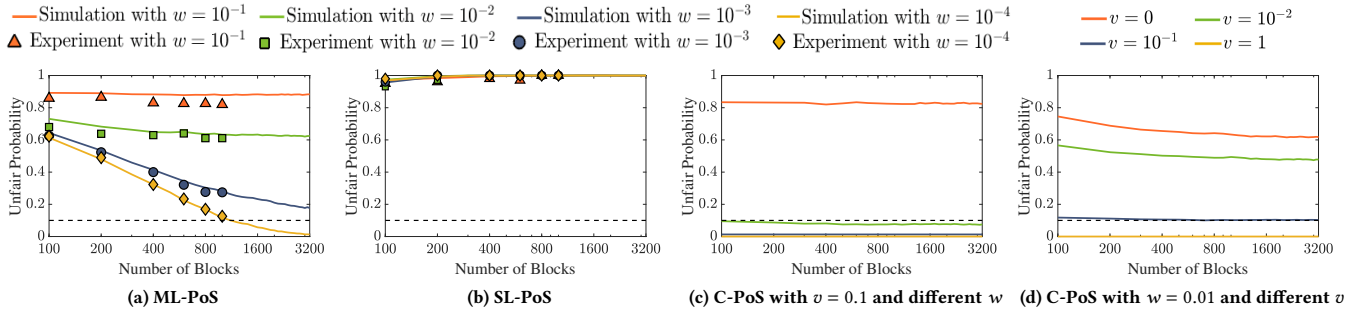


Figure 5: Unfair probabilities for PoW, ML-PoS, SL-PoS and C-PoS under  $a = 0.2$  and different settings of  $w$  and  $v$ .

is smaller for a miner controlling more stakes  $a$ . This implies that rich miners are more likely to feel fair than poor miners.

Figure 3(c) compares the unfair probability for SL-PoS. We observe that the unfair probability of miner  $A$  initiates over a wide range, i.e., a tiny miner with  $a = 0.1$  starts with an unfair probability of 98% and the unfair probability for a large miner with  $a = 0.4$  is 82%. However, in all settings, the unfair probability gradually increases with a larger block number and eventually converges to 100%. Moreover, the result shows that the unfair probability of rich miners deteriorates slower than that of poor miners. As an example, the unfair probability of a tiny miner with  $a = 0.1$  converges to 100% when  $n$  reaches 200 but a large miner with  $a = 0.4$  turns to 100% unfair when  $n$  exceeds 800.

Figure 3(d) reports the simulation result for C-PoS. In general, C-PoS has similar trends with ML-PoS but the unfair probability of the former is much lower and converges more rapidly. Specifically, for a medium miner with  $a = 0.3$ , the unfair probability of ML-PoS is as high as  $\sim 50\%$  but that of C-PoS is less than 10% such that the  $(\epsilon, \delta)$ -fairness is achieved for C-PoS but not for ML-PoS.

**5.4.2 Impact of Block Reward.** We further study how a block reward affects the robust fairness for ML-PoS, SL-PoS and C-PoS. (Note that PoW is insensitive to block reward  $w$ .) Figure 5(a) shows the unfair probability for ML-PoS under  $a = 0.2$  and different block reward settings. The setting of a large block with  $w = 0.1$ , where the block reward is close to the initial stake circulation, suffers from a severe fairness issue. In particular, the unfair probability of miner  $A$  is at least 85%. The reason is that the mining outcome of the first few blocks will significantly change the distribution of staking power among miners, which in turn heavily affects the mining game subsequently. As a contrary, in the setting of tiny block with  $w = 10^{-4}$ , achieving an  $(\epsilon, \delta)$ -fairness for miner  $A$  is easy. The reason is that if block reward is much smaller compared with the initial stakes, the earned stakes from the mining game have a negligible contribution to the staking power. In other words, the probability for a miner to propose a new block remains roughly unchanged when time evolves. Therefore, to improve the fairness for ML-PoS, we may set a small reward for each block or release more stakes at the very beginning of the mining game.

Figure 5(b) illustrates the unfair probability for SL-PoS, which is relatively insensitive to the block reward  $w$ . In particular, the unfair probabilities for SL-PoS initiate around 95% and then increase to 100% after 200 blocks for all the settings of  $w$  tested. Figure 5(c) reports the result for C-PoS when  $w$  varies. Again, C-PoS outperforms

ML-PoS significantly, though they have similar trends under different settings of  $w$ . Moreover, we also compare the unfair probability under different settings of the inflation reward  $v$  in Figure 5(d). As can be seen, the unfair probability decreases along with inflation reward  $v$ . Specifically, the unfair probability under  $v = 0$  is  $\sim 70\%$  whereas this value sharply reduces to  $\sim 50\%$  under  $v = 0.01$  and even to  $\sim 10\%$  under  $v = 0.1$ . In intuition, the inflation reward distributed to every miner is completely proportional to their staking power and hence the income uncertainty from the proposer reward is significantly diluted.

## 6 DISCUSSION

The previous analysis is based on a two-miner scenario, and this section first discusses how to extend our analysis to a general setting with multiple miners. In addition, the previous analysis reveals that SL-PoS can accomplish neither expectational fairness nor robust fairness while ML-PoS cannot easily achieve robust fairness. This section also discusses some remedies and improvements for these two protocols. Finally, we will discuss more incentives and some practicalities that can benefit from our fairness analysis.

### 6.1 Extension to Multiple Miners

Our analysis above is based on a simple two-miner scenario. In the following, we discuss the fairness in a general setting with multiple miners for the four aforementioned blockchain incentives.

In PoW, ML-PoS and C-PoS, using similar arguments in Section 2, it is trivial to get that the probability of a miner proposing a block is proportional to her computation/staking power regardless of the resource distribution of the other miners. Consequently, according to our analysis in Section 3 and Section 4, one can easily verify that the results of both expectational fairness and robust fairness still hold for the three incentives, by considering  $B$  as a set of miners.

On the other hand, such an extension for SL-PoS is non-trivial. In fact, we show that the probability of proposing a block is not proportional to the miner's staking power in general (unless all miners possess an identical amount of staking power).

**LEMMA 6.1.** *In SL-PoS with multiple miners, there exists a miner such that the probability of the miner proposing a block is not proportional to her staking power unless all miners possess an identical amount of staking power.*

By Lemma 6.1, analogous to our analysis in Section 3.4 and Section 4.4, we can get that neither expectational fairness nor robust fairness is accomplished by SL-PoS when there are multiple miners.

**Table 1: Results for Multi-Miner Game.**

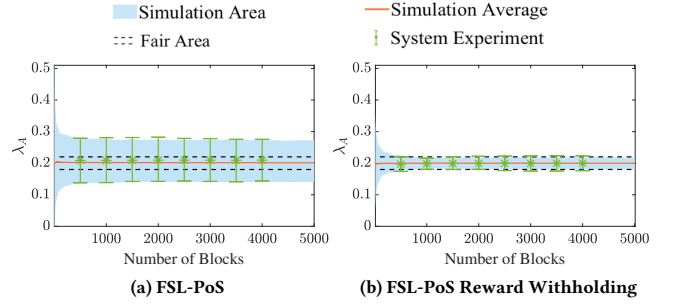
	No. of Miners	PoW	ML-PoS	SL-PoS	C-PoS
Avg. of $\lambda_A$	2 Miners	0.20	0.20	0.00	0.20
	3 Miners	0.20	0.20	0.00	0.20
	4 Miners	0.20	0.20	0.00	0.20
	5 Miners	0.20	0.20	0.20	0.20
	10 Miners	0.20	0.20	0.98	0.20
Unfair Prob.	2 Miners	0	0.14	1	0.08
	3 Miners	0	0.13	1	0.09
	4 Miners	0	0.14	1	0.08
	5 Miners	0	0.15	0.98	0.08
	10 Miners	0	0.13	1	0.08
Cvg. Time	2 Miners	1055	Never	Never	110
	3 Miners	1016	Never	Never	104
	4 Miners	1087	Never	Never	115
	5 Miners	1010	Never	Never	122
	10 Miners	1030	Never	Never	137

