PAPER *Special Section on Emerging Communication Technologies in Conjunction with Main Topics of ICETC2020*

An Incentivization Mechanism with Validator Voting Profile in Proof-of-Stake-Based Blockchain*

Takeaki MATSUNAGA†, **Yuanyu ZHANG**†, ***Nonmembers***, **Masahiro SASABE**†, ***Member***, ***and*** **Shoji KASAHARA**†a), ***Fellow***

SUMMARY The Proof of Stake (PoS) protocol is one of the consensus algorithms for blockchain, in which the integrity of a new block is validated according to voting by nodes called validators. However, due to validator-oriented voting, voting results are likely to be false when the number of validators with wrong votes increases. In the PoS protocol, validators are motivated to vote correctly by reward and penalty mechanisms. With such mechanisms, validators who contribute to correct consensuses are rewarded, while those who vote incorrectly are penalized. In this paper, we consider an incentivization mechanism based on the voting profile of a validator, which is estimated from the voting history of the validator. In this mechanism, the stake collected due to the penalties are redistributed to validators who vote correctly, improving the incentive of validators to contribute to the system. We evaluate the performance of the proposed mechanism by computer simulations, investigating the impacts of system parameters on the estimation accuracy of the validator profile and the amount of validator's stake. Numerical results show that the proposed mechanism can estimate the voting profile of a validator accurately even when the voting profile dynamically changes. It is also shown that the proposed mechanism gives more reward to validators who vote correctly with high voting profile.

key words: *blockchain, proof-of-stake, incentivization mechanism, reliability*

1. Introduction

Blockchain is a distributed ledger technology which is supporting cryptocurrencies used on the Internet. In a blockchain, transactions are stored in a block, and the new block is linked to a ledger database with a chain structure by consensus algorithms [2]. One of the consensus algorithms for blockchains is Proof-of-Work (PoW) [3], which has been adopted in Bitcoin [4] and Ethereum [5], [6]. In PoW, the block-approving procedure is called mining. However, mining requires a huge amount of computing power, causing a huge amount of electricity consumption [7], [8]. In addition, there is an upper limit to the amount of transaction data included in a block, called the block size limit [9]. Bitcoin's transaction throughput is limited by the block size limit (about 1 MB [10]) and the interval between two consecutive block generation (about 10 minutes [11]). The number of transactions that Bitcoin can process per second is approximately 7 transactions per second (7 TPS) [12], [13]. On the contrary, VISA, a well-known payment system, is known to be able to support approximately 1,800 TPS [14]. It has been pointed out in [15] that Bitcoin transaction throughput, in other words the scalability of the system, is also a problem. The energy waste and the low scalability are drawbacks of PoW.

Proof-of-Stake (PoS) [16] is a consensus algorithm developed for addressing the drawbacks of PoW. In PoS, the computing power is replaced by a deposit paid in the cryptocurrency, called stake. In PoW, a node with higher computing power obtains a higher chance to create a new block. In PoS, unlike PoW, a node that holds a larger amount of stake is more likely to create a new block. Since PoS does not perform mining, it is possible to flexibly shorten the block generation interval or increase the block size depending on the application. It is easy for PoS to improve the transaction throughput and PoS is considered to be more scalable than PoW.

In PoS, some participating nodes are selected as validators according to their amounts of stake. The main role of validators is to validate the validity of a new block and to confirm it by voting. It is important for validators not only to verify generated blocks precisely but also to vote correctly. In order to make validators vote correctly, reward-penalty-based incentive mechanism plays an important role [17].

In this paper, we consider a reward-penalty incentive mechanism based on validators' voting profiles. The voting profile of a validator represents its reliability according to its voting history. A reward or penalty for the vote of a validator is calculated according to its reliability. Conducting simulation experiments, we evaluate the performance of the proposed mechanism, investigating the impact of system parameters on the estimation accuracy of the validator profile and the amount of validator's stake.

This paper is organized as follows. Section 2 introduces the related work on the consensus algorithms for blockchains. In Sect. 3, we describe the details of our reward-penalty-based incentive mechanism in the consensus algorithm of a blockchain with PoS. Numerical examples are shown in Sect. 4, and we conclude the paper and show future work in Sect. 5.

Manuscript received February 22, 2021.
Manuscript revised June 4, 2021.
Manuscript publicized August 5, 2021.

†The authors are with Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.

*This paper is an expanded version of the paper presented at 2020 International Conference on Emerging Technologies for Communications (ICETC2020) [1].

a) E-mail: kasahara@ieee.org

DOI: 10.1587/transcom.2021CEP0004

Copyright © 2022 The Institute of Electronics, Information and Communication Engineers

2. Related Work

In Bitcoin, a participating node obtains a reward if it solves a cryptographic puzzle associated with a new block, and the new block is appended to the blockchain [18]. Since the winning probability of mining depends on the computing power of participating nodes, a lot of high-performance special hardware is invested in mining. As a result, an enormous amount of electricity is consumed, leading to environmental damage [19]. In order to address the energy-consuming drawback of PoW, PoS has been proposed. PPcoin [20] is known as the first cryptocurrency with PoS consensus mechanism.

In Bitcoin and PPcoin, when more than one node solves a cryptographic puzzle, multiple new blocks are added to the latest block and the chain branches. This phenomenon is called a fork. Bitcoin and PPcoin solve the fork issue with the longest chain rule, in which the chain with the largest number of blocks is considered as the valid chain [4], [21]. However, the longest chain rule is not enough to guarantee the finality of transactions because of the possibility of another new long chain in the future.

One of PoS-based implementations guaranteeing the transaction finality is Tendermint [22]. In Tendermint, a participating node deposits tokens as a security deposit stake. A predefined number of participating nodes are selected as validators in a descending order of stake. The validator responsible for generating a block is called a proposer. When the proposer generates a block, validators except the proposer validate the generated block.

The selected validators form a committee, and each validator votes based on its validation result of the generated block. If the number of votes for the new block is greater than or equal to the threshold prespecified by the network, the committee makes a consensus that the block is valid. The block approved by the committee is eventually appended to the blockchain. Due to the consensus with voting, fork never happens. In addition, the finality can be defined as the point at which the committee reaches the consensus on the generated block. (See Fig. 1 for details of PoS-based consensus procedure.) The blockchain technologies such as Cosmos [23], Polkadot [24], and Ethereum 2.0 [25] adopt the PoS-based consensus algorithm, in which a consensus on the validity of a block is made by simple voting of dichotomous choices, an approval or not (including an abstention). Note that in the original PoS, the validator profile is not taken into consideration. The same amount of reward (resp. penalty) is paid (incurred) to the validators who vote correctly (reps. incorrectly), regardless of their voting history.

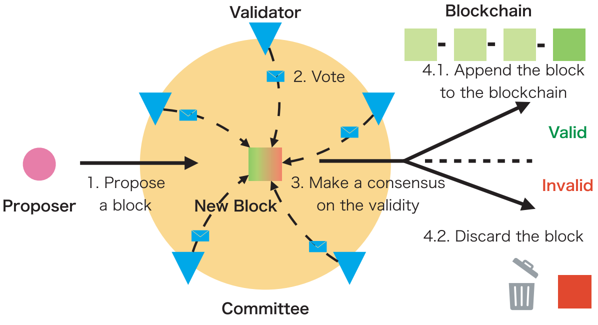


Fig. 1 Consensus procedure for PoS-based blockchain.

