The Comprehensive Investigation and Discussion on the Progress of Ethereum Proof-of-Stake

Xiaojie Feng*

College of Software Engineering, Sichuan University, 610207 Chengdu, China

Abstract. Proof of Stake (PoS), as an important consensus algorithm in blockchain technology, has received widespread attention and research in recent years. PoS aims to solve the problems of high energy consumption and low efficiency in Proof of Work (PoW) algorithms. By allowing validators to participate in the consensus process based on their economic interests, it achieves low energy consumption and high efficiency in block verification. This article examines the latest developments in five consensus algorithms for proof of stake, with the aim of identifying errors, omissions, and shortcomings in the original proof of stake. This article finds that although proof of stake has been widely used in the market, PoS algorithm still faces challenges such as how to further improve consensus efficiency, enhance system security, and effectively respond to malicious attacks. In order to address these challenges, this article presents some of its own opinions and proposes some summarized solutions and areas for improvement. This article aims to promote the continuous development of PoS algorithm through these perspectives, in order to adapt to more complex and diverse blockchain application scenarios.

1 Introduction

Ethereum is a blockchain with a virtual computer buried within. Ethereum employs smart contract technology to enable a decentralized Ethereum Virtual Machine to process peer-to-peer contracts using its dedicated coinage, Ether (ETH). E Ethereum initially employed proof-of-work, but transitioned to proof-of-stake in September 2022. Using this consensus process, anyone wishing to add new blocks to the chain must first stake ETH, Ethereum's native currency, as collateral before running validator software. These "validators" can then be chosen at random to propose blocks for other validators to review and add to the blockchain. Compared with proof-of-work, proof-of-stake brings many advantages, such as higher energy efficiency which reduces computational consumption and hardware requirements, and increased number of network security nodes to improve security. However, proof-of-stake is still in its infancy and has undergone limited practical testing.

Therefore, current research has also pointed out many flaws and shortcomings in the proof-of-stake mechanism. For instance, Mišić et al. point out that some design principles

* Corresponding author: 2019141500001@alu.scu.edu.cn

used in proof-of-stake implementation actually enable attackers to launch attacks at a very low cost, and the authors also propose some simple remedial measures [1]. In another paper, Asgaonkar et al. designed a confirmation rule for Ethereum's consensus protocol Gasper, which may standardize fast block confirmations in Gasper. Gasper is a process that determines how validators are paid and penalized, which blocks are accepted and rejected, and which branch of the blockchain to develop on [2].

Gasper is a combination of Casper the Friendly Finality Gadget (Casper-FFG) and the LMD-GHOST fork choice algorithm. Recently, researchers have also made corrections and improvements to these two algorithms. D'Amato et al. points out a new improved algorithm, RLMD-GHOST, a synchronous consensus system that enables dynamic availability while maintaining safety during asynchronous times. In addition, they propose the "generalized sleepy model," which supports their findings. their approach improves on Pass and Shi's sleepy model by imposing stronger limitations on the adversary's corruption and drowsiness power. This approach enables us to test various dynamic participation regimes, including complete dynamic participation and no dynamic participation [3].

There are also some studies that focus on analyzing the practical results of proof-of-stake mechanisms, pointing out some errors and areas for improvement observed in experiments. A PRISM+ model written by Bistarelli et al., customized to the proof-of-stake protocol, provides significant insights into validator behavior and stake distribution over time. Their simulations show that the wealthiest validators keep their edge, while the less affluent struggle to enhance their standing [4].

In the following paper, the author will summarize the shortcomings of the original Gasper. Then based on previous research, the author will propose new ideas about an improved algorithm concept. The remainder of the chapter is organized as follows. Firstly, the author provides an overview of the five recent improvements made to the Ethereum consensus mechanism algorithm in Section 2. Then, in Section 3, the author discusses the innovative ideas obtained from various improvement schemes and future prospects. Finally, Section 4 provides a summary of this paper and presents the conclusions drawn from the improvement proposals discussed in this article.

