University of Milano-Bicocca

Master Degree

Thesis Summary

Decentralized Wealth Distribution: A Simulation Study of Proof-of-Stake Algorithm Variations

Authors:
Poveromo Marco

Supervisor:
Prof. Leporati Alberto

March 23, 2025

1 Introduction

In the world of blockchain technology, consensus algorithms play a crucial role in maintaining the integrity and security of the network. Two popular consensus mechanisms used nowadays are **Proof-of-Work (PoW)** and **Proof-of-Stake (PoS)**. While both PoW and PoS aim to reach consensus on the state of the blockchain, they differ in their approach and have distinct advantages and disadvantages.

A node with excessive control over the blockchain can compromise the security, decentralization, transparency and reliability of the network. Therefore, it is important for blockchains to maintain a balanced distribution of the control to ensure a fair network. One of the most famous cases that highlighted the problem of centralization was the case of Ethereum[1] when it switched from the PoW to the PoS algorithm on December 1, 2020. The news outlets published headlines such as "Two Addresses Control Over 45% of Ethereum Validator Nodes Post Merge" [2] and "40%+ of Ethereum PoS nodes are controlled by 2 addresses, says Santiment data" [3]. This strong dominance by a few addresses has led many users to criticize PoS, saying that this consensus model **leads to centralization** as validators are weighted based on the amount of ETH they have staked.

The simulation based study presented in this thesis report was born to deepen the critique of **reward distribution** in blockchain PoS networks, and examine the effects of variants of PoS algorithms in the redistribution of wealth and how these can contribute to the centralization effect.

2 State of art

Consensus algorithms within a blockchain are useful for securing the network and protecting it from attacks. These algorithms acts on the nodes of a blockchain and are currently divided into two main types[4]: proof-of-work (PoW) and proof-of-stake (PoS).
Proof of stake (Pos) consensus algorithms have been proposed as a solution to reduce the huge energy demand of PoW consensus algorithms [5] to solve a computationally difficult problem.

The introduction of the proof-of-stake (PoS) consensus algorithm has raised the concern for the problems of **inequality** [6][2][3] discussed within the blockchain community[7]. In particular, the community has been concerned that PoS algorithms could give the power to validate transactions to a **few validators**, the richest ones.

Studies on this topic have been carried out to analyze the problem of wealth compounding and the equitable redistribution of block rewards. The compounding of wealth [8] in PoS introduces the notion of equitability, to quantify how much a proposer can amplify his stake compared to the initial investment. They proves that the geometric reward function is maximally equitable.

Statistical indices will be considered, including the Gini coefficient[9] and Lorenz curves[10],

to measure economic inequality in a blockchain network and in particular to analyse wealth distributions and income distributions for a set of nodes in a blockchain network.
The notion of inequality can be defined in terms of change of the **Gini coefficient** over a computer simulations of block rewards. [11], where the optimum model is the one that mantains the same gini coefficient for the whole simulation.
Other PoS consensus blockchains have introduced the concept of **coin age** as mining resource, for instance, Peercoin [12], Cloakcoin [13], and Novacoin [14] use coin age to append new blocks to the blockchain distribuited ledged. The coin age variant chooses nodes based on how long ago the tokens were staked by a node. Once a node has forged a block, the coin age is restored and has to wait a certain period to be able to forge another block. This prevents large staking nodes from dominating the blockchain.

3 Project

In order to simulate the PoS variants I coded a pos parallel simulator in python, (available at *https://github.com/Povex/model*) that describes the evolution of a PoS staking system over the time. The simulator receives in input the parameters that are used to set the initial state, the selection function and the amount of rewards dispensed.

Two measures of system democracy/equality are introduced, based on the calculation of the Gini coefficient based on the Lorenz curve. Furthermore, these metrics are reproposed in order to assign a level of equality to the entire simulation rather than a single model state. The model generally follows some theoretical concepts expressed in [8] with some modifications. The simulator described below makes some **necessary assumptions** in order to describe the distribution of stakes in a non-real-world context. The set of nodes/agents that act as validators, i.e. the nodes that propose themselves to validate the transactions, remains fixed. In order to analyze the distribution of node stakes over time, it is assumed that the transaction fees are null. Since only one of the two leaders can receive the block reward, it was decided not to model the fork process. It is also assumed that all rewards received by validators are instantly staked.