Leonardos et al. proposed the weighted voting for the PoS-based blockchains [26]. The weighted voting is a group-decision procedure with fixed population size [27], [28], in which the voting result is calculated with weights associated with voters' reliability such that the likelihood of the better choice of two alternatives is maximized. In [26], the authors applied the Multiplicative Weights Update (MWU) algorithm to their voting mechanism, in which the node profile is updated in a multiplicative-increase, multiplicative-decrease manner. Their update algorithm includes several parameters to be predetermined, such as a parameter controlling the increase/decrease ratio for the amount of stake and parameters controlling rates for increasing and decreasing the amount of stake. Therefore, parameter tuning is an important design issue for MWU algorithm, however, it is difficult to determine those parameters such that the MWU provides accurate estimates. Note also that rewards or penalties for validators have not been considered in [26].

The authors of [29], [30] considered a reward-penalty based mechanisms for PoS-based blockchains. In those studies, the reward/penalty is considered for the voting result of a newly generated block, whereas the voting profile of a validator is not taken into consideration. In this paper, we consider a reward-penalty incentive mechanism based on the voting profile of a validator, which is estimated from the voting history of the validator. A validator that votes correctly is given a reward. On the contrary, when the validator votes incorrectly, some of its stake are confiscated.

Notable voting-based consensus algorithms except PoS are Proof of Authority (PoA) [31] and Proof of Importance (PoI) [32]. In PoA, a group of nodes is selected as a mining leader, responsible for proposing new blocks. Aura [33] is a blockchain system adopting the PoA as the consensus algorithm. Since the authority of user(s) must be managed strictly, all the users joining blockchain network must be identified. This implies that the PoA is suitable for consortium-type blockchain[†], but not applicable to public blockchain.

The PoI is the extension of PoS and adopted in NEM [36]. In PoI, the importance of a node is quantified as score and calculated with three measures: the amount of stake held by the node, the number of transactions dealt by the node, and its partner nodes. In general, however, it is difficult to quan-

[†] A blockchain technology is classified into public and consortium/private types [34]. In a public blockchain network, any node can join the network and verify transactions and blocks without any permission. Bitcoin and Ethereum are categorized into public blockchain. In consortium/private blockchain network, only the authenticated nodes are allowed to maintain blockchains. Hyperledger projects [35] including Hyperledger Fabric are of consortium-type blockchain.

tify the relation among the nodes participating the blockchain network. In NEM, the system monitors all the nodes joining the blockchain network for identifying malicious nodes. This monitoring process becomes overhead with the increase in the number of nodes joining the blockchain network.

3. Reward-Penalty Mechanism for Proof of Stake

In this section, we show the proposed reward-penalty mechanism for PoS in detail. We summarize in Table 1 the notations used throughout the paper.

3.1 Design Goals

The voting mechanism proposed in the paper is regarded as a special case of PoI, because we focus only on the node reliability which can be measured by voting history. The design goals of the proposed voting mechanism are as follows.

- The proposed algorithm works not only for public blockchains but also for consortium and private blockchains.
- The profile of a validator can be estimated accurately and updated dynamically with the Exponentially Weighted Moving Average (EWMA) in a light-weight computing manner.
- Validators with correct voting can obtain the rewards according to their profiles, which basically motivates them to vote correctly.
- We classify incorrect voting into big fault and small one, designing the penalty mechanism in which validators are motivated to vote correctly.

In terms of penalty, we consider multiple voting as big fault, and validators that maliciously perform multiple voting are greatly penalized. This penalty gives validators an incentive to vote correctly. On the other hand, validator's incorrect voting due to hardware/software error is judged as small fault and a small amount of penalty is incurred. Note that the penalty for small fault depends on the validator profile. If a validator continues incorrect vote due to hardware/software error, the profile value of the validator decreases, making the amount of penalty for small fault large. This penalty mechanism is also expected to motivate validators to quickly recover from hardware/software trouble.

Table 1 Notations for reward-penalty mechanism.

Symbol	Description
V	Number of validators
$\mathcal{N}$	Set of validators
b_t	Validity of a proposed block at time slot t
c_t	Consensus result of the committee at t
$a_{i,t}$	Validator i's vote in time slot t for the new block
$x_{i,t}$	Voting result of validator i at t
$bf_{i,t}$	Indicator of a big fault by validator i at t
$s_{i,t}$	Amount of stake of validator i at t
$r_{i,t}$	Amount of reward/penalty for validator i at t
$reward_{i,t}$	Reward for validator i at t
$reward^{vote}_{i,t}$	Reward for validator i that cast a correct vote at t
$penalty_{i,t}$	Penalty for validator i at t
$penalty^{big}_{i,t}$	Big penalty for validator i at t
$penalty^{small}_{i,t}$	Small penalty for validator i at t
γ	Weight parameter of a big penalty
β	Amount of standard penalty per time slot
$p_{i,t}$	Voting profile of validator i at t
$\mathcal{N}_{a,t}$	Set of validators who voted correctly at t

3.2 Reward and Penalty

Let V denote the number of validators that maintain the blockchain. We define $\mathcal{N} = \{1, 2, \ldots, V\}$ as the set of validators. A block is generated in every time slot $t \in \mathbb{N} \cup \{0\}$. Let $b_t \in \{0, 1\}$ denote the validity of a block proposed at time slot t. $b_t = 1$ represents that the block is valid, while $b_t = 0$ implies that it is invalid.

For the proposed block, each validator judges its validity. We define $a_{i,t} \in \{0, 1\}$ as the vote of validator i at time slot t. If validator i judges the proposed block as valid (resp. invalid), validator i casts a confidence (resp. no-confidence) vote and $a_{i,t}$ is set to 1 (resp. 0).

The consensus of the proposed block is made by votes of validators. We define $c_t \in \{0, 1\}$ as the consensus result of the committee in time slot t. If more than two thirds of the validators cast confidence votes, i.e., $\sum_{i \in \mathcal{N}} a_{i,t} > 2V/3$, the committee eventually reaches a consensus that the proposed block is valid and c_t is set to 1. Otherwise, it reaches a consensus that the block is invalid and c_t is set to 0. The reason of the requirement of more than two-thirds confidence votes is to make the blockchain fault-tolerant for the Byzantine general problem [37].

With the consensus result, we can judge the correctness of validators' voting. We define $x_{i,t}$ as the correctness of validator i's vote at time slot t. If the vote of validator i is the same as the consensus result, i.e., $a_{i,t} = c_t$, we say validator i voted correctly, setting $x_{i,t} = 1$. If $a_{i,t} \neq c_t$, we say validator i voted incorrectly and $x_{i,t}$ is set to 0. A correct vote means that a validator casts $a_{i,t} = 1$ for a block with $c_t = 1$, or casts $a_{i,t} = 0$ for a block with $c_t = 0$. On the contrary, an incorrect vote implies that a validator casts $a_{i,t} = 0$ for the block with $c_t = 1$, or casts $a_{i,t} = 1$ for the block with $c_t = 0$. The incorrect voting may be intentional or unintentional[†]. However, it is difficult to distinguish intentional voting from unintentional one.

We consider penalties to a validator for two actions: block validation and voting. In the block validation, a validator investigates the validity of a proposed block, and the agreement of the validator's investigation result with the consensus of the proposed block is important. In terms of the voting action, on the other hand, we focus on whether the validator casts a vote correctly or not. A typical malicious manner for voting is multiple voting with which the validator casts more than one vote in a single time slot. Multiple voting is an intentional attack leading to an incorrect consensus

[†] With "intentionally," we mean that the validator casts an incorrect vote in order to attack the blockchain for falsification or malfunction. With "unintentionally," on the contrary, the validator casts an incorrect vote due to software/hardware error.

result.

Since the block validity is judged by the majority rule of more than two thirds of confidence votes by validators, the impact of one vote of a validator on the consensus is relatively small. Therefore, we regard an incorrect vote by a validator as a small fault, even when the incorrect vote is intentional. On the contrary, multiple votes by a validator significantly affect the consensus of the block, and therefore we consider multiple voting as a big fault.