2 Methods

2.1 e-PoS

Saad et al. notice that there are two main drawbacks in the original PoS-based protocols [5]. One is that a wealthy miner has a higher probability of winning more auctions, which increases his wealth and raises his chances of winning additional auctions. Another one is the absence of fairness guidelines during a system attack. To handle these, Saad et al. introduce an extended version of Pos, e-Pos, to resist network centralization, and promotes fair mining.

An irreversible smart contract that implements the guidelines of a PoS auction is executed by a group of miners in the abstract implementation of e-PoS. The e-PoS smart contract modifies the traditional PoS architecture in a modular way to provide the desired characteristics, such as decentralization and fairness. On the other hand, consensus from several parties (miners in a cryptocurrency) is necessary for its proper execution. Saad et al. employ the Practical Byzantine Fault Tolerance (PBFT) protocol for that. Block mining in e-PoS is divided into several epochs, denoted as E1, ..., and Ej. The smart contract generates a series of blocks B1, ..., Bl in each epoch. For every block, the smart contract computes the baseline stakes (ST1, ..., STl) and notifies the network of them. In order to participate in the block auction, candidate miners C (mc, bc) compare their balance to the baseline stake and then bid on the block they want to mine. Based on their bids, the smart contract determines

which miners make up the final list, K (mk, bk). The miners of the previous epoch Ej−1 execute the smart contract using PBFT for each epoch Ej, then hand off control to the miners of the subsequent epoch.

2.2 A refined confirmation rule for LMD-GHOST-HFC

Asgaonkar et al. introduce a refined confirmation rule which is predicated on the assumption that the protocol's set of participants remains constant, with no new additions, and that there are no rewards, exits, or penalties for honest participants [2].

Asgaonkar et al. start by creating an independent protocol that serves as the basis for the LMD-GHOST Confirmation Rule. Additionally, they improve this Confirmation Rule by adding the effects of FFG-Casper, another essential Gasper component that makes blocks definitive. Fast block confirmations are the goal of the integrated Confirmation Rule for LMD-GHOST-HFC that is presented in this study, which strikes a balance between the trade-off between confirmation speed and safety assurances [2]. Within the Gasper protocol, the Confirmation Rule put out in this work may provide a standardized method for quicker and more dependable block confirmations.

2.3 Goldfish

An essential part of Ethereum's proof-of-stake is the LMD-GHOST consensus mechanism. However, in the current form, this protocol is fragile as it stands, as demonstrated by recent attacks and patching efforts. D'Amato et al. notice this fact and present Goldfish, a novel protocol that meets the essential criteria needed to serve as a drop-in replacement for LMD GHOST: it is safe in the sleepy model, presuming that most validators adhere to the protocol [6].

A unique coordination structure that enables sincere voters to unite behind sincere initiatives is essential to Goldfish. This method, in contrast to the previously stated ones, permits quick confirmations and reorganizes resilience in the sleepy model. It is predicated on two methods that are uncommon in the literature. One is Message buffering, which implies that every validator carefully considers when to include votes from the network into its local view, giving the proposer's votes top priority. Each validator buffers the votes it receives. Another one is voting expiry, which implies that only votes from the slot that came right before have an impact on the actions of honest validators during each slot [6].

2.4 RLMD-GHOST

Even in the case of complete involvement, Neu et al. show that the initial iteration of LMD-GHOST, Gasper's dynamically available component, is not secure [7].

Goldfish which introduced in 2.3 is a solution put up by D'Amato et al. to deal with the problems caused by LMD-GHOST. Goldfish is also introduced by the team led by D'Amato, but the protocol's vulnerability to transitory asynchrony makes it difficult to substitute LMD-GHOST in Ethereum [3].

To investigate the trade-off space between resilience to temporary asynchrony and dynamic availability, D'Amato et al. propose a new family of synchronous consensus protocols, namely Recent Latest Message Driven GHOST (RLMD-GHOST), which uses the generalized sleepy model to analyze the properties [3].