Table 1 shows the empirical results via simulations. We compare the average value of  $\lambda_A$  and the corresponding unfair probability. We also record the number of blocks when the  $(\epsilon, \delta)$ -fairness is achieved. In our simulation, miner  $A$  controls 20% of the initial mining resource and the other miners equally share the 80% remaining mining resource. By default,  $w = 0.01$  and  $v = 0.1$ . We can see that for PoW, ML-PoS and C-PoS, the results of multi-miner are similar to those of two-miner. This indicates that our analysis on these three protocols also holds when more than two players are included in the mining game. However, for SL-PoS, we observe that the average of  $\lambda_A$  remains 0 when 2–4 miners are considered in the mining game, and it suddenly increases to 0.2 and 0.98 under 5 and 10 miners, respectively. The result implies that miner  $A$ 's reward depends not only on her staking power but also on the staking distribution of  $A$ 's competitors. Specifically, only the biggest miner will monopolize the network with a high probability and the rest miners will finally lose their wealth, which extends the conclusion on two-miner to multi-miner. For example, when 5 miners compete in the network, all miners including miner  $A$  have identical 20% stakes initially. Thus, as discussed above, the average income of miner  $A$  should be 20% of the total reward. However, when more than 5 miners are included, miner  $A$  controls more stakes than the others and thus the average income of miner  $A$  improves dramatically.

## 6.2 Treatment for SL-PoS

SL-PoS will finally turn to monopolization due to the unfair winning probability for each block. One potential treatment is to adjust the time function so that the winning probability becomes asymptotically proportional to the staking power.

Specifically, assume that  $X$  and  $Y$  denote the hash values of candidate blocks issued by  $A$  and  $B$ , which are uniformly distributed in the range of  $[0, 2^{256} - 1]$  such that  $\frac{X}{2^{256}}$  and  $\frac{Y}{2^{256}}$  follow uniform distribution  $U(0, 1)$  asymptotically. To ensure both expectational and robust fairness, the time function  $T(\cdot)$  is required to satisfy  $\Pr[T(X; S_A) < T(Y; S_B)] = S_A / (S_A + S_B)$ , where  $S_A$  and  $S_B$  are



**Figure 6: Evolution of  $\lambda_A$  along with the number  $n$  of blocks under  $a = 0.2$ ,  $w = 0.01$ .**

the staking power of  $A$  and  $B$ . Motivated by the PoW incentive model in Section 2.1, if  $T_A = T(X; S_A)$  and  $T_B = T(Y; S_B)$  follow negative exponential distributions with rate parameters  $S_A$  and  $S_B$ , i.e., the probability density function of  $T_A$  is  $f(t_A; S_A) = S_A e^{-S_A t_A}$  for  $t_A \geq 0$ . To achieve this goal, we make use of inverse transform sampling. In particular, consider the cumulative distribution function of  $T_A$  as  $F(t_A; S_A) = 1 - e^{-S_A t_A}$ . Now, let  $F(T_A; S_A) = X$  such that  $T_A = \frac{-\ln(1-X)}{S_A}$ . We know that if  $X$  is uniformly distributed in the range of  $[0, 1]$ ,  $T_A$  is a random variable following the negative exponential distribution with a rate parameter  $S_A$ . Applying the same approach for  $T_B$ , we can obtain that  $\Pr[T_A < T_B] = \frac{S_A}{S_A + S_B}$ . To conclude, our treatment for SL-PoS is to set  $\text{time} = \text{basetime} \cdot \frac{-\ln(1-\text{Hash}(\text{pk}, \dots)/2^{256})}{\text{stake}}$ . We conduct both experiments and simulations on NXT to evaluate our treatment, referred to as FSL-PoS (i.e., fair-single-lottery PoS). Figure 6(a) shows the evolution of  $\lambda_A$  along with the number  $n$  of blocks that miners compete. In contrast to the original SL-PoS in Figure 2(c), the average income of miner  $A$  in FSL-PoS is 20% of the total reward, which confirms the expectational fairness of our treatment. However, we observe that quite a few cases of  $\lambda_A$  locate outside the fair area, which indicates that robust fairness is not achieved yet. In what follows, we further discuss how to improve robust fairness.

## 6.3 Improvement for Robust Fairness

**Reward Withholding.** One potential solution that may improve robust fairness is to withhold block rewards that will take effect periodically. As an example, the block reward will be issued to the proposer immediately but only take effect at the next 1,000-th block, e.g., the reward is issued at the 1,024-th block but takes effect at the 2,000-th block. The incentives obtained by a miner during two successive effective time points should concentrate to the expectation due to the law of large numbers. As a consequence, fairness will be improved. We perform experiments and simulations that apply reward withholding to FSL-PoS, where the reward takes effect at the block of the next thousand. Figure 6(b) reports the evolution of  $\lambda_A$  when time evolves. Clearly, almost all cases locate in the fair area, which demonstrates the effectiveness of our improvement.

**Less Block Reward.** As analyzed theoretically and empirically, a small block reward  $w$  is in favor of fairness for ML-PoS. However, this action should be carefully performed since less subsidy will reduce miners' motivation. In addition, increasing the initial circulation of stakes can indirectly reduce the relative block reward  $w$ ,

which will eventually benefit the fairness of ML-PoS. Initial Coin Offering (ICO) and airdrop are two common ways to allocate initial stake circulation. ICO allows a project team to sell a fraction of stakes to investors before the mining competition. Airdrop allocates cryptocurrencies towards community users for free during the early stage of mining.

#### 6.4 Fairness of More Incentive Protocols

The metrics and insights of fairness that we learned from PoW and three PoS protocols can be applied to more incentive protocols. In the following, we discuss six more incentives.

**NEO and NEO Gas.** NEO [73] is a PoS based blockchain that adopts a decentralized Byzantine fault tolerance consensus algorithm among authenticated stakers. The stakers compete for rewards depending on the share of *base asset* (e.g., NEO token) that they possess. Different from other PoS protocols, the rewards and transaction fees are paid for by a separate *reward asset* (e.g., NEO gas) that does not affect the future mining power. Therefore, such a PoS incentive works as same as the conventional PoW protocol, which preserves both types of fairness in a long-term mining game.

**Algorand.** Algorand [30] is a scalable blockchain adopting verifiable random functions and the Byzantine agreement. It just provides inflation rewards to the stakers who possess Algorand in wallet while no proposer reward is released in the mining process. As a result, the stakers will always obtain fair rewards without uncertainty. Despite its fairness, the incentive model has been questioned by criticisms, since consensus participants may lose motivation to maintain the ledger.

**EOS.** EOS [70] is a delegated PoS protocol based on the practical Byzantine fault tolerance. It achieves consensus among a committee with 21 elected delegates who propose blocks by turns. Every delegate proposes the same amount of blocks in a consensus round if she is active and honest. As for the incentive, each delegate receives an inflation reward proportional to her stakes (or votes) and a proposer reward which is a constant for everyone regardless of her stake share. Therefore, in general, neither expectational fairness nor robust fairness is achieved in EOS.