Let $bf_{i,t} \in \{0,1\}$ denote the indicator of a big fault by validator i in time slot t. $bf_{i,t} = 0$ indicates that validator i didn't make a big fault, while $bf_{i,t} = 1$ implies that validator i made a big fault.

In terms of the stake held by a validator, we define $s_{i,t} \in \mathbb{R}^{+} \cup \{0\}$ as the amount of stake of validator i at time slot t. The amount of stake of validator i at time slot $t+1$, $s_{i,t+1}$, is given by the following equation

$$s_{i,t+1} = s_{i,t} + r_{i,t}. \tag{1}$$

Here, $r_{i,t}$ is the amount of reward/penalty for validator i at time slot t and given by

$$r_{i,t} = penalty_{i,t} + reward_{i,t}, \tag{2}$$

where $reward_{i,t} \in \mathbb{R}^{+} \cup \{0\}$ and $penalty_{i,t} \in \mathbb{R}^{-}$ are the reward and penalty for validator i at time slot t, respectively. The first term of (2), $penalty_{i,t}$, is divided into three cases according to $bf_{i,t}$ and $x_{i,t}$:

$$penalty_{i,t} = \begin{cases} penalty^{big}_{i,t}, & \text{if } bf_{i,t} = 1, \\ penalty^{small}_{i,t}, & \text{if } bf_{i,t} = 0, x_{i,t} = 0, \\ 0, & \text{if } bf_{i,t} = 0, x_{i,t} = 1. \end{cases} \tag{3}$$

The big penalty $penalty^{big}_{i,t}$ in (3) is given by

$$penalty^{big}_{i,t} = -\gamma \cdot s_{i,t-1}, \tag{4}$$

where γ ($0 \leq \gamma \leq 1$) is the reduction ratio by which the amount of validator's stake is reduced. That is, the value of γ determines how much of stake is confiscated based on the amount of stake $s_{i,t-1}$ of validator i at the last slot $t-1$.

If validator i casts a single vote which is eventually different from the consensus result, i.e., $x_{i,t} = 0$, we regard validator i's voting as a small fault. We define the small penalty $penalty^{small}_{i,t}$ as

$$penalty^{small}_{i,t} = -\beta \cdot (1 - p_{i,t}), \tag{5}$$

where β ($\beta \geq 0$) is the amount of standard penalty per time slot, which is specified by the blockchain protocol. $p_{i,t}$ is the voting profile of validator i at time slot t and is described in details in the next subsection. Note that $penalty^{small}_{i,t}$ is determined independently of the stake amount for validator i.

If a validator makes a big fault, a big penalty is imposed on the validator, which reduces the validator's stake largely, depending on the value of γ. On the other hand, for the validator who made no big fault but voted incorrectly, a small penalty is imposed, which is based on the standard penalty amount β and the validator's profile $p_{i,t}$.

In order to understand impacts of the above penalties on the validator's stake, we consider the following two simple cases for the evolution of the stake of validator i. In the first case, we assume that validator i makes a big fault at every time slot. Noting that no reward is given to validator i, we obtain from (1)–(4) the amount of stake of validator i at time t as

$$s_{i,t} = (1-\gamma)^{t} \cdot s_{i,0},$$

where $s_{i,0}$ is the initial amount of stake of validator i.

The second case is that validator i makes a small fault at every time slot. Suppose that validator i's profile is $p_{i,t} = 0$ for any t. Then, $s_{i,t}$ is yielded as

$$s_{i,t} = s_{i,0} - \beta t.$$

The above two approximations imply that our penalty mechanism provides multiplicative (resp. additive) decrease of validator's stake for a big (resp. small) fault, i.e., the stake of the validator who made a big (resp. small) fault is decreased rapidly (resp. gradually). It is expected that this penalty mechanism discourages validators from performing multiple voting, and also motivates them to recover from software/hardware error.

The reward of validator i at time slot t in (2), $reward_{i,t}$, is defined as follows

$$reward_{i,t} = \begin{cases} reward^{vote}_{i,t}, & \text{if } bf_{i,t} = 0, x_{i,t} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The reward for a correct vote, $reward^{vote}_{i,t}$, is given to the validator who had no big fault ($bf_{i,t} = 0$) and cast a correct vote ($x_{i,t} = 1$). Note that if the committee has not reached a consensus that the new block is valid ($c_t = 0$), the new token is not issued and hence not included in the reward. Let $\mathcal{N}_{a,t}$ denote the set of validators who voted correctly at time slot t. We calculate $reward^{vote}_{i,t}$ by the following equation

$$reward^{vote}_{i,t} = \left(token \cdot 1_{\{c_t=1\}} - \sum_{j=1}^{V} penalty_{j,t-1} \right) \cdot \frac{p_{i,t}}{\sum_{k \in \mathcal{N}_{a,t}} p_{k,t}}, \quad i \in \mathcal{N}_{a,t}, \tag{7}$$

where $token$ is the amount of new tokens issued by the system, and 1_{χ} represents an indicator function of the event χ. The $reward^{vote}_{i,t}$ is designed so that the sum of penalties at the last time slot $penalty_{i,t-1}$ is redistributed. The reward for a correct vote is also calculated according to the profile $p_{i,t}$. It is designed so that the higher the $p_{i,t}$, the more reward is given. Note that the right-hand side of the equation (7) does not include $token$ when the committee has not reached a consensus that the new block is valid ($c_t = 0$).

3.3 Validator's Profile Update

In this subsection, we describe how to estimate the voting profile of a validator, which is the indicator of the accuracy for block validation. Let $p_{i,t} \in [0,1]$ denote the profile of validator i at time slot t. The profile $p_{i,t}$ is updated by the following EWMA

$$p_{i,t} = \delta \cdot p_{i,t-1} + (1-\delta) \cdot 1_{\{bf_{i,t}=0\}} \cdot x_{i,t}, \quad 0 \le \delta \le 1, \ (8)$$

where $\delta \in (0,1)$ is the smoothing parameter of EWMA. Note that large $p_{i,t}$ implies that validator i is likely to validate a new block correctly. We can change by δ the impact of the currently observed sample on the estimation. A small δ gives more importance to the current sample than the previously estimated average. In (8), small δ results in $p_{i,t}$ that is dominantly affected by the current voting result, while large δ gives a smooth estimate of $p_{i,t}$.

The EWMA is one of Moving Average (MA) methods for forecasting in time-series analysis, and known to exhibit better accuracy among those MA methods in the short run [38]. The strong points of the EWMA are as follows.

1. The EWMA has a simple formulation for the average estimation.
2. Model components and parameters have some intuitive meaning to users.
3. We can estimate the average with limited data storage and computational effort.

A weak point of EWMA is that the accuracy of the estimation results becomes worse with the increase of the targeting period. That is, the EWMA provides good estimates in a short time interval, while its accuracy degrades for a long-term estimation.

Note that storing data and computation in blockchain incur a high cost. This implies that the estimation algorithm must estimate the average with as low cost as possible. Note also that the number of validators is not small and changes. This also requires that the estimation algorithm must be scalable. Taking into these requirements, EWMA is a good solution to predict profiles of all the validators.

The reasons for adopting EWMA to calculate $p_{i,t}$ are as follows.

- The previous voting performance of the validators should be taken into consideration for voting profiles $p_{i,t}$'s.
- The validator may temporarily cast an incorrect vote due to hardware failures and/or unstable network environments. In such situations, if the validator's computational environment returns to normal and resumes correct voting, the voting profile should follow the voting correctness as accurately as possible.