2.5 An improved Casper network using genetic algorithm on hyperparameters selection

Sharma S applies the fundamental ideas that were used for HotStuff to the Casper finality device. Gasper-Siesta was designed with the fork-choice rule's liveness and safety for finality devices in mind. Sharma demonstrates that Gasper-Siesta is practicable to integrate into the Ethereum ecosystem and will, in fact, lower commit latency [8].

The fundamental ideas underlying Gasper-Siesta are similar to those of BeeGees [9]. Sharma can ascertain whether epochs not in the non-contiguous commit path could have been committed by looking at earlier attestations. Sharma can stop any conflicting commits if Sharma discovers that there's a chance that another epoch was also committed. This method stays clear of the "ping-pong effect" that the previous example illustrates [8].

The Ethereum consensus specification has about 75 lines of code delta as a result of Sharma's protocol changes being implemented. This covers modifications to the auxiliary functions, core functions, and data structures. Additionally, Sharma developed testing for Sharma's adjustments to guarantee consistency with current Ethereum functionality in terms of updating the network to make advantage of Sharma's updates and the accuracy of Sharma's data structures. Most significantly, Sharma additionally verifies that finalization is valid under other conditions, such as those tested by Ethereum and tests that guarantee finalization over non-contiguous epochs [8].

3 Discussion

Overall, PoS uses a mechanism that is akin to conventional distributed consistency verification, with a distributed algorithm to choose the checkpoint nodes for each iteration and the quantity of tokens (or storage capacity, etc.) serving as the weight basis. This approach has the benefit of not requiring a self-certification procedure, which saves computational resources. However, it has the limitations of placing a lot of strain on the network during each election, affecting a significant number of nodes.

Meanwhile, in contrast to PoW, PoS has greater requirements for code implementation and needs an excessive number of nodes to confirm and communicate with one another. According to the aforementioned research, high code complexity also results in an excessive number of intermediate steps, which increases the likelihood of security vulnerabilities. There are some of the shortcomings of proof-of-stake mentioned in the aforementioned research and summarized by in this study. They can be summarized as follows.

1. The trend towards centralization: PoS-based applications eventually tend to become more centralized in favour of a small number of miners. This is because a wealthy miner has a higher probability of winning more auctions, which makes him richer and raises his chances of winning more auctions in the future.
2. No policy to punish opponents and compensate victims: False voting practices can also be used as an attack vector because they often carry little to no consequences (nothing at stake) and only rarely call for the perpetrator(s) to actually commit a crime.
3. The length of time required to accept block proposals: While the block proposer should submit the proposal in the first segment of the allotted slot, any block proposals received within that or the subsequent slot will be accepted as valid [1].
4. The mortgage cost is too high: The liquidity of the tokens may decline as a result of nodes having to stake a significant number of tokens in order to take part in the packaging of blocks. Should the tokens collateralized by nodes not be redeemed promptly, the network as a whole can be affected.
5. Security risks: A malicious node could endanger the network if it attempts to conduct an attack and possesses a sizable number of tokens. Additionally, the network's security can

be impacted if the tokens that nodes have collateralized are lost or compromised. This assault can even be carried out by an attacker with a 30% stake, despite the virtually official Ethereum literature stating that this percentage of the total stake is insufficient to change the chain's finality [1].

Based on the limitations in the proof-of-stake mentioned above, this view proposes some directions for improvement in the future. They can be summarized as follows.