**Wave and Vixify.** Begicheva and Kofman [8] proposed a variant of PoS, called Wave, on the basis of NXT (i.e., SL-PoS), which improves the time function in NXT in a way similar to our treatment FSL-PoS. Orlicki [54] proposed Vixify by imitating the Nakamoto consensus in PoS leveraging verifiable random functions and verifiable delay functions. These two protocols ensure that a miner proposes a new block with a probability proportional to her stakes and only provide a proposer reward that will constitute future mining power. Therefore, analogous to FSL-PoS or ML-PoS, both Wave and Vixify can achieve expectational fairness but do not ensure robust fairness.

**Filecoin.** Filecoin [71] aims to build a storage network where clients upload and retrieve data in a decentralized way. The system utilizes a Proof-of-Storage-and-Time protocol to ensure the retrievability of stored data. The incentive is based on the miners' contributions on both storage space and pledge stakes, which constitute mining power. Hence, our analysis of PoW and PoS protocols is useful for understanding the fairness of the Filecoin incentive.

#### 6.5 Practicality of Fairness Analysis

**Protecting Data Reliability and Integrity.** Transaction processing and data provenance in permission-less blockchains rely on decentralized governance. A fair incentive is an essential component of public ledgers, since if a system is unfair by design, attackers or whale miners may easily accumulate their wealth during a mining game so that the network becomes centralized gradually. Monopoly miners can maliciously rollback transactions and tamper with data by concentrating mining power on launching a 51% attack. Recently, transactions in Ethereum Classic were rolled back due to the 51% attack, resulting in 1.68 million dollars of loss [51]. Therefore, as one of the FAT principles of responsible data science [29], fairness protects data reliability and integrity in practice.

**Preventing Mining Pools.** To reduce the uncertainty of reward, an effective strategy for miners is to join mining pools, which however encourages a centralized network and hence betrays the foundation of blockchain. Arguably, large mining pools are bad since they may concentrate power on launching severe attacks, e.g., 51% attack. This issue can be well addressed by leveraging the concept of robust fairness. In particular, an incentive preserving robust fairness ensures that miners receive stable rewards, i.e., the random outcome of a miner's reward is concentrated to its initial investment with high probability. With such an incentive mechanism, miners will lose motivation to join mining pools.

**Enhancing Security.** As discussed above, improving (expectational and robust) fairness can prevent miners from monopolizing the network or joining mining pools, which decreases the risk of adversarial control of a blockchain. In addition, there are several malicious attacks directly targeting on incentives so as to obtain an unfair profit, such as selfish mining [22, 40, 53], block withholding [21, 44] and bribery [27, 47]. Our analysis provides insight into further study of the incentive-based attacks, especially in PoS protocols which are rarely explored due to technical challenges. This will eventually be useful for developing secure blockchains resistant to these attacks.

### 7 RELATED WORK

**Incentive and Fairness.** To ensure data immutability and security of permission-less blockchains, a fair incentive mechanism is often required. To our knowledge, rare research work studied the fairness of blockchain incentives, though there have been some concerns raised by cryptocurrency communities. Fanti et al. [23] introduced the concept of *equitability* defined as the ratio of the incentive variance to the initial resource variance, which unfortunately cannot answer the fairness concern directly. Rosu and Saleh [64] analyzed the rich-get-richer phenomenon for ML-PoS using martingale and Dirichlet distribution, and claimed that ML-PoS will not face fairness issue as the fraction of rewards obtained by a miner in expectation is equal to her initial resource share, i.e., expectational fairness in our context. However, we introduce a new concept of robust fairness that can better capture the uncertainty of a mining game, and show that ML-PoS may not achieve robust fairness. Moreover, these studies [23, 64] focused on the classical ML-PoS protocol deployed on earlier PoS implementations such as Qtum [75] and Blackcoin [69], whereas our analysis, in addition to

ML-PoS, covers more state-of-the-art implementations including SL-PoS by NXT [72] and C-PoS by Ethereum 2.0 [26]. Pass and Shi [55] designed a fair protocol in similar spirit of Nakamoto’s PoW protocol. Different from their work that targeted at protocol design, we analyze the fairness of blockchain incentives for several popular protocols, including PoW, ML-PoS, SL-PoS and C-PoS.

The incentives of permission-less blockchains have attracted broader interests from researchers. The attacks on blockchain incentives may result in resource accumulation and further increase the risk of transaction tampering. Kwon et al. [41] discussed the movement of miners when mining rigs are applicable on two PoW networks. Tsabary and Eyal [76] and Carlsten et al. [12] found that miners may periodically suspend their mining rigs if no block reward is provided by the Bitcoin protocol. Some work studied the attacks on incentives, including selfish mining [22, 24, 40, 53, 67], block withholding [21, 44] and bribery [27, 47]. Our work is from the perspective of fairness that complements these existing studies on blockchain incentives.

**Transaction Processing.** Designing a transaction processing pipeline with high performance under large scale while ensuring security has been a major research topic [13]. Zakhary et al. [80] proposed an atomic cross-chain commitment for permission-less ledgers which ensures an all-or-nothing atomicity property. Herlihy et al. [32] extended such an atomicity to the cross-chain deal which can be applied to more types of adversarial commerce. Maiyya et al. [46] integrated fault tolerant replication into atomic commitment for cloud data management. Tao et al. [68] adopted a dynamic sharding algorithm on smart contracts to avoid empty blocks and waste of energy. Amiri et al. [3] adopted a directed acyclic graph on permissioned blockchain which supports both confidential transactions and cross-application transactions. In addition, some benchmark evaluations studied the transaction throughput and network latency of various blockchain systems [16, 17], the performance of blockchain index structures [79], and the performance of memory intensive PoW hash functions [25]. The security of transaction processing relies on the decentralization of the resource, which is heavily affected by the fairness of incentives, e.g., a 51% attack is likely to occur if the rich get richer. Our work evaluates the fairness of various commonly used incentives and provides insights into blockchain designs to ensure reliable data.

**Blockchain-as-a-Database.** Blockchain, as a distributed database, becomes popular for various applications. vChain [77, 78] and GEM<sup>2</sup> [82] applied an authenticated data structure to blockchain to ensure query integrity. Merkle<sup>ino</sup> and Chameleon<sup>ino</sup> [81] further reduced the maintenance cost of data authentication on hybrid-storage blockchains by leveraging cryptographic proof and chameleon commitment. ResilientDB [31] utilized a network-topology-aware consensus algorithm to achieve both lower communication latency and network decentralization. FalconDB [56] adopted database servers with verification interfaces accessible to clients and stored the digests for query/update authentications on a blockchain to enable efficient and secure collaboration. Buchnik and Friedman [10] proposed FireLedger, a new communication frugal optimistic permissioned blockchain protocol, to improve throughput. Abadi et al. [1] introduced AnyLog, a decentralized data sharing and publishing platform for IoT data. Ruan et al. [65] developed simple

interfaces that support smart contracts based provenance information. Qi et al. [57] improved storage scalability for blockchain systems by integrating erasure coding. Ruan et al. [66] enhanced the execute-order-validate architecture inspired by the optimistic concurrency control in modern databases. Nawab and Sadoghi [52] designed a middleware and communication infrastructure to ensure byzantine fault-tolerance in datacenter. Amiri et al. [4] leveraged a hybrid state machine replication protocol that avoids crash and malicious failures in cloud environment. Blockchain database usually requires incentives to attract participants, though it is not the main focus of the aforementioned studies. Our work provides insights into incentive designs to further expand the applicability of these blockchain databases.