3.4 Management of Validators' Votes

The proposed method requires not only storing the approved transactions but also storing and referencing the voting history. We assume that the transactions are stored in a blockchain called a main chain, and the voting history is stored in a blockchain associated with the main chain called a side chain. The reason for requiring a side chain is as follows. If the committee comes to a consensus that a proposed new block is "invalid," the new block will be rejected. When storing votes in a block of the main chain, votes for a block on which the committee has reached a consensus as "invalid" cannot be stored because of block rejection. As a result, the voting history for invalid blocks is not recorded and hence not used in the subsequent profile updates, i.e., the voting profiles cannot be estimated accurately. To avoid this inaccurate estimation, the voting history is managed in the side chain different from the main chain. By introducing the side chain, we can maintain the voting history independent of the consensus result of the committee. See Appendix for an example of the blockchain structure with main and side chains.

4. Numerical Examples

In this section, we evaluate the performance of the proposed reward/penalty mechanism by discrete-event simulations.

4.1 Simulation Model

In our simulation, the unit of the simulation time is the block-generation time, which is constant and equal to d [s]. The simulation starts at time slot 1 and ends at T, and thus the simulation time is given by dT [s]. Note that we also regard time slot 0 is the initial state.

In each time slot $t \in \{1,\dots,T\}$, the following events occur in sequence.

1. Generating a new block.
2. Voting by validators.
3. Updating the profile based on the voting result.
4. Calculating the reward and the penalty based on the profile.
5. Updating the amount of stake.

At the beginning of time slot t, a new block is generated. The generated block is valid with probability $\xi \in [0,1]$, independent of the other generated blocks.

After a new block is proposed, each validator votes for/against the validity of the new block. We assume that validator i makes a big fault with probability η_i^{big} ($0 \le \eta_i^{big} \le 1, i \in \mathcal{N}$), or a small fault with probability η_i^{small} ($0 \le \eta_i^{small} \le 1, i \in \mathcal{N}$). Let $\zeta_i \in [0,1]$ denote the probability that validator i votes without fault. ζ_i is given by

$$\zeta_i = 1 - \eta_i^{big} - \eta_i^{small}.$$

Note that the probability that validator i makes a big or small fault is given by $1 - \zeta_i$.

Let S_t denote the total amount of stake on the network at time slot t, which is given by

$$S_t = \sum_{i=1}^{V} s_{i,t}.$$

Based on the total amount of tokens at the beginning of the simulation, S_0, we set the amount of tokens to be issued by the system at each time slot, $token$, by

$$token = \frac{S_0 \cdot \epsilon}{T}$$

where ϵ ($\epsilon > 0$) is the inflation rate [39] for the period T. Note that $token$ is constant and independent of t.

The standard penalty amount β is defined by the following equation:

$$\beta = \frac{token}{V}.$$

The amount of stake of the validator i at time slot t, $s_{i,t}$, is updated by (1) with the reward and penalty $r_{i,t}$.

In terms of the performance measure, we consider the average amount of stake of validators in time slot t, s_t^{avg}, which is given by

$$s_t^{avg} = \frac{\sum_{i \in \mathcal{N}} s_{i,t}}{V}.$$

We also consider the maximum value $s_t^{max} = \max_{i \in \mathcal{N}} s_{i,t}$ and the minimum value $s_t^{min} = \min_{i \in \mathcal{N}} s_{i,t}$. With s_t^{max} (resp. s_t^{min}), we investigate how the system rewards (resp. penalizes) validators who contribute (resp. do not contribute) to correct consensuses.

In our simulation, the block generation time is set to $d = 60$ [s], and a block generation process for two weeks is simulated. From this assumption, we have $T = 20160$. The number of validators is set to $V = 1000$. For all the validators, the initial amount of stake is set to $s_{i,0} = 1000$, and the initial profile is set to $p_{i,0} = 0.5$. From these assumptions, we have $s_0^{avg} = 1000$ and $S_0 = 10^6$. In [39], the inflation rate for a year is set to 0.15. Using this value, the inflation rate for two weeks, ϵ, is set to

$$\epsilon = 0.15 \cdot \frac{14}{365} \simeq 5.7534 \times 10^{-3}.$$

With this inflation rate, the amount of tokens issued at each time slot is $token = 0.28539$. Note that if a validator votes correctly for all the time slots of two weeks in a simulation, the resulting amount of stake of the validator is at least 1005.753. Therefore, in this simulation scenario, we assume that a user who wants to earn 5 tokens in two weeks becomes a validator.

In terms of the penalty, noting that the amount of tokens incurred for one big fault is $\gamma \cdot s_{i,t-1}$. When $\gamma = 10^{-2}$, 10 tokens are reduced if the validator holds 1000 tokens. For the validator who is attracted to earning 5 tokens in two weeks, the penalty of 10 tokens per one big fault is a severe punishment and expected to motivate the validator to vote correctly.

In a small fault case, the standard penalty amount β is given by 2.8539×10^{-4}. This is a small amount of penalty compared to the big fault case. If a validator consecutively votes incorrectly due to software/hardware error, however, the stake of the validator is continuously reduced. Therefore, it is expected to motivate the validator to fix the machine trouble as soon as possible.

Table 2 Parameter settings.

Parameter	Values
Weight of EWMA δ	$\{0.9, 0.99, 0.999\}$
Weight of the big penalty γ	$\{0, 10^{-2}, 2 \times 10^{-2}\}$
Big fault probability of validator i η_i^{big}	$\{0, 10^{-4}, 10^{-3}\}$
Small fault probability of validator i η_i^{small}	$\{0, 10^{-2}, 10^{-1}\}$

For the events of big and small faults, we consider the followings for η_i^{big} and η_i^{small}.

- The big fault probability of validator i: η_i^{big}
  Due to a large amount of penalty for big fault, we assume that validators are likely not to perform multiple voting. We consider the three cases: no multiple-voting case ($\eta_i^{big} = 0$), rare multiple-voting case ($\eta_i^{big} = 10^{-4}$), and not-rare-not-frequent multiple-voting case ($\eta_i^{big} = 10^{-3}$).
- The small fault probability of validator i: η_i^{small}
  Since a small fault results from hardware/software error, we can assume that small-fault event is more likely to occur than big-fault one. Therefore we consider the three occurrence cases for the small fault: no error case ($\eta_i^{small} = 0$), not-frequent error case ($\eta_i^{small} = 10^{-2}$), and frequent error case ($\eta_i^{small} = 10^{-1}$).

We comprehensively investigate all the combinations of the parameter values in order to show its sensitivity to the amount of validator stake. We summarize the parameter sets in Table 2.

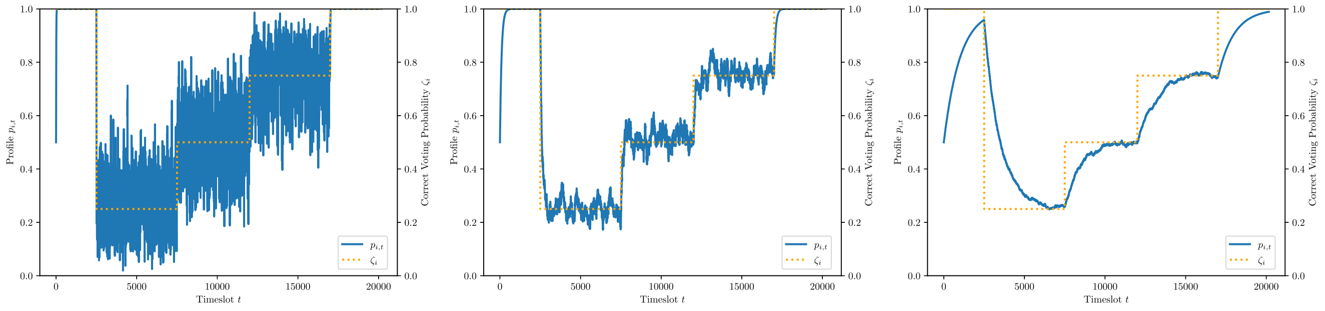
4.2 Estimation Accuracy of Voting Profile $p_{i,t}$

In this subsection, we investigate how the EWMA weight parameter δ of Eq. (8) affects the estimation accuracy of the validator-voting profile $p_{i,t}$.