1. Adopt more advanced systems to ensure the decentralization of proof-of-stake. For example, the DPos mechanism adopts a representative system, where holders delegate their rights to representatives for block generation and verification. Additionally, Proof-of-Believability introduced the concept of 'Servi'. Servi not only measures users' contributions to the community, but also encourages members to contribute to the sustainable development of virtual currencies that adopt this approach. All of these can further reduce the degree of network centralization, improve the security and decentralization characteristics of the network.
2. Increase the penalty for breaking rules pertaining to virtual money. It is necessary to guarantee that users of virtual currency strictly adhere to established regulations in the procedures of trading, holding, and using, and to uphold fairness and order in the virtual currency market by bolstering the deterrent effect of community rules and platform policies. Building an effective and intelligent monitoring system is essential to establishing a reliable system for identifying violations involving virtual currencies. Big data analysis and other contemporary information technologies can also be leveraged to increase the efficiency and precision of violation detection.
3. As the time needed to accept block proposals is too long, it is necessary to reduce the amount of time that block proposals are allowed to, ideally, one slot. This can be achieved by setting global variables to limit time consumption, but it should also be achieved by reducing algorithm complexity, investing in network infrastructure, and minimizing necessary time consumption.
4. Mixing other collateral methods to reduce the amount of token collateral and improve node flexibility and token liquidity
5. Building an effective and intelligent monitoring system is essential to establishing a reliable system for identifying violations involving virtual currencies. Big data analysis and other contemporary information technologies can also be leveraged to increase the efficiency and precision of violation detection [10].

4 Conclusion

In this article, the author provides a comprehensive overview of Ethereum's proof of stake. Regarding the improvement methods for proof of stake methods in recent years, this paper mainly explored five different improvement schemes, including e-PoS, Goldfish, RLMD-GHOST, and improvement schemes for two specific aspects. The author, through discussion and analysis, believes that this field still faces great challenges in the areas of security and decentralization, and urgently needs researchers to conduct in-depth research and improvement. This article can provide a reference for researchers and interested individuals in the field of proof of stake, allowing them to clarify some of the developments in this area.

However, due to limitations in research conditions, the algorithms that the author can explore and investigate are relatively limited, and the analysis of various systems is still not deep enough. In the future, the author will personally learn some existing consensus mechanism algorithms in the market, including not only proof of stake algorithms, but also proof-of-work, Proof-of-Authority, proof-of-capacity, proof-of-Weight, proof-of-Elapsed-Time, proof-of-history, etc. The author will summarize their strengths and weaknesses, and find out a way to improve a more comprehensive and efficient consensus mechanism.

References

1. B. Mišić V, S. Naderi Mighan, J. Mišić, et al. Decentralization Is Good or Not? Defending Consensus in Ethereum 2.0. Blockchains, 2(1), 1-19 (2024).
2. A. Asgaonkar, F. D'Amato, R. Saltini, et al. A Confirmation Rule for the Ethereum Consensus Protocol. arXiv preprint arXiv:2405.00549 (2024).
3. F. D'Amato, L. Zanolini. Recent Latest Message Driven GHOST: Balancing Dynamic Availability With Asynchrony Resilience. Cryptology ePrint Archive (2023).
4. S. Bistarelli, C. Laneve, I. Mercanti, et al. Analyzing the Fairness of Proof of Stake Ethereum. dlt2024.di.unito.it (2021).
5. M. Saad, Z. Qin, K. Ren, et al. e-PoS: Making proof-of-stake decentralized and fair. IEEE Transactions on Parallel and Distributed Systems, 32(8), 1961-1973 (2021).
6. F. D'Amato, J. Neu, N. Tas E, et al. No more attacks on proof-of-stake ethereum. arXiv preprint arXiv:2209.03255 (2022).
7. J. Neu, N. Tas E, D. Tse. Ebb-and-flow protocols: A resolution of the availability-finality dilemma. In: 2021 IEEE Symposium on Security and Privacy (SP), IEEE, pp. 446-465 (2021).
8. S. Sharma. Gasper-Siesta: Reducing Ethereum's Commit Latency. eecs.berkeley.edu (2024).
9. N. Giridharan, F. Suri-Payer, M. Ding, et al. BeeGees: stayin' alive in chained BFT. In: Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing, pp. 233-243 (2023).
10. X. Tang, X. Lan, L. Li, Y. Zhang & Z. Han. Incentivizing proof-of-stake blockchain for secured data collection in UAV-assisted IoT: A multi-agent reinforcement learning approach. IEEE Journal on Selected Areas in Communications, 40(12), 3470-3484 (2022).