**Polya Urn Process.** Our analysis utilizes some useful tools, including Azuma inequality [7] for martingales [18] and stochastic approximation [60, 61]. In particular, we use Doob’s martingale and Azuma’s inequality to derive the tail probability on concentration. Moreover, the mining process of PoS is related to the (nonlinear) generalized Pólya urn [5, 14, 43, 60]. For example, the mining process of ML-PoS can be modeled by a classical Pólya urn [45], where the fraction  $\lambda_A$  of blocks proposed by miner  $A$  will converge to a beta distribution almost surely. In addition, to solve the asymptotic convergence for the generalized Pólya urn, various methods are proposed, including stochastic approximation [37], brownian motion embedding [14] and exponential embedding [19, 48]. In this paper, we apply stochastic approximation to SL-PoS, which shows that  $\lambda_A$  will converge to either 1 or 0 no matter how much initial staking power miner  $A$  possesses. Using these approaches, our analysis may be extended to more complicated scenarios, e.g., with malicious attacks and games.

## 8 CONCLUSION

We study the fairness of incentives for several blockchain protocols, including PoW, ML-PoS, SL-PoS and C-PoS. We define two types of fairness, including expectational fairness and robust fairness. Our results show that all the protocols except SL-PoS can preserve expectational fairness. We also find that robust fairness is always achievable for PoW as long as the mining process runs for a sufficiently long time. Meanwhile, ML-PoS is difficult to achieve robust fairness while C-PoS can more easily achieve robust fairness thanks to inflation reward and sharding. Unfortunately, SL-PoS will finally turn to monopolization no matter how the initial stakes distribute, which never achieves robust fairness. Both real system experiments and numerical simulations are carried out to demonstrate our analysis. We provide some insights, e.g., increasing inflation reward and reducing proposer reward, to shed light on future study of blockchain incentives. For future work, we aim to take into account malicious attacks on incentives that can change reward distribution so that more fairness issues will be raised.

## ACKNOWLEDGMENTS

We are grateful to Michel van Kessel from Blackcoin, Wenbin Zhong from Qtum, Prysmatic Labs and NXT community for their technical supports. This research is supported by Singapore National Research Foundation under grant R-252-000-A27-490, and by HK-RGC GRF projects 12201520 and 12200819.