For simplicity, we consider the case of $\xi = 1$, that is, all the generated blocks are valid. In this experiment, we change the correct voting probability ζ_i, investigating how correctly $p_{i,t}$ follows ζ_i. ζ_i is changed according to the following scenarios.

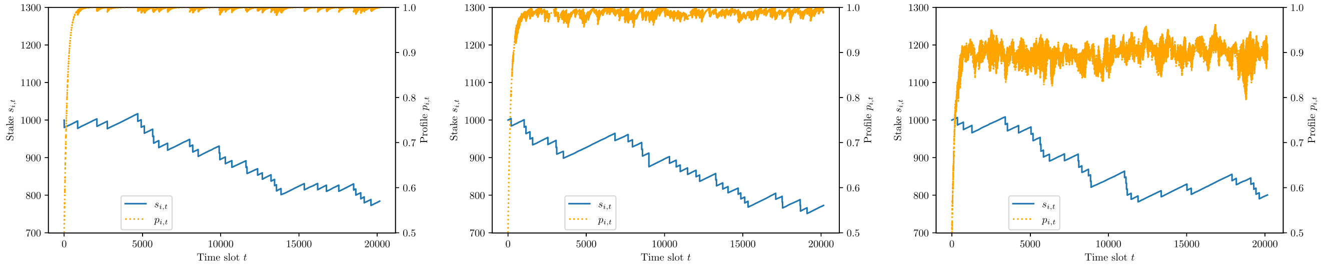
- $\zeta_i = 1$ for $1 \leq t < 2500$ and $17000 \leq t$,
- $\zeta_i = 0.25$ for $2500 \leq t < 7500$,
- $\zeta_i = 0.5$ for $7500 \leq t < 12000$,
- $\zeta_i = 0.75$ for $12000 \leq t < 17000$.

Figure 2(a) shows the evolution of $p_{i,t}$ for a certain validator. Here, δ is set to 0.9. It is observed that $p_{i,t}$ takes the value close to 1 until $t = 2500$. Then, $p_{i,t}$ suddenly decreases to the value lower than $\zeta_i = 0.25$ at $t = 2500$, and fluctuates greatly around 0.25. After $t = 7500$, $p_{i,t}$ fluctuates around 0.5, and we also observe the same tendency from



(a) $\delta = 0.9$ (b) $\delta = 0.99$ (c) $\delta = 0.999$

Fig. 2 Samples of $p_{i,t}$.



(a) $\eta_i^{small} = 0$ (b) $\eta_i^{small} = 10^{-2}$ (c) $\eta_i^{small} = 10^{-1}$

Fig. 3 Samples of $p_{i,t}$ and $s_{i,t}$.

$t = 12000$ to 17000. After $t = 17000$, $p_{i,t}$ returns to the value close to 1.

Figures 2(b) and 2(c) show the evolution of $p_{i,t}$ in cases of $\delta = 0.99$ and 0.999, respectively. It is found that $p_{i,t}$ exhibits less fluctuation with the increase of δ. This is because large δ makes current samples less significant, smoothing the estimate of $p_{i,t}$. It is also observed that $p_{i,t}$ slowly approaches the true value with the increase of ζ_i.

When designing the reward-penalty mechanism with $p_{i,t}$, it is important to consider the tradeoff between the fluctuation and convergence speed. In the following, we set $\delta = 0.99$.

4.3 Relation between Validator Profile and Stake

In this subsection, we investigate the effect of the validator profile on the amount of stake. Figure 3 illustrates sample paths of the profile and amount of stake of the validator with the minimum amount of stake at the end of simulation $T = 20160$. In this figure, we set $\delta = 0.99$, $\gamma = 2 \times 10^{-2}$ and $\eta_i^{big} = 10^{-3}$. We compare three cases of $\eta_i^{small} = 0$, 10^{-2}, and 10^{-1}.

Figure 3(a) illustrates sample paths of the profile and amount of stake of the validator in case of $\eta_i^{small} = 0$. Noting that the probability of correct voting of the validator ζ_i is 0.999, we observe in this figure that the validator profile remains the value around 0.999 in most time slots. In terms of the amount of stake, we observe the declining trend with several sudden drops. This is caused by big-fault events. Since $\gamma = 2 \times 10^{-2}$, the amount of penalty for a big fault for the validator with the amount of stake of 1000 (resp. 800) is 20 (resp. 16). We can observe sudden drops with the amount of 20 for $t \in [0, 5000]$ and those of 16 for $t \geq 1500$.

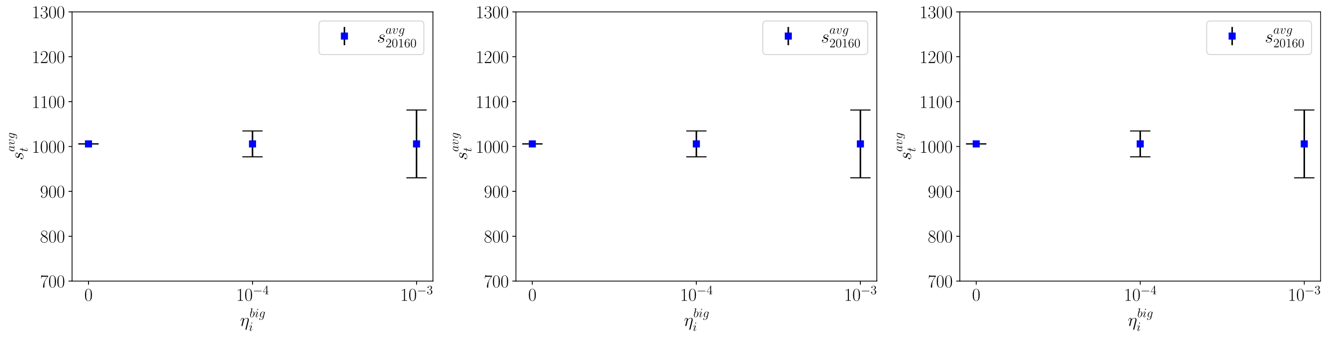
Figure 3(b) represents the sample paths in case of $\eta_i^{small} = 10^{-2}$. We observe in Fig. 3(b) that the validator profile fluctuates with larger variation than that in Fig. 3(a). This is simply due to the small-fault event that occurs with probability $\eta_i^{small} = 10^{-2}$. In terms of the amount of stake, we observe the declining trend similar to Fig. 3(a). Note that in this numerical experiment, we set the standard penalty amount of a small fault, β, is set to 2.8539×10^{-4}, which is equal to the amount of token generated in a time slot, $token$. Since the correct voting probability ζ_i is 0.989, the expected penalty amount for a small fault event is $\beta(1 - p_{i,t}) \approx \beta(1 - \zeta_i) = 3.1393 \times 10^{-6}$. That is, the penalty amount for small fault is smaller than the reward one, resulting in a small impact of penalty for small fault on the amount of the validator stake.

Figure 3(c) shows the sample paths in case of $\eta_i^{small} = 10^{-1}$. In this figure, the validator profile greatly fluctuates around 0.9. This comes from the fact that the probability of correct voting ζ_i is 0.899. In terms of the validator's stake, we observe the same tendency as Fig. 3(b).

4.4 Average Amount of Stake s_t^{avg}

In this subsection, we investigate the mean stake s_t^{avg} at the end of the period, i.e., $t = T = 20160$.