In a PoS based blockchain, nodes that want to take the role of validators must put a certain number of tokens at stake. This is known as **staking**. The set of validating nodes (also called agents) is then defined as $A = \{A_1, A_2, ..., A_m\}$. These agents are responsible for validating transactions and securing the network.

The model evolves in discrete time $t \in \{1, 2, .., T\} \subset N$ where each t denotes a epoch in the model life-time. Each epoch starts with a block proposal and terminates with the update of the model state, that is, the addition of the reward to the selected validator stake or a punishment for a malicious intent.

At each time t the proof-of-stake algorithm of the blockchain elects a validator node $W(t) \in \{A_1, A_2, ..., A_m\}$, and a new block composed of a set of transaction is added to the blockchain. The validator $W(t)$ is rewarded with the block reward $r(t)$ that is an amount of tokens that is added to the $W(t)$ agent.

The stake of the validator $W(t)$ is then updated consequentially as:

$$S_{W(t)}(t+1) = S_{W(t)}(t) + r(t) \tag{1}$$

In order to generate the **initial state** at time $t = 0$, each node must set an amount of stake. The simulator provides the following functions for generating the initial distribution: constant, linear, polynomial, custom, gini. The function that generates the initial function is therefore an input parameter to the simulator.

The model can distribute a total of R tokens in T epochs. At each epoch $t \in [0, T]$ a validator $W(t) \in A$ is selected by the selection function and the reward $r(t)$ is added to the $W(t)$ stake as the equation 1. The **reward function** r is defined as $r : N \to R$, takes a time epoch in input and returns the block reward. The functions that can be chosen as input to the model are: constant reward and geometric reward.

The function $selection_f$ selects the validator $W(t) \in A$ for each epoch $t \in [0, T]$. The simulator receives in input one of the following selection functions $selection_f$: random, weighted, coin age, dynamic.

Given two states of the system $\vec{s_0}$, $\vec{s_T} \in R^m$ the metric of **Gini stake** g_s is defined as $g_s : R^m \times R^m \to [0, 1] \subset R$. In particular, g_s is a function that receives as input the final state $\vec{s_T}$ and the initial state $\vec{s_0}$ of the system and returns the level of inequality of the simulation as:

$$g_s(\vec{s_0}, \vec{s_T}) = g(\vec{s_T}) - g(\vec{s_0}) \tag{2}$$

Given two states of the system $\vec{s_0}$, $\vec{s_T} \in R^m$ the metric of **Gini reward** g_r is defined as $g_r : R^m \times R^m \to [0, 1] \subset R$. In particular, g_r is a function that receives as input the final state $\vec{s_T}$ and the initial state $\vec{s_0}$ of the system and returns the level of inequality of the simulation:

$$g_r(\vec{s_0}, \vec{s_T}) = g(\vec{s_T} - \vec{s_0}) - g(\vec{s_0}) \tag{3}$$

The newly introduced metrics g_s and g_r measure the **gain or loss of equality** of the system during the simulation from time $t = 0$ to time $t = T$, i.e. by how much the homogeneity of wealth has changed from the state $\vec{s_0}$ to the final state $\vec{s_T}$. In this sense, a positive value indicates that the system has introduced inequality, a positive value indicates that it has become more homogeneous, while a 0 value indicates that the system has maintained the same initial homogeneity.

4 Results

In this section I present the results that I have obtained by **running the simulator**, varying the input model, the selection function and the strategy used to distribute the block rewards. Graphs relating to the trend of the model are shown and graphs relating to the trend of the Gini coefficient over the epochs T will be analysed. For each model analysed, the correlations of the variables between them and the correlations with the final value of the metrics g_s and g_r will be presented. In this summery i present the model comparison experiment, using of the metrics g_s and g_r. The **goal** will be to identify the model that approaches the **optimum**, i.e. the one that satisfies $g_s = 0$ and $g_r = 0$.