## REFERENCES

- [1] Daniel Abadi, Owen Arden, Faisal Nawab, and Moshe Shadmon. 2020. AnyLog: A Grand Unification of the Internet of Things. In *Proc. CIDR*.
- [2] Central Intelligence Agency. 2019. List of countries by electricity consumption. <https://www.cia.gov/library/publications/the-world-factbook/rankorder/2233rank.html>
- [3] Mohammad Javad Amiri, Divyakant Agrawal, and Amr El Abbadi. 2019. Caper: A Cross-Application Permissioned Blockchain. *Proc. VLDB Endowment* 12, 11 (2019), 1385–1398.
- [4] Mohammad Javad Amiri, Sujaya Maiyya, Divyakant Agrawal, and Amr El Abbadi. 2020. Seemore: A Fault-Tolerant Protocol for Hybrid Cloud Environments. In *Proc. IEEE ICDE*. 1345–1356.
- [5] W Brian Arthur, Yu M Ermoliev, and Yu M Kaniovski. 1987. Non-linear Urn Processes: Asymptotic Behavior and Applications.
- [6] Anonymous Author(s). 2020. Do the Rich Get Richer? Fairness Analysis for Blockchain Incentives (Missing Proof). <https://anonymous.4open.science/r/9b6f2851-dea4-4303-9649-e4e07f7cb8a9/>
- [7] Kazuoki Azuma. 1967. Weighted Sums of Certain Dependent Random Variables. *Tohoku Mathematical Journal, Second Series* 19, 3 (1967), 357–367.
- [8] A Begicheva and A Kofman. 2018. Fair Proof of Stake.
- [9] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A Kroll, and Edward W Felten. 2015. Sok: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In *Proc. IEEE S&P*. 104–121.
- [10] Yehonatan Buchnik and Roy Friedman. 2020. FireLedger: A High Throughput Blockchain Consensus Protocol. *Proc. VLDB Endowment* 13, 9 (2020), 1525–1539.
- [11] Buckkts. 2019. *The Rich Get Richer Concept in PoS*. <https://medium.com/peercoin/the-rich-get-richer-concept-in-proof-of-stake-systems-82c5eeff326>
- [12] Miles Carlsten, Harry Kalodner, S Matthew Weinberg, and Arvind Narayanan. 2016. On the Instability of Bitcoin Without the Block Reward. In *Proc. ACM CCS*. 154–167.
- [13] Sara Cohen, Adam Rosenthal, and Aviv Zohar. 2020. Reasoning about the Future in Blockchain Databases. In *Proc. IEEE ICDE*. 1930–1933.
- [14] Andrea Collevocchio, Codina Cotar, and Marco LiCalzi. 2013. On a Preferential Attachment and Generalized Pólya’s Urn Model. *The Annals of Applied Probability* 23, 3 (2013), 1219–1253.
- [15] Digiconomist. 2019. *Bitcoin Energy Consumption Index*. <https://digiconomist.net/bitcoin-energy-consumption>
- [16] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. 2018. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering* 30, 7 (2018), 1366–1385.
- [17] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. 2017. Blockbench: A Framework for Analyzing Private Blockchains. In *Proc. ACM SIGMOD*. 1085–1100.
- [18] Joseph Leo Doob. 1953. *Stochastic Processes*. Vol. 101. New York Wiley.
- [19] Eleni Drinea, Alan Frieze, and Michael Mitzenmacher. 2002. Balls and Bins Models with Feedback. In *Proc. SODA*. 308–315.
- [20] Ethereum. 2020. Geth v1.9.11. <https://github.com/ethereum/go-ethereum>
- [21] Ittay Eyal. 2015. The Miner’s Dilemma. In *Proc. IEEE S&P*. 89–103.
- [22] Ittay Eyal and Emin Gün Sirer. 2018. Majority Is Not Enough: Bitcoin Mining Is Vulnerable. *Commun. ACM* 61, 7 (2018), 95–102.
- [23] Giulia Fanti, Leonid Kogan, Sewoong Oh, Kathleen Ruan, Pramod Viswanath, and Gerui Wang. 2019. Compounding of Wealth in Proof-of-Stake Cryptocurrencies. In *Proc. FC*. 42–61.
- [24] Chen Feng and Jianyu Niu. 2019. Selfish Mining in Ethereum. In *Proc. IEEE ICDCS*. 1306–1316.
- [25] Zonghao Feng and Qiong Luo. 2020. Evaluating Memory-Hard Proof-of-Work Algorithms on Three Processors. *Proc. VLDB Endowment* 13, 6 (2020), 898–911.
- [26] Ethereum Foundation. 2020. Github: Ethereum 2.0 Specifications. <https://github.com/ethereum/eth2.0-specs>
- [27] Shang Gao, Zecheng Li, Zhe Peng, and Bin Xiao. 2019. Power Adjusting and Bribery Racing: Novel Mining Attacks in the Bitcoin System. In *Proc. ACM CCS*. 833–850.
- [28] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The Bitcoin Backbone Protocol: Analysis and Applications. In *Proc. EUROCRYPT*. 281–310.
- [29] Lise Getoor. 2019. Responsible Data Science. SIGMOD Keynote.
- [30] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nikolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proc. ACM SOSP*. 51–68.
- [31] Suyash Gupta, Sajjad Rahnama, Jelle Hellings, and Mohammad Sadoghi. 2020. ResilientDB: Global Scale Resilient Blockchain Fabric. *Proc. VLDB Endowment* 13, 6 (2020), 868–883.
- [32] Maurice Herlihy, Barbara Liskov, and Liuba Shrira. 2019. Cross-chain Deals and Adversarial Commerce. *Proc. VLDB Endowment* 13, 2 (2019), 100–113.
- [33] Wassily Hoeffding. 1994. Probability Inequalities for Sums of Bounded Random Variables. In *The Collected Works of Wassily Hoeffding*. Springer, 409–426.
- [34] Ciamac Gur Huberman, Jacob D Leshno, and C Moallemi. 2017. Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System.
- [35] Markus Jakobsson and Ari Juels. 1999. Proofs of Work and Bread Pudding Protocols. In *Secure Information Networks*. 258–272.
- [36] Jelurida. 2020. NXT Evaluation Toolkit v1.12.2. <https://bitbucket.org/Jelurida/nxt-clone-starter/src/master/>
- [37] Yu Kaniovski and Georg Pflug. 1995. Non-standard Limit Theorems for Urn Models and Stochastic Approximation Procedures. *Comm. in Statistics* 11, 1 (1995), 79–102.
- [38] Lucianna Kiffer and Rajmohan Rajaraman. 2018. A Better Method to Analyze Blockchain Consistency. In *Proc. ACM CCS*. 729–744.
- [39] Joshua A Kroll, Ian C Davey, and Edward W Felten. 2013. The Economics of Bitcoin Mining, or Bitcoin in the Presence of Adversaries. In *Proc. WEIS*.
- [40] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. 2017. Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin. In *Proc. ACM CCS*. 195–209.
- [41] Yujin Kwon, Hyoungshick Kim, Jinwo Shin, and Yongdae Kim. 2019. Bitcoin vs. Bitcoin Cash: Coexistence or Downfall of Bitcoin Cash? *Proc. IEEE S&P* (2019), 935–951.
- [42] Yujin Kwon, Jian Liu, Minjeong Kim, Dawn Song, and Yongdae Kim. 2019. Impossibility of Full Decentralization in Permissionless Blockchains. In *Proc. ACM AFT*. 110–123.
- [43] Sophie Laruelle and Gilles Pagès. 2019. Nonlinear Randomized Urn Models: A Stochastic Approximation Viewpoint. *Electronic Journal of Probability* 24, 98 (2019), 1–47.
- [44] Loi Luu, Ratul Saha, Inian Parameshwaran, Prateek Saxena, and Aquinas Hobor. 2015. On Power Splitting Games in Distributed Computation: The Case of Bitcoin Pooled Mining. In *Proc. IEEE CSF*. 397–411.
- [45] Hosam Mahmoud. 2008. *Pólya Urn Models*. CRC press.
- [46] Sujaya Maiyya, Faisal Nawab, Divyakant Agrawal, and Amr El Abbadi. 2019. Unifying Consensus and Atomic Commitment for Effective Cloud Data Management. *Proc. VLDB Endowment* 12, 5 (2019), 611–623.
- [47] Patrick McCorry, Alexander Hicks, and Sarah Meiklejohn. 2018. Smart Contracts for Bribing Miners. In *Proc. FC*. 3–18.
- [48] Michael Mitzenmacher, Roberto Oliveira, and Joel Spencer. 2004. A Scaling Result for Explosive Processes. *Electronic Journal of Combinatorics* 11, 1 (2004), 1–14.
- [49] Malte Möser and Rainer Böhm. 2015. Trends, Tips, Tolls: A Longitudinal Study of Bitcoin Transaction Fees. In *Proc. FC*. 19–33.
- [50] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System.
- [51] Nasdaq. 2020. Ethereum Classic Attacker Successfully Double-Spends 1.68M Dollars in Second Attack: Report. <https://www.nasdaq.com/articles/ethereum-classic-attacker-successfully-double-spends-%241.68m-in-second-attack%3A-report-2020>
- [52] Faisal Nawab and Mohammad Sadoghi. 2019. Blockplane: A Global-Scale Byzantizing Middleware. In *Proc. IEEE ICDE*. 124–135.
- [53] Kartik Nayak, Srikanth Kumar, Andrew Miller, and Elaine Shi. 2016. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. In *Proc. IEEE EuroS&P*. 305–320.
- [54] José L Orlicki. 2020. Sequential Proof-of-Work for Fair Staking and Distributed Randomness Beacons. arXiv preprint arXiv:2008.10189.
- [55] Rafael Pass and Elaine Shi. 2017. Fruitchains: A Fair Blockchain. In *Proc. ACM PODC*. 315–324.
- [56] Yanqing Peng, Min Du, Feifei Li, Raymond Cheng, and Dawn Song. 2020. FalconDB: Blockchain-based Collaborative Database. In *Proc. ACM SIGMOD*. 637–652.
- [57] Xiaodong Qi, Zhao Zhang, Cheqing Jin, and Aoying Zhou. 2020. BFT-Store: Storage Partition for Permissioned Blockchain via Erasure Coding. In *Proc. IEEE ICDE*. 1926–1929.
- [58] Qtum. 2020. Qtum Core v0.19.0.1. <https://github.com/qtumproject/qtum>
- [59] Gert Rammelo. 2019. The Economics of the Proof of Stake Consensus.
- [60] Henrik Renlund. 2010. Generalized Pólya Urns via Stochastic Approximation. arXiv preprint arXiv:1002.3716.
- [61] Herbert Robbins and Sutton Monro. 1951. A Stochastic Approximation Method. *The Annals of Mathematical Statistics* 22, 3 (1951), 400–407.
- [62] Meni Rosenfeld. 2014. Analysis of Hashrate-Based Double Spending. arXiv preprint arXiv:1402.2009.
- [63] Sheldon M Ross, John J Kelly, Roger J Sullivan, William James Perry, Donald Mercer, Ruth M Davis, Thomas Dell Washburn, Earl V Sager, Joseph B Boyce, and Vincent L Bristow. 1996. *Stochastic Processes*. Vol. 2. Wiley New York.
- [64] Ioanid Rosu and Fahad Saleh. 2019. Evolution of Shares in a Proof-of-Stake Cryptocurrency. SSRN 3377136.
- [65] Pingcheng Ruan, Gang Chen, Tien Tuan Anh Dinh, Qian Lin, Beng Chin Ooi, and Meihui Zhang. 2019. Fine-Grained, Secure and Efficient Data Provenance on Blockchain Systems. *Proc. VLDB Endowment* 12, 9 (2019), 975–988.
- [66] Pingcheng Ruan, Dumitrel Loghin, Quang-Trung Ta, Meihui Zhang, Gang Chen, and Beng Chin Ooi. 2020. A Transactional Perspective on Execute-order-validate Blockchains. In *Proc. ACM SIGMOD*. 543–557.
- [67] Ayelet Sapirshstein, Yonatan Sompolsky, and Aviv Zohar. 2016. Optimal Selfish Mining Strategies in Bitcoin. In *Proc. FC*. 515–532.