Figure 4 represents s_t^{avg} and its standard deviations at $t = 20160$. Here, we set $\delta = 0.99$ and $\gamma = 2 \times 10^{-2}$, and

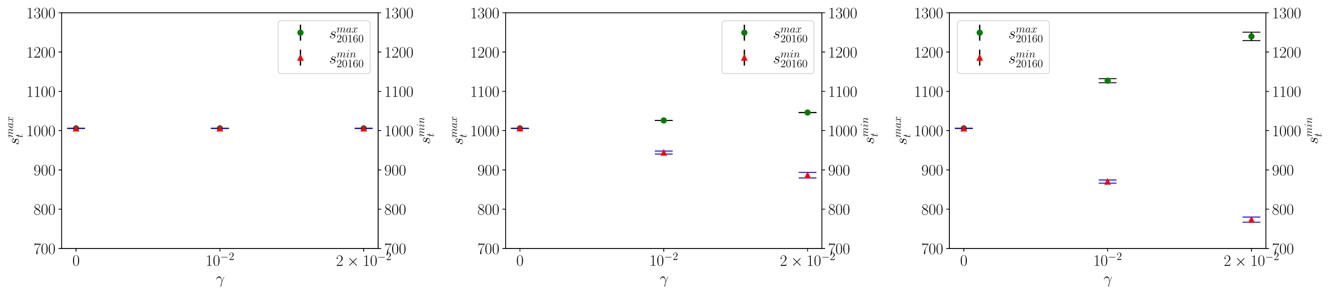


(a) $\eta^{small}_i = 0$

(b) $\eta^{small}_i = 10^{-2}$

(c) $\eta^{small}_i = 10^{-1}$

Fig. 4 s^{avg}_t vs. η^{big}_i ($\gamma = 2 \times 10^{-2}$).



(a) $\eta^{big}_i = 0$

(b) $\eta^{big}_i = 10^{-4}$

(c) $\eta^{big}_i = 10^{-3}$

Fig. 5 s^{max}_t and s^{min}_t vs. γ ($\eta^{small}_i = 0$).

s^{avg}_t's in three cases of $\eta^{big}_i = 0$, 10^{-4}, and 10^{-3} are plotted in each graph. We also compare three cases of $\eta^{small}_i = 0$, 10^{-2}, and 10^{-1}.

In Fig. 4(a), s^{avg}_t's in three cases are the same. This result implies that the stake generated during $T = 20160$ time slots is appropriately distributed to all the validators in the system. We also observe that the standard deviation of s^{avg}_t is growing with increase in η^{big}_i. For a large η^{big}_i, validators are likely to make big faults and the amount of confiscated stake increases. As a result, the amount of stake of the validator who makes some big faults decreases, while the validator who votes correctly is likely to be much rewarded. In order to clarify the rewarding/penalizing performance of the proposed system, we consider the maximum and minimum amounts of stake s^{max}_t and s^{min}_t in the following subsection.

Note that when η^{small}_i increases, there is no significant change in s^{avg}_t. This implies that small penalties do not have a remarkable impact on the validators' amounts of stake in the two-week block generation period.

4.5 Impact of Weight for Big Penalty γ

From this and the following subsections, we investigate the maximum and minimum amounts of stake s^{max}_t and s^{min}_t in detail.

Figure 5 represents the average values of s^{max}_t and s^{min}_t, and their 95% confidence intervals at $t = 20160$. Here, we set $\delta = 0.99$ and $\eta^{small}_i = 0$, and s^{max}_t and s^{min}_t's in three cases of $\gamma = 0$, 10^{-2}, 2×10^{-2} are plotted in each graph. We also compare three cases of $\eta^{big}_i = 0$, 10^{-4}, 10^{-3}.

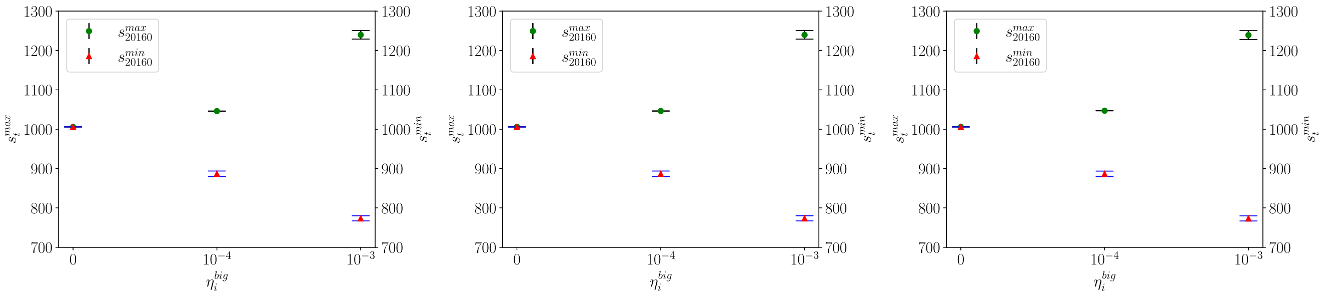
In Fig. 5(a), s^{max}_t and s^{min}_t are the same and insensitive to γ. This is simply because validators do not make big faults nor small ones with $\eta^{big}_i = \eta^{small}_i = 0$. In Fig. 5(b), when γ increases, s^{max}_t increases and s^{min}_t decreases. Remind that in the proposed mechanism, when a validator makes a big fault, the amount of its stake determined by (4) is confiscated and the resulting stake of the validator decreases. On the other hand, the confiscated stake is redistributed to validators that vote correctly by (7). This result implies that the proposed reward/penalty mechanism works correctly. In Fig. 5(b), we observe a large discrepancy between s^{max}_t and s^{min}_t with increase of γ. This suggests that γ in (4) significantly affects the stake redistribution due to big faults.

In the following Sects. 4.6 and 4.7, we set $\gamma = 2 \times 10^{-2}$.

4.6 Impact of Big Fault Probability η^{big}_i

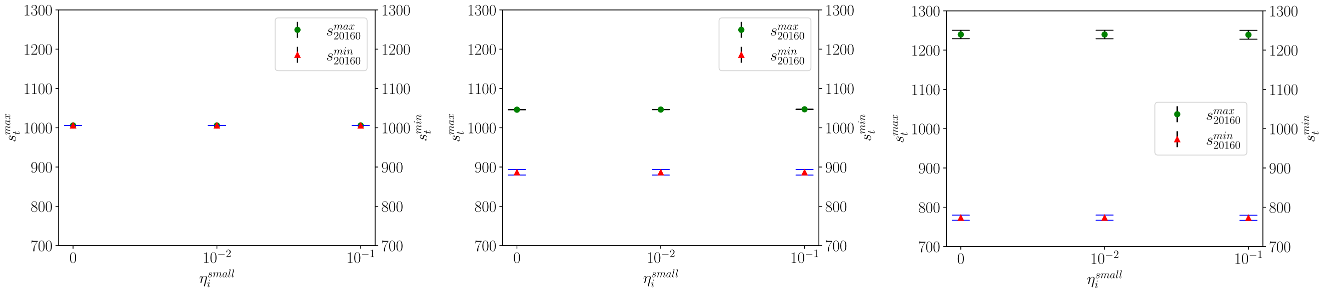
Figure 6 represents the averages of s^{max}_t and s^{min}_t, and their 95% confidence intervals at $t = 20160$. Here, we set $\delta = 0.99$, $\gamma = 2 \times 10^{-2}$, and s^{max}_t and s^{min}_t's in three cases of $\eta^{big}_i = 0$, 10^{-4}, 10^{-3} are plotted in each figure. We also compare the three cases of $\eta^{small}_i = 0$, 10^{-2}, 10^{-1}.

In Fig. 6(a), we observe that s^{max}_t (resp. s^{min}_t) greatly increases (resp. decreases) with increase in η^{big}_i. This result implies more big faults a validator makes, larger amount



(a) $\eta_i^{small} = 0$ (b) $\eta_i^{small} = 10^{-2}$ (c) $\eta_i^{small} = 10^{-1}$

Fig. 6 s_t^{max} and s_t^{min} vs. η_i^{big} ($\gamma = 2 \times 10^{-2}$).