The experiment takes into account parameters that are common to all models, varying the initial distributions according to their initial Gini coefficient. The following table summarizes the input parameters used for the experiment, which executes 240 models with 4 parallel executions for a total of 960 executions.

Parameter	**Value**
Parallel executions	4
Number of agents	10
Number of epochs	10k
Initial Stake Volume	1k
Reward type	Constant, Geometric
θ function	$\theta(x) = 0.475x - 0.2325$
Gini threshold	80%
Pos type	random, weighted, coin age, dynamic
Total reward	100, 1k, 10k, 100k, 1M
Initial gini coefficient	0, 0.2, 0.4, 0.6, 0.8, 1

The figures 1 and 2 show the result of the experiment, and allows to compare the wealth distribution (g_s) and the income distribution (g_r) for the presented PoS variants.

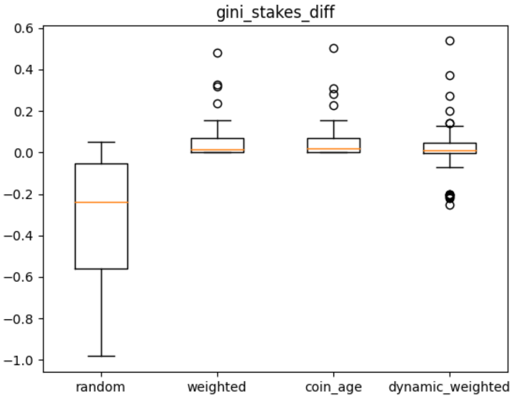

Figure 1: g_s metric grouped by pos type

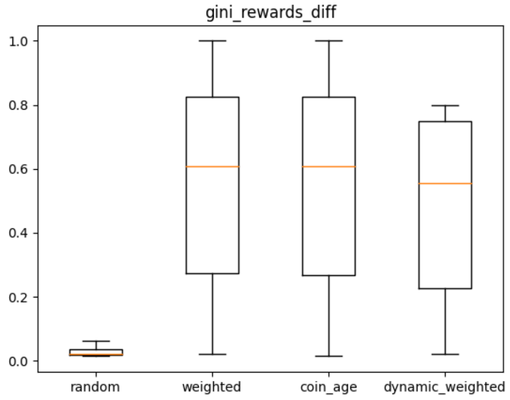

Figure 2: g_r metric grouped by pos type

The simulations based on the **Random PoS** models show an average g_s = -0.3304 ± 0.3188 with a maximum value of 0.046232 and a minimum value of -0.983612. The average metric based on the difference in rewards g_r assumes a value of 0.0279 ± 0.0114 with a minimum value of 0.0146 and a maximum of 0.055195.

The boxplots describing the **Weighted PoS** models show an average metric $g_s = 0.0614$ $\pm$ 0.102044 with a maximum value of 0.5653 and a minimum value of 0 and $g_r = 0.5664 \pm$ 0.319786 with a maximum value of 1 and a minimum value of 0.0201 . Furthermore, the boxplot values for the g_s and g_r metrics are always positive.

The **Coin Age PoS** instead present an average metric $g_s = 0.054248 \pm 0.095568$ with a maximum value of 0.5453 and a minimum value of $4.0404 * 10^-7$. The average metric takes on the value $g_r = 0.566422 \pm 0.319786$ with a maximum of 1 and a minimum value of 0.0201.

Finally, the **Dynamic PoS** simulations show an average value for the metric $g_s = 0.0176$ $\pm$ 0.132995 with a maximum value of 0.5197 and a minimum value of -0.2223, and an average metric $g_r = 0.4897 \pm 0.2613$ with a maximum value of 0.7978 and minimum value in 0.0178.

5 Conclusions

The models presented in the results chapter are compared in this section, showing their **differences** when the system is in the initial state of maximum concentration ($g = 1$), in the state of equidistribution ($g = 0$) and in the intermediate states of concentration ($g \in (0, 1)$).