- [68] Yuechen Tao, Bo Li, Jingjie Jiang, Hok Chu Ng, Cong Wang, and Baochun Li. 2020. On Sharding Open Blockchains with Smart Contracts. In *Proc. IEEE ICDE*. 1357–1368.
- [69] Blackcoin Team. 2020. *Blackcoin Cryptocurrency*. <https://blackcoin.org/>
- [70] EOS Team. 2020. *EOS Cryptocurrency whitepaper*. <https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md>
- [71] Filecoin Team. 2020. *Filecoin Cryptocurrency whitepaper*. <https://filecoin.io/filecoin.pdf>
- [72] Jelurida Team. 2020. *NXT Cryptocurrency*. <https://www.jelurida.com/nxt>
- [73] Neo Team. 2020. *Neo Cryptocurrency whitepaper*. <https://docs.neo.org/docs/en-us/basic/whitepaper.html>
- [74] Peercoin Team. 2020. *Peercoin Cryptocurrency*. <https://www.peercoin.net/>
- [75] Qtum Team. 2020. *Qtum Cryptocurrency*. <https://qtum.org>
- [76] Itay Tsabary and Ittay Eyal. 2018. The Gap Game. In *Proc. ACM CCS*. 713–728.
- [77] Haixin Wang, Cheng Xu, Ce Zhang, and Jianliang Xu. 2020. vChain: A Blockchain System Ensuring Query Integrity. In *Proc. ACM SIGMOD*. 2693–2696.
- [78] Cheng Xu, Ce Zhang, and Jianliang Xu. 2019. vChain: Enabling Verifiable Boolean Range Queries over Blockchain Databases. In *Proc. ACM SIGMOD*. 141–158.
- [79] Cong Yue, Zhongle Xie, Meihui Zhang, Gang Chen, Beng Chin Ooi, Sheng Wang, and Xiaokui Xiao. 2020. Analysis of Indexing Structures for Immutable Data. In *Proc. ACM SIGMOD*. 925–935.
- [80] Victor Zakhary, Divyakant Agrawal, and Amr El Abbadi. 2019. Atomic Commitment Across Blockchains. *Proc. VLDB Endowment* 13, 9 (2019), 1319–1331.
- [81] Ce Zhang, Cheng Xu, Haixin Wang, Jianliang Xu, and Byron Choi. 2021. Authenticated Keyword Search in Scalable Hybrid-Storage Blockchains. In *Proc. IEEE ICDE*.
- [82] Ce Zhang, Cheng Xu, Jianliang Xu, Yuzhe Tang, and Byron Choi. 2019. GEM<sup>2</sup>-Tree: A Gas-Efficient Structure for Authenticated Range Queries in Blockchain. In *Proc. IEEE ICDE*. 842–853.

## A MISSING PROOFS

PROOF OF THEOREM 3.3. Let  $X_i \in \{0, 1\}$  be a binary random variable indicating whether  $A$  is the proposer for the  $i$ -th block. Let  $S_i$  be the total staking power possessed by  $A$  after  $i$  blocks, e.g.,  $S_0 = a$ . Then, it is easy to know that  $X_i$  follows Bernoulli distribution with success probability  $\frac{S_{i-1}}{1+w(i-1)}$ , as the total staking power of all miners for competing the  $i$ -th block is  $1 + w(i-1)$ . Thus, we have

$$S_{i+1} = S_i + wX_{i+1}.$$

Taking expectation conditioned on  $S_i$  gives

$$\mathbb{E}[S_{i+1} | S_i] = S_i + \frac{wS_i}{1+wi}.$$

As a result, we have

$$\mathbb{E}[S_{i+1}] = \mathbb{E}[\mathbb{E}[S_{i+1} | S_i]] = \frac{1+w(i+1)}{1+wi} \cdot \mathbb{E}[S_i].$$

Recursively, we can get that

$$\mathbb{E}[S_i] = S_0 \prod_{k=0}^{i-1} \frac{1+w(k+1)}{1+wk} = a(1+wi).$$

Therefore,  $\mathbb{E}[\lambda_A] = \frac{\mathbb{E}[S_n]-a}{wn} = a$ , which concludes the theorem.  $\square$

PROOF OF THEOREM 3.4. Again, let  $X_i \in \{0, 1\}$  be a binary random variable indicating whether  $A$  is the proposer for the  $i$ -th block. Consider that  $a \leq b$  such that  $\Pr[X_1 = 0] = 1 - \frac{a}{2b}$  and  $\Pr[X_1 = 1] = \frac{a}{2b}$ . Thus,  $\mathbb{E}[X_1] = \frac{a}{2b} < a$  unless  $a = b$ , which shows that the expected reward of the first block for miner  $A$  is unfair in general.

Next, we show that even an infinity number of blocks are proposed by  $A$  and  $B$ , there exists some  $a$  such that  $\mathbb{E}[\lambda_A] \neq a$ , where  $\lambda_A = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n X_i$ . We prove it by contradiction and assume that  $\mathbb{E}[\lambda_A] = a$  for all  $a$ . We observe that

$$\mathbb{E}[n\lambda_A] = \mathbb{E}[n\lambda_A | X_1 = 0] \Pr[X_1 = 0] + \mathbb{E}[n\lambda_A | X_1 = 1] \Pr[X_1 = 1].$$

By our assumption,  $\mathbb{E}[n\lambda_A - X_1 | X_1] = (n-1) \cdot \frac{a+wX_1}{1+w}$ , since miner  $A$  possesses a fraction  $\frac{a+wX_1}{1+w}$  of total staking power after the outcome of the first block is observed. Thus,

$$\mathbb{E}[n\lambda_A] = \frac{a(n-1)}{1+w} \cdot \Pr[X_1 = 0] + \left(1 + \frac{(a+w)(n-1)}{1+w}\right) \cdot \Pr[X_1 = 1].$$

As a result,

$$na = \frac{a(n-1)}{1+w} \cdot \left(1 - \frac{a}{2b}\right) + \left(1 + \frac{(a+w)(n-1)}{1+w}\right) \cdot \frac{a}{2b}.$$

Rearranging it yields  $a(b-a)(nw+1) = 0$ . This shows a contradiction when  $0 < a < b$  and hence the theorem is proved.  $\square$

PROOF OF THEOREM 3.5. The proof is analogous to that of Theorem 3.3. Let  $Y_i$  be the random variable representing the number of shard proposers assigned to miner  $A$  at epoch  $i$ . Let  $S_i$  be the total staking power possessed by  $A$  after epoch  $i$ , e.g.,  $S_0 = a$ . Then, it is easy to know that  $Y_i \sim \text{Bin}\left(P, \frac{S_{i-1}}{1+(w+v)(i-1)}\right)$ , as the total staking power of all miners at the beginning of epoch  $i$  is  $1 + (w+v)(i-1)$ . Thus,

$$S_{i+1} = S_i + \frac{wY_{i+1}}{P} + \frac{vS_i}{1+(w+v)i}.$$

Taking expectation conditioned on  $S_i$  gives

$$\mathbb{E}[S_{i+1} | S_i] = S_i + \frac{wS_i}{1+(w+v)i} + \frac{vS_i}{1+(w+v)i}.$$

As a result, we have

$$\mathbb{E}[S_{i+1}] = \mathbb{E}[\mathbb{E}[S_{i+1} | S_i]] = \frac{1+(w+v)(i+1)}{1+(w+v)i} \cdot \mathbb{E}[S_i].$$

Recursively, we can get that

$$\mathbb{E}[S_i] = S_0 \prod_{k=0}^{i-1} \frac{1+(w+v)(k+1)}{1+(w+v)k} = a(1+(w+v)i).$$

Therefore,  $\mathbb{E}[\lambda_A] = \frac{\mathbb{E}[S_n]-a}{(w+v)n} = a$ , which concludes the theorem.  $\square$

PROOF OF THEOREM 4.2. According to Hoeffding inequality [33], we know that

$$\Pr[(1-\epsilon)a \leq \lambda_A \leq (1+\epsilon)a] \geq 1 - 2e^{-2na^2\epsilon^2}.$$

Thus, if  $n \geq \frac{\ln(\frac{2}{\delta})}{2a^2\epsilon^2}$  such that  $1 - 2e^{-2na^2\epsilon^2} \geq 1 - \delta$ , an  $(\epsilon, \delta)$ -fairness is preserved.  $\square$