(a) $\eta_i^{big} = 0$ (b) $\eta_i^{big} = 10^{-4}$ (c) $\eta^{big} = 10^{-3}$

Fig. 7 s_t^{max} and s_t^{min} vs. η_i^{small} ($\gamma = 2 \times 10^{-2}$).

of stake of the validator is confiscated. On the contrary, validators who correctly votes receive a large amount of reward from the system. From this result, it is expected that the proposed reward/penalty mechanism will motivate validators to vote correctly.

Note that when η_i^{small} increases, there is no significant change in s_t^{min} and s_t^{max}. This implies that small penalties do not have a remarkable impact on the validators' amounts of stake in the two-week block generation period.

4.7 Impact of Small Fault Probability η_i^{small}

Figure 7 illustrates the averages of s_t^{max} and s_t^{min}, and their 95% confidence intervals in the final time slot $t = 20160$. Here, we set $\delta = 0.99$, $\gamma = 2 \times 10^{-2}$, and s_t^{max} and s_t^{min}'s in three cases of $\eta_i^{small} = 0, 10^{-2}, 10^{-1}$ are plotted in each figure. We also compare three cases of $\eta_i^{big} = 0, 10^{-4}, 10^{-3}$.

In all the graphs of Fig. 7, no significant change of s_t^{min} and s_t^{max} is observed with increase of η_i^{small}. When η_i^{big} increases, on the contrary, the discrepancy between s_t^{max} and s_t^{min} greatly increases. This result also confirms that a big fault is penalized with a large reduction in the amount of stake, while a small fault is not penalized severely, as we designed.

The reason for the proposed reward/penalty mechanism is as follows. A validator may make a small fault when hardware failure, maintenance, or other factors make its voting temporarily impossible. If the amount of stake is significantly reduced in such cases, the risk of participating in the blockchain system becomes too high for the validator. As a result, existing validators may leave the system to avoid the risk as well as new validators may have less incentive to join the system. If this happens, the number of validators in the system is not enough to make the PoS-based consensus correctly. By suppressing the amount of reduced stake for the validator with a small fault, we prevent the validator from leaving the system.

4.8 Summary of Numerical Results

The purpose of the simulation experiment is to confirm that the validator stake is changed according to our reward-penalty policy under real economic scenarios. We investigated whether the validator stake is rewarded or penalized according to our policy in which the validator making a big (resp. small) fault is greatly (resp. slightly) penalized, while the validator voting correctly is rewarded appropriately.

We summarize the findings from the simulation results as follows.

- The EWMA-based estimation of validator profiles provides good estimates with fast convergence when the smoothing parameter is set appropriately.
- The stake generated during the simulation period is appropriately distributed to the validators.
- The validator stake is widely distributed with the increase of the weight parameter of the big-fault penalty.
- The difference between the maximum and minimum of the validator stake grows with the increase in the

probability that a validator makes a big fault.

- The validator stake slightly changes with the increase in the probability that a validator makes a small fault.

From the above results, we can confirm that the proposed reward/penalty mechanism works according to our reward-penalty policy. By adopting the proposed mechanism, it is expected that big-fault incidents such as multiple voting can be prevented by reducing a large amount of stake of malicious validators. If a validator casts an incorrect vote due to software/hardware error, the amount of stake of the validator is slightly reduced. That is, the proposed mechanism is tolerant of incorrect voting due to temporary failure. Furthermore, the proposed mechanism redistributes stake paid by the validators making big/small faults to the validators that vote correctly. This also gives validators a strong incentive to vote correctly.

5. Conclusion

In this paper, we have considered a reward-penalty-based incentivization mechanism for PoS. In the proposed mechanism, the reliability of a validator is characterized as a profile, and each validator is rewarded or penalized according to its voting result, depending on its current profile. Numerical results have shown that the proposed approach can estimate the profile of a validator accurately even when the voting profile dynamically changes. In addition, it has been shown that by adopting the profiles, validators with high profile can obtain a large amount of reward, while the amount of stakes for validators who make a big fault are significantly confiscated. Besides, we suppress the stake reduction due to a small fault to prevent validators from leaving the system.

In terms of scalability, the processing overhead for consensus becomes large with the increase of the number of validators if we allow all the nodes joining the blockchain network to be validators. Since validators play a crucial role of block consensus and hence maintaining the consistency of the blockchain, validators must be selected carefully among the nodes joining the blockchain network. In Polkadot [40], validator selection is designed such that the nodes depositing a sufficiently high bond can be nominated as validators. If we consider this bond depositing mechanism for validator selection, it is possible to control the number of validators with which the consensus overhead is prevented.

Related to the organization of validators, detecting malicious validators is a difficult task. One possible approach to this issue is to identify a malicious validator from voting history, however, this approach requires a sufficient amount of voting history for each validator, resulting in a large detection delay. Note that the above validator selection mechanism is also effective to organize a committee with honest validators. Therefore, the validator selection mechanism is significantly important for blockchains with PoS-based consensus algorithm.

In the proposed approach, the number of approval votes required for the committee to reach a consensus on the validity of a new block was set to be greater than two-thirds of the total number of validators. Therefore, if more than a third of the total number of validators are unable to vote due to some fault, consensus will not be achieved. In order to avoid such consensus failures, it is necessary to reduce the number of impaired validators who are unable to vote correctly. One of approaches to prevent the consensus failures is to increase the amount of penalty for a validator depending not only on the validator's profile but also on the time interval during which the validator's profile remains below a prespecified threshold. Our future work is designing an incentive mechanism to make validators to vote correctly such that the low-profile state of validators does not last a long time.

Acknowledgments

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI (A) under Grant 19H01103, and the Support Center for Advanced Telecommunications (SCAT) Technology Research Foundation.

References

[1] T. Matsunaga, Y. Zhang, M. Sasabe, and S. Kasahara, "Reward and penalty mechanism in proof-of-stake consensus algorithm for blockchain," Proc. 2020 IEICE International Conference on Emerging Technologies for Communications (ICETC), pp.1–4, 2020.

[2] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp.2567–2572, 2017.

[3] Bitcoin Wiki, "Proof of Work," https://en.bitcoin.it/wiki/Proof_of_work

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008.

[5] V. Buterin, "A next-generation smart contract and decentralized application platform," https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf, 2014.

[6] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," 2014.

[7] M.J. Krause and T. Tolaymat, "Quantification of energy and carbon costs for mining cryptocurrencies," Nature Sustainability, vol.1, no.11, pp.711–718, 2018.

[8] A. De Vries, "Bitcoin's growing energy problem," Joule, vol.2, no.5, pp.801–805, 2018.

[9] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," 2017 IEEE international congress on big data (BigData Congress), pp.557–564, IEEE, 2017.

[10] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," 2017 IEEE International Congress on Big Data (BigData Congress), pp.557–564, 2017.

[11] K.J. O'Dwyer and D. Malone, "Bitcoin Mining and its Energy Footprint," Proc. 25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC/CIICT), pp.280–285, 2014.