In conclusion, it is possible to see how **Random PoS** models are those which, over the course of the ages, make the redistribution of wealth more homogeneous by redistributing wealth equally to all nodes of the network. This produces the effect of reducing the initial Gini coefficient highlighted by the metric g_s. Furthermore it is the model that comes closest to the optimum for the g_r metric, indicating a good equality for income distribution. If the goal is to mitigate the rich get richer problem, the Random PoS model seems to be the best choice. The negative average g_s value indicates a tendency towards reducing wealth concentration and provides a more balanced opportunity for participants. In practice, however, this model **is not feasible**, as the nodes would have no interest in staking large capital, investing in network security.

The **Weighted PoS, Coin Age PoS** and **Dynamic PoS** models show positive average g_s values, implying that nodes with larger stakes are favored, leading to a higher concentration of wealth in the hands of a few. Theese models appear to have a rich get richer problem in terms of stake distribution but they get very close to the optimal value for the metric g_s, in fact they tends to maintain the same initial Gini coefficient.

Further aspects could be considered to add complexity and to the project, also transforming the simulator into a **distributed** one (BlockSim [15]), in order to simulate ledger ramifications and network latencies. This could open up the possibility of analyzing the effects of any Denial-of-Service(DoS) or Stake Bleeding attacks to a blockchain network (LUNES-Blockchain [16]).

References

[1] G. Wood *et al.*, “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[2] Two addresses control over 45% of ethereum validator nodes post merge. [Online]. Available: https://bitcoinke.io/2022/09/ethereum-validator-nodes-control-post-merge/

[3] 40%+ of ethereum pos nodes are controlled by 2 addresses, says santiment data. [Online]. Available: https://cointelegraph.com/news/40-ethereum-pos-nodes-are-controlled-by-two-addresses-says-santiment-data

[4] O. Vashchuk and R. Shuwar, “Pros and cons of consensus algorithm proof of stake. difference in the network safety in proof of work and proof of stake,” *Electronics and Information Technologies*, vol. 9, no. 9, pp. 106–112, 2018.

[5] Ethereum’s energy usage will soon decrease by 99.95%. [Online]. Available: https://blog.ethereum.org/2021/05/18/country-power-no-more

[6] G. Rammeloo. (Oct 29, 2017) The economics of the proof of stake consensus algorithm. [Online]. Available: https://medium.com/@gertrammeloo/the-economics-of-the-proof-of-stake-consensus-algorithm-e28adf63e9db

[7] (2017) Inequality in proof-of-stake schemes: A simulation study. [Online]. Available: https://www.reddit.com/r/ethereum/comments/6x0xv8/how_does_pos_stake_concept_deal_with_rich/

[8] G. Fanti, L. Kogan, S. Oh, K. Ruan, P. Viswanath, and G. Wang, “Compounding of wealth in proof-of-stake cryptocurrencies,” in *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers 23.* Springer, 2019, pp. 42–61.

[9] R. I. Lerman and S. Yitzhaki, “A note on the calculation and interpretation of the gini index,” *Economics Letters*, vol. 15, no. 3-4, pp. 363–368, 1984.

[10] J. L. Gastwirth, “The estimation of the lorenz curve and gini index,” *The review of economics and statistics*, pp. 306–316, 1972.

[11] L. Bandelli. (2021) Inequality in proof-of-stake schemes: A simulation study. [Online]. Available: https://fse.studenttheses.ub.rug.nl/23978/

[12] S. N. Sunny King. (2012) Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. [Online]. Available: https://www.peercoin.net/read/papers/peercoin-paper.pdf

[13] (April 2015) Cloak posa v3.0 - a trustless, anonymous transaction system for cloakcoin. [Online]. Available: http://cryptochainuni.com/wp-content/uploads/CloakCoin-posa3-whitepaper.pdf

[14] Novacoin - proof of stake,. [Online]. Available: https://github.com/novacoin-project/novacoin/wiki/Proof-of-stake

[15] M. Alharby and A. van Moorsel, "Blocksim: An extensible simulation tool for blockchain systems," *Frontiers in Blockchain*, vol. 3, 2020. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fbloc.2020.00028

[16] E. Rosa, G. D'Angelo, and S. Ferretti, "Agent-based simulation of blockchains," in *Communications in Computer and Information Science*. Springer Singapore, 2019, pp. 115–126. [Online]. Available: https://doi.org/10.1007\%2F978-981-15-1078-6_10