PROOF OF THEOREM 4.3. Let  $X_i \in \{0, 1\}$  be a binary random variable indicating whether  $A$  is the proposer for the  $i$ -th block. Let  $S_i := a + w \sum_{j=1}^i X_j$  be the number of stakes possessed by  $A$  after  $i$  blocks. We define  $M_i := \mathbb{E}[S_n | X_1, X_2, \dots, X_i]$  as the expectation of  $S_n$  conditioned on  $X_1, X_2, \dots, X_i$ . In particular,  $M_0 = \mathbb{E}[S_n]$  and  $M_n = S_n$ . Thus,

$$\begin{aligned} \mathbb{E}[M_i | X_1, X_2, \dots, X_{i-1}] \\ &= \mathbb{E}[\mathbb{E}[S_n | X_1, X_2, \dots, X_i] | X_1, X_2, \dots, X_{i-1}] \\ &= \mathbb{E}[S_n | X_1, X_2, \dots, X_{i-1}] = M_{i-1}, \end{aligned}$$

which indicates that  $M_0, M_1, \dots, M_n$  are martingales [18]. In addition, after observing  $X_1, X_2, \dots, X_i$ , miners  $A$  and  $B$  possess  $S_i$  and  $1+iw-S_i$  stakes, respectively. Since then, the mining game becomes that  $A$  possesses a fraction  $\frac{S_i}{1+iw}$  of staking power to compete  $(n-i)$  blocks. Similar to the proof of Theorem 3.3, we can get that

$$M_i = \mathbb{E}[S_n | S_i] = S_i + \frac{(n-i)wS_i}{1+iw} = \frac{1+nw}{1+iw} \cdot S_i.$$



Furthermore,

$$\begin{aligned} M_i - M_{i-1} &= \frac{1+nw}{1+iw} \cdot S_i - \frac{1+nw}{1+(i-1)w} \cdot S_{i-1} \\ &= \frac{1+nw}{1+iw} \cdot (S_{i-1} + wX_i) - \frac{1+nw}{1+(i-1)w} \cdot S_{i-1}. \end{aligned}$$

Since  $0 \leq X_i \leq 1$ , we have

$$M_i - M_{i-1} \leq \frac{1+nw}{1+iw} \cdot (S_{i-1} + w) - \frac{1+nw}{1+(i-1)w} \cdot S_{i-1} \triangleq \Delta_{\max}.$$

Similarly, we also have

$$M_i - M_{i-1} \geq \frac{1+nw}{1+iw} \cdot S_{i-1} - \frac{1+nw}{1+(i-1)w} \cdot S_{i-1} \triangleq \Delta_{\min}.$$

Hence,

$$\Delta_{\max} - \Delta_{\min} = \frac{(1+nw)w}{1+iw}.$$

Finally, by Azuma inequality [7], we have

$$\begin{aligned} &\Pr[|M_n - M_0| \geq \gamma] \\ &\leq 2 \exp\left(-\frac{2\gamma^2}{w^2 \sum_{i=1}^n \left(\frac{1+nw}{1+iw}\right)^2}\right) \\ &\leq 2 \exp\left(-\frac{2\gamma^2}{w(1+nw)^2 \sum_{i=1}^n \left(\frac{1}{1+(i-1)w} - \frac{1}{1+iw}\right)}\right) \\ &= 2 \exp\left(-\frac{2\gamma^2}{w^2(1+nw)n}\right). \end{aligned}$$

Setting  $\gamma = nwa\varepsilon$  and rearranging it concludes the theorem.  $\square$

**PROOF OF THEOREM 4.9.** Let  $X_i \in \{0, 1\}$  be a binary random variable indicating whether  $A$  is the proposer for the  $i$ -th block. Let  $Z_n = \frac{a+w \sum_{i=1}^n X_i}{1+nw}$  be fraction of stakes possessed by  $A$  after  $n$  blocks. Then, the difference between  $Z_{n+1}$  and  $Z_n$  can be written as

$$\begin{aligned} Z_{n+1} - Z_n &= \frac{a+w \sum_{i=1}^{n+1} X_i}{1+(n+1)w} - Z_n \\ &= \frac{(1+nw)Z_n + X_{n+1}w}{1+(n+1)w} - Z_n \\ &= \frac{w}{1+(n+1)w} \cdot (X_{n+1} - Z_n). \end{aligned}$$

Moreover, let  $\gamma_{n+1} = \frac{w}{1+(n+1)w}$ ,  $f(Z_n) = \mathbb{E}[X_{n+1} | Z_n] - Z_n$  and  $U_{n+1} = X_{n+1} - \mathbb{E}[X_{n+1} | Z_n]$ . Then,

$$Z_{n+1} - Z_n = \gamma_{n+1}(f(Z_n) + U_{n+1}).$$

Next, we verify that conditions (i)–(iv) given in Definition 4.4 hold almost surely. For condition (i), we know that  $\frac{w}{(1+w)n} \leq \gamma_n \leq \frac{1}{n}$  and set  $c_l = w/(1+w)$  and  $c_u = 1$ . For condition (ii), we set  $K_u = 1$  as  $|U_n| \leq 1$ . For condition (iii), we know from (1) that

$$f(Z_n) = \begin{cases} \frac{Z_n}{2(1-Z_n)} - Z_n, & \text{if } Z_n \leq \frac{1}{2}, \\ 1 - \frac{1-Z_n}{2Z_n} - Z_n, & \text{otherwise.} \end{cases} \quad (2)$$

Thus, it can be seen that  $|f(Z_n)| \leq 1$  and hence we set  $K_f = 1$ . Finally, for condition (iv), we find that  $\mathbb{E}[\gamma_{n+1}U_{n+1} | \mathcal{F}_n] = 0$  and hence we set  $K_e = 0$ .

In addition, by (2), we observe that  $f(Z_n)$  is continuous for  $Z_n \in [0, 1]$ . Thus, by Lemma 4.5,  $\lim_{n \rightarrow \infty} Z_n$  exists almost surely and is in one of the zeros of  $f(\cdot)$ . Let  $f(x) = 0$  such that the zeros are

found as  $Q_f = \{0, \frac{1}{2}, 1\}$ . Then, it remains to show that  $q = 1/2$  is an unstable point and  $q = 0$  and  $q = 1$  are two stable points.

Clearly, we have

$$f(x)(x-1/2) = \begin{cases} \frac{x(x-1/2)}{1-x} \cdot (x-1/2) \geq 0, & \text{if } x \leq \frac{1}{2}, \\ \frac{(1-x)(x-1/2)}{x} \cdot (x-1/2) \geq 0, & \text{otherwise.} \end{cases}$$

Furthermore,

$$\begin{aligned} \mathbb{E}[U_{n+1}^2 | \mathcal{F}_n] &= \mathbb{E}[X_{n+1}^2 | Z_n] - \mathbb{E}^2[X_{n+1} | Z_n] \\ &= \mathbb{E}[X_{n+1} | Z_n] - \mathbb{E}^2[X_{n+1} | Z_n]. \end{aligned}$$

Thus, if  $Z_n$  is close to  $1/2$ , i.e.,  $Z_n \in [1/2 - \varepsilon, 1/2 + \varepsilon]$  for some  $\varepsilon > 0$ , we have  $\mathbb{E}[X_{n+1} | Z_n] \in [\frac{1/2 - \varepsilon}{1+2\varepsilon}, \frac{1/2 + 3\varepsilon}{1+2\varepsilon}]$ . As a result,

$$\mathbb{E}[U_{n+1}^2 | \mathcal{F}_n] \geq \frac{1/2 - \varepsilon}{1+2\varepsilon} \cdot \frac{1/2 + 3\varepsilon}{1+2\varepsilon} \triangleq K_L,$$

which implies  $q = 1/2$  is an unstable point. Hence, according to Lemma 4.8,  $\Pr[Z_n \rightarrow 1/2] = 0$ .