[12] K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," Financial Cryptography and Data Security, J. Clark, S. Meiklejohn, P.Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, eds., Berlin, Heidelberg,

pp.106–125, Springer Berlin Heidelberg, 2016.
[13] T. Shimizu and T. Ogawa, "Proposal of transaction processing performance improvement technology of blockchain," Proc. 2020 ICETC International Conference on Emerging Technologies for Communications (ICETC), pp.1–4, 2020.
[14] K. Otsuki, Y. Aoki, R. Banno, and K. Shudo, "Effects of a simple relay network on the Bitcoin network," Proc. Asian Internet Engineering Conference (AINTEC), p.41–46, 2019.
[15] A. Chauhan, O.P. Malviya, M. Verma, and T.S. Mor, "Blockchain and scalability," Proc. 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp.122–128, 2018.
[16] C.T. Nguyen, D.T. Hoang, D.N. Nguyen, D. Niyato, H.T. Nguyen, and E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities," IEEE Access, vol.7, pp.85727–85745, 2019.
[17] European Blockchain Association, "Staking infrastructure provider," https://github.com/European-Blockchain-Association/DSAO/blob/master/Governance/EBA_WG_EUPOS_Postition_Paper.pdf, 2020.
[18] A.M. Antonopoulos, Mastering Bitcoin, O'Reilly, 2014.
[19] C. Mora, R.L. Rollins, K. Taladay, M.B. Kantar, M.K. Chock, M. Shimada, and E.C. Franklin, "Bitcoin emissions alone could push global warming above 2°C," Nature Climate Change, vol.8, no.11, pp.931–933, 2018.
[20] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," https://decred.org/research/king2012.pdf, 2012.
[21] StackExchange, "What Does the Term "Longest Chain" Mean?," https://bitcoin.stackexchange.com/questions/5540/what-does-the-term-longest-chain-mean, 2012.
[22] J. Kwon, "Tendermint: Consensus without Mining," https://tendermint.com/static/docs/tendermint.pdf, 2014.
[23] K. Jae and B. Ethan, "Cosmos a network of distributed ledgers," https://cosmos.network/cosmos-whitepaper.pdf
[24] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," https://www.win.tue.nl/˜mholende/seminar/references/ethereum_polkadot.pdf, 2016.
[25] GitHub, "Ethereum 2.0 Specifications," https://github.com/ethereum/eth2.0-specs
[26] S. Leonardos, D. Reijsbergen, and G. Piliouras, "Weighted voting on the blockchain: Improving consensus in proof of stake protocols," Proc. 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp.376–384, 2019.
[27] R.C. Ben-Yashar and S.I. Nitzan, "The optimal decision rule for fixed-size committees in dichotomous choice situations: The general result," Int. Econ. Rev., vol.38, no.1, pp.175–186, 1997.
[28] L. Shapley and B. Grofman, "Optimizing group judgmental accuracy in the presence of interdependencies," Public Choice, vol.43, no.3, pp.329–343, 1984.
[29] A. Ouaguid, N. Abghour, and M. Ouzzif, "Towards a new reward and punishment approach for blockchain-based system," Proc. 2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS), pp.1–7, 2019.
[30] V. Buterin, "Slasher: A punitive proof-of-stake algorithm," https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm, 2014.
[31] S.D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," Proc. Italian Conference on Cyber Security, Jan. 2018.
[32] "What is POI?," https://docs.nem.io/ja/gen-info/what-is-poi
[33] "The aura blockchain consortium," https://auraluxuryblockchain.com/
[34] W. Wang, D.T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D.I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," IEEE Access, vol.7, pp.22328–22370, 2019.
[35] "Hyperledger profect," https://www.hyperledger.org/
[36] "NEM ecosystem blockchain," https://nem.io/
[37] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," ACM Trans. Program. Lang. Syst., vol.4, no.3, pp.382–401, July 1982.
[38] E.S. Gardner, Jr., "Exponential smoothing: The state of the art," J. Forecast., vol.4, no.1, pp.1–28, 1985.
[39] GitHub, "Signal non-final statu of base reward and desired issuance goal #971," https://github.com/ethereum/eth2.0-specs/pull/971
[40] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," https://polkadot.network/PolkaDotPaper.pdf, 2016.

Appendix: Blockchain Structure with Main and Side Chains

It is possible to construct blockchain systems with one main chain and side chain(s) without PoWs. A notable instance of such blockchain framework is Ethereum 2.0, in which we can connect a PoS-based main chain with PoS-based side chains.

We consider the following simple architecture of a main chain connected to a side chain. At each voting slot, voting results of validators are stored in a block for the side chain. At the same time, those are hashed and stored into a block for the main chain. When the block for the main chain is confirmed with PoS consensus algorithm, the block for the side chain is also confirmed indirectly.

Figure A· 1 shows an example of our blockchain structure. In this figure, black squares and white ones with dotted line are confirmed and discarded blocks in the main chain, respectively, while gray squares below the main chain are blocks for the side chain. At time slot 1, the proposed block is confirmed by voting result $c_1 = 1$ and connected to the main chain. The validator votes of this block are stored into block 1 of the side chain and its hash value is included in the block of the main chain proposed at time slot 2. If the secondly proposed block for the main chain is validated, block 1 of the side chain is also confirmed.

At time slot 2, the proposed block is also confirmed and the voting result is stored in block 2 of the side chain and its hash value is included in the block of the main chain proposed at time slot 3. However, the thirdly proposed block is discarded according to the voting result $c_3 = 0$. In this case, the hash value for data merged with blocks 2 and 3 in the side chain is computed and included in the block of the main chain proposed at time slot 4. If the block of the main chain proposed at time slot 4 is confirmed, blocks 2 and 3 of the side chain are also confirmed.

With this blockchain structure, we can avoid recursive construction for side chains.

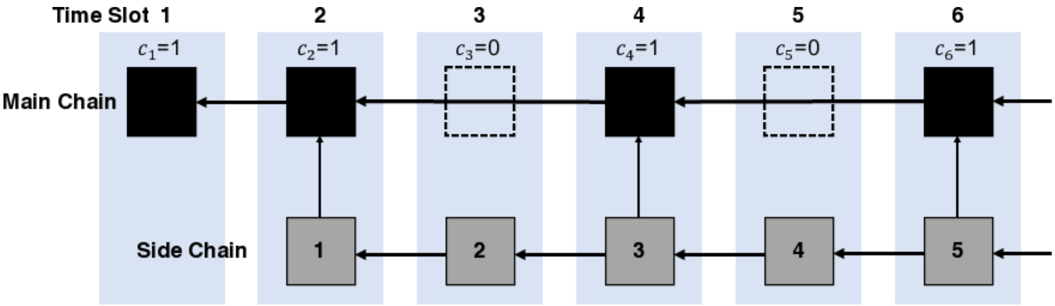


Fig. A· 1 Blockchain Structure with Main and Side Chains.

Takeaki Matsunaga received the B.A. degree from Hiroshima University, Hiroshima, Japan in 2016. He is currently a Master's Student with the Division of Information Science, Graduate School of Science and Technology, Nara Institute of Science and Technology, Nara, Japan. His current research interest is blockchain.

Yuanyu Zhang received the B.E. and M.S. degrees from Xidian University, Xi'an, China, in 2011 and 2014, respectively, and received the Ph.D. degree from the School of Systems Information Science, Future University Hakodate, Hokkaido, Japan in 2017. He is currently an Assistant Professor with the Division of Information Science, Graduate School of Science and Technology, Nara Institute of Science and Technology, Japan.

Masahiro Sasabe received the B.S., M.E., and Ph.D. degrees from Osaka University, Japan, in 2001, 2003, and 2006, respectively. He was an Assistant Professor with the Cybermedia Center, Osaka University from 2004 to 2007, an Assistant Professor of Graduate School of Engineering, Osaka University from 2007 to 2014. He is currently an Associate Professor of Graduate School of Science and Technology, Nara Institute of Science and Technology, Japan. His research interests include P2P/NFV networking, game-theoretic approaches, and network optimization. Dr. Sasabe is a member of IEEE and IEICE.

Shoji Kasahara received the B. Eng., M. Eng., and Dr. Eng. degrees from Kyoto University, Kyoto, Japan, in 1989, 1991, and 1996, respectively. Currently, he is a Professor of Nara Institute of Science and Technology, Nara, Japan. His research interests include stochastic modeling and analytics of large-scale complex systems based on computer/communication networks.