Finally, we prove that  $q = 0$  is a stable point, with  $q = 1$  being analogous. Obviously,  $f(x)x < 0$  when  $x > 0$  is close to 0. Meanwhile, if  $Z_0 = 0$ , it always holds that  $\Pr[Z_n = 0] = 1$ , which implies every neighborhood of  $q$  is attainable. Consequently, by Lemma 4.7,  $\Pr[Z_n \rightarrow 0] > 0$ . Note that when  $Z_n \rightarrow 0$ , we must have  $\lambda_A \rightarrow 0$ . Therefore, when  $n \rightarrow \infty$ ,  $\Pr[(1-\varepsilon)a \leq \lambda_A \leq (1+\varepsilon)a] = 0$  for any positive  $\varepsilon$ , which concludes the theorem.  $\square$

**PROOF OF THEOREM 4.10.** The proof is, again, similar to that of Theorem 4.3 by utilizing Doob's martingale and Azuma's inequality. Let  $Y_i$  and  $S_i$  be the notations same as those in the proof of Theorem 3.5. Then,

$$S_{i+1} = S_i + \frac{wY_{i+1}}{P} + \frac{vS_i}{1+(w+v)(i)}.$$

Analogous to the analysis for Theorem 3.5, we have

$$\mathbb{E}[S_n | S_i] = S_i + \frac{(w+v)(n-i)S_i}{1+(w+v)i} = \frac{1+(w+v)n}{1+(w+v)i} \cdot S_i.$$

Moreover, in each epoch, we manually sort the shards and consider the every shard is provided  $w/P$  proposer reward and  $v/P$  inflation reward. Let  $Y_{i,j} \in \{0, 1\}$  be a binary random variable indicating whether  $A$  is the block proposer for the  $j$ -th shard of the  $i$ -th mining epoch. Let  $S_{i,j}$  be the number of stakes possessed by  $A$  after completing the  $j$ -th shard of the  $i$ -th epoch, e.g.,  $S_{i,P} = S_i$ . Conditioned on  $S_{i-1}$  and  $S_{i,j}$ , the expectation of  $S_i$  can be computed as

$$\mathbb{E}[S_i | S_{i-1}, S_{i,j}] = S_{i,j} + \frac{(P-j)(w+v)S_{i-1}}{P(1+(w+v)(i-1))}.$$

Let  $M_{i,j} = \mathbb{E}[S_n | Y_{1,1}, \dots, Y_{1,P}, Y_{2,1}, \dots, Y_{2,P}, \dots, Y_{i,1}, \dots, Y_{i,j}]$ . Obviously,  $M_0, M_{1,1}, M_{1,2}, \dots, M_{n,P}$  is a Doob's martingale with respect to  $Y_{1,1}, Y_{1,2}, \dots, Y_{n,P}$ . Furthermore,  $M_{i,j}$  can be rewritten as

$$\begin{aligned} M_{i,j} &= \mathbb{E}[S_n | S_{i-1}, S_{i,j}] = \mathbb{E}[\mathbb{E}[S_n | S_i] | S_{i-1}, S_{i,j}] \\ &= \frac{1+(w+v)n}{1+(w+v)i} \cdot \mathbb{E}[S_i | S_{i-1}, S_{i,j}]. \end{aligned}$$

Then, we bound the difference between the maximum and minimum values of martingale difference sequence. In particular, given  $S_{i-1}$  and  $S_{i,j}$ , for any  $j \leq P-1$ , we have

$$\max\{M_{i,j+1} - M_{i,j}\} - \min\{M_{i,j+1} - M_{i,j}\} \leq \frac{1+(w+v)n}{1+(w+v)i} \cdot \frac{w}{P}.$$

Meanwhile, we can also get that

$$\max\{M_{i,1} - M_{i-1,P}\} - \min\{M_{i,1} - M_{i-1,P}\} \leq \frac{1 + (w+v)n}{1 + (w+v)i} \cdot \frac{w}{P}.$$

Finally, by Azuma inequality, we have

$$\begin{aligned} \Pr[|M_{n,P} - M_0| \geq \gamma] &\leq 2 \exp\left(-\frac{2\gamma^2}{\sum_{i=1}^n \sum_{j=1}^P \left(\frac{1+(w+v)n}{1+(w+v)i} \cdot \frac{w}{P}\right)^2}\right) \\ &\leq 2 \exp\left(-\frac{2\gamma^2}{\frac{w^2(1+(w+v)n)^2}{P(w+v)} \cdot \sum_{i=1}^n \left(\frac{1}{1+(w+v)(i-1)} - \frac{1}{1+(w+v)i}\right)}\right) \\ &= 2 \exp\left(-\frac{2\gamma^2 P}{w^2(1+(w+v)n)n}\right) \end{aligned}$$

Setting  $\gamma = na(w+v)\epsilon$  concludes the theorem.  $\square$

PROOF OF LEMMA 6.1. Suppose that there are  $m$  miners. Denote by  $S^i$  the fraction of stakes possessed by miner  $i$  such that  $\sum_{k=1}^m S^k = 1$  and by  $T^i$  the waiting time of miner  $i$ 's candidate block becoming valid. Without loss of generality, we assume that  $S^1 \leq S^2 \leq \dots \leq S^m$ . As discussed in Section 2.3,  $T^i = \text{base time} \cdot X^i / S^i$ , where  $X^i$  is a random hash value uniformly distributed in the range of  $[0, 2^{256} - 1]$  such that  $\frac{X^i}{2^{256}}$  follows the continuous uniform distribution  $U(0, 1)$  asymptotically. Let  $Z^i = \frac{X^i}{2^{256} \cdot S^i}$  such that  $Z^i \sim U(0, \frac{1}{S^i})$ . Then,

given  $Z^i = z$ , we have

$$\Pr\left[\bigwedge_{j \neq i} (Z^j \geq z)\right] = \prod_{j \neq i} (1 - S^j z)^+,$$

where  $(1 - S^j z)^+ = \max\{1 - S^j z, 0\}$ . Therefore, the probability of miner  $i$  winning the next block is

$$\begin{aligned} \Pr\left[\bigwedge_{j \neq i} (T^j \geq T^i)\right] &= \Pr\left[\bigwedge_{j \neq i} (Z^j \geq Z^i)\right] \\ &= \int_0^{\frac{1}{S^i}} S^i \prod_{j \neq i} (1 - S^j z)^+ dz = \int_0^{\frac{1}{S^m}} S^i \prod_{j \neq i} (1 - S^j z) dz. \end{aligned}$$

We consider miner 1 with the minimum staking power. We have

$$\begin{aligned} \Pr\left[\bigwedge_{j \neq 1} (T^j \geq T^1)\right] &= \int_0^{\frac{1}{S^m}} S^1 \prod_{j=2}^m (1 - S^j z) dz \\ &\leq \int_0^{\frac{m-1}{1-S^1}} S^1 \left(1 - \frac{1-S^1}{m-1} \cdot z\right)^{m-1} dz = \frac{m-1}{m} \cdot \frac{S^1}{1-S^1} \leq S^1, \end{aligned}$$

where the first inequality is because the maximum is achieved at  $S^2 = \dots = S^m = (1 - S^1)/m$  and the second inequality is from the fact that  $\frac{1}{1-S^1} \leq \frac{m}{m-1}$  since  $S^1 \leq 1/m$ . Moreover, in the above inequality, “=” holds if and only if  $S^1 = S^2 = \dots = S^m = 1/m$ , and when  $S^1 < 1/m$ , such a probability is less than  $S^1$ . This completes the proof.  $\square$