Université de Genève

Advanced Security
14X040

Blockchain paradigms

Author: Deniz Sungurtekin

E-mail: Deniz.Sungurtekin@etu.unige.ch

May 2021

Contents

1 Introduction

1.1 Blockchain

Blockchain is a system allowing to record information in a decentralized and distributed way. This system stores a list of blocks which contains data and a cryptographic hash of the previous block which make the link between each block. Generally, these data are represented as a merkle tree, this representation is a binary tree where each node contains the result of the hash of their two children. (Not the leaf-node which are the hash of an original data).

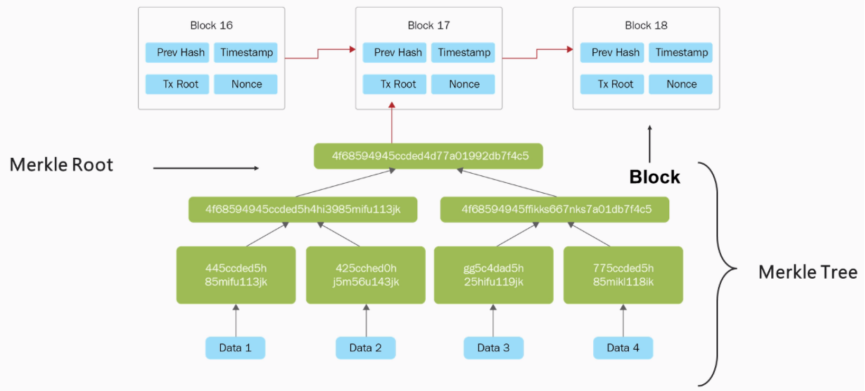


Figure 1: Blockchain Structure

Source: Here

The combination of hash functions and this merkle tree data structure is the key to keep the blockchain's data manageable and secure. This one way function produce a set of unique numbers and letters of fixed size which can be used to verify the integrity of the blocks and thus the integrity of the data. Beside, the merkle tree structure allows to significantly reduce the amount of data required to be stored and broadcast over the network by keeping only the root hash of several data.

1.2 Use and Security

Blockchain is mostly known by its application in cryptocurrencies where each block contains a number of transactions, every time a new transaction occurs, it is attached to a block which can be added to the chain. This decision is made by a consensus between all the nodes in the network, the majority of these participants must authorise it by using a consensus algorithm. The most common consensus is the proof of work which is used to select a "miner" validating the next block in the chain. These miners receive a reward in cryptocurrency for completing the task, this energy intensive process is made to dissuade malicious users to create new blocks that benefit themselves because their block will be verified and if it is refused by others, they lose the reward. Because all the blocks in the chain are attached, if one block is changed it would be immediately apparent, so if a hacker want to falsify a blockchain system, he needs to change every block across all of the distributed version of the chain which is nearly impossible. Furthermore, system like Bitcoin and Ethereum are continually growing which adds security to the ledger as blocks are being added to the chain.

Blockchain can resist traditional cyber attacks quite well, still blockchain isn't perfectly secure as we tend to think because cybercriminals are coming up with new approaches. Double-spending is a recurrent blockchain attack using to good advantage the transaction verification process. The transactions need to be verified in order to be considered as valid, it takes time and attackers can use this delay to trick the system into using the same coins in many transactions. Cybercriminals can also exploit some network vulnerabilities, for example they can execute a DDos attack on server with numerous requests, mining pools or crypto exchange platform like Bitfinex are good targets. However users' wallets are the main target for cybercriminals, hackers use methods like phishing, dictionary attacks or weakness in cryptographic algorithms to obtain the private key required to authorize each transaction.
Because blockchain have consensus rules based on majority voting, it is possible that a group of malign users acts together to corrupt the system. It means that a group controlling more than 50% of the computing power

can decide which transactions are added to the chain. Through their superiority, they can form a bigger alternative chain and therefore become the "true" chain because consensus protocol stipulate that the long chain win and that all participants must follow it. The more the blockchain has split pool of miners the more this attack is possible. Of course, all the issues related to consensus protocol depends on which algorithm is used in a blockchain system. As we mentioned it, the most known/used one is the proof of work but there is several others protocol and each of them has their own possible variation. In this work, we will mostly discuss and understand the proof of work and the proof of stake which will be soon used by Ethereum. We will implement a simple proof of concept of these protocols and compare them on some properties to acknowledge their advantages and disadvantages.

2 Proof of work

2.1 Functionality

Initially, proof of work was a way to prevent denial of service attacks such as spam on a network by demanding some work to the requester before sending a request. Nowadays, proof of work is a consensus mechanism mainly used in cryptocurrencies by which a network agree on a single source of truth, unlike centralized systems where the only source of truth is a single controlling entity. Distinct nodes must have the computational power to arrive at an agreement, all of them solves an identical cryptographic problem to arrive at consensus. When a node solves the problem, he proves to others node that a certain amount of computational effort has been made and become the trusted node.
This process has an essential role in cryptocurrencies because when a certain amount of transaction is authenticated a block is created containing the merkle tree hash of the transaction and some information on the block like timestamp or the previous block hash. Then the goal of each miner is to validate this block and to put it in the blockchain, for that they need to find a specific hash value for the block. This hash value must be below a target value which define the level of difficulty to find a hash. The more the computational power of the miners is bigger the more this difficulty is high and thus the target hash is small. This difficulty is updated each 2016 blocks, so that a new block is validate approximately every 10 minutes. Here is the recent evolution of the bitcoins difficulty:

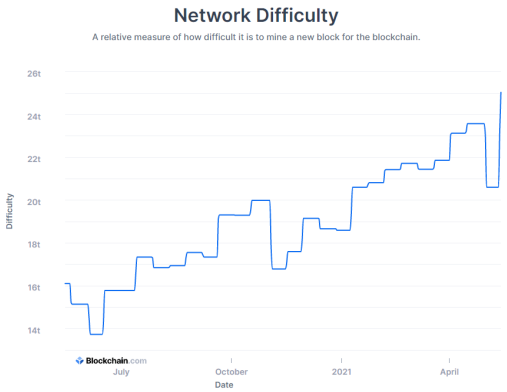


Figure 2: Difficulty evolution

Source: Here

As bitcoin is currently a pretty hot topic, there are a lot of miners who joined the race and increased the computation power and so the difficulty over the past months.
Finding the hash value requires to compute the hexadecimal value of all information of the block, putting them together and compute the digest. All this information is fixed except one, the nonce. For bitcoins this number is a 4 byte value which can be changed by the miners in order to find a correct hash, otherwise the computed hash will always be the same. In fact, we can reduce the problem of obtaining a correct hash to obtain a specific nonce, but there is no way to know the correlation between this 4 byte number and the obtained hash. So, apart from being a matter of computation power, mining is also a matter of luck. Of course, the more we have computation power the faster we can try different nonce value. Once a miner finds a correct hash, he validates the block by adding it to the chain and receive a reward.

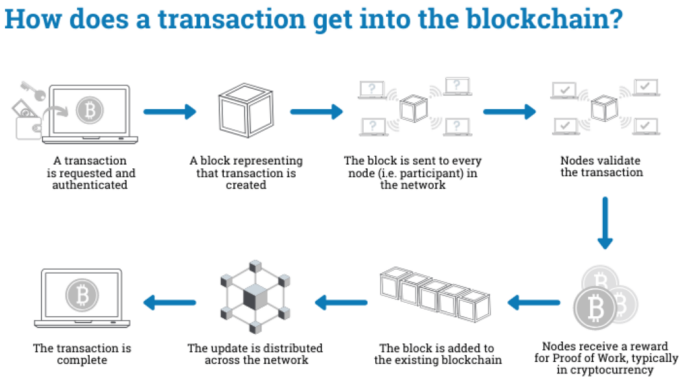


Figure 3: Blockchain Structure

Source: Here

2.2 Advantages and Issues

The main advantage of using proof of work is that this model ensures that the coins are distributed fairly and evenly across all the mining nodes. Beside, even if the cryptocurrencies using proof of work are tending to be centralized by few mining pools, these entities need to use a fair algorithm to distribute the reward between their contributors to subsist. Moreover, this phenomenon reduces considerably the search space to find a good nonce value. However, only four mining pools located in China control more than 50% of the total Bitcoin mining power. This centralization of the computational power is dangerous because the blockchain becomes more susceptible to a 51% attack and make the mining impossible for an ordinary computer which is unfair. Even so, no matter how many coins you have, it is always crucial to have a great computing power to validate blocks, so the holders of big capital are not able to take decisions for the entire network. Having said that, this method has a real environmental cost, the average electricity to mine bitcoins has surpassed the annual energy usage of more than 150 countries. It is explained by the complexity of the problem to solve, each miner uses a lot of energy by trying the different nonce value to find a valid hash but only one is taken and thus "useful". All work done by others miners is not productive despite the fact that they engaged a lot of computational power. If the miners don't belong to the same mining pool, they will try same nonce value which is pure waste. In addition, the more there are miners the bigger is the difficulty to find a hash, so the more there is a waste of energy which create a dangerous snowballing effect.

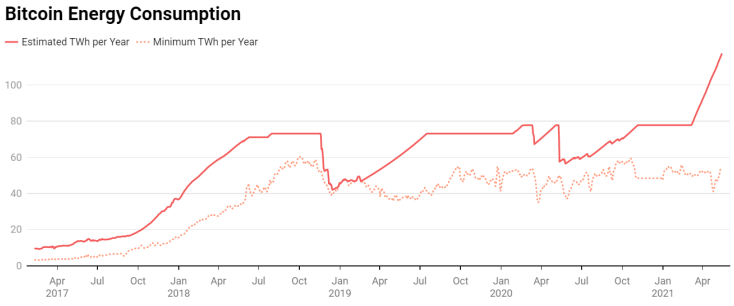


Figure 4: Evolution of estimated consumption

Source: Here

This environmental cost could place the bitcoin in the top 30 energy consumers if it were a country. Obviously, this is the main issue which can put in danger these cryptocurrencies in a foreseeable future. That is why some cryptocurrencies like Ethereum are transitioning into a proof of stake, an alternative consensus algorithm.

3 Proof of stake

3.1 Potential alternative

To avoid the environmental cost and the centralization effect, we need another solution to validate and create blocks. In order to do so, proof of stake has been invented and discussed among the users of cryptocurrencies. This model is developed on the idea that the more a node invest money in the blockchain the more he has a chance to be selected as the validator of the next blocks which saves substantial computation power because no mining is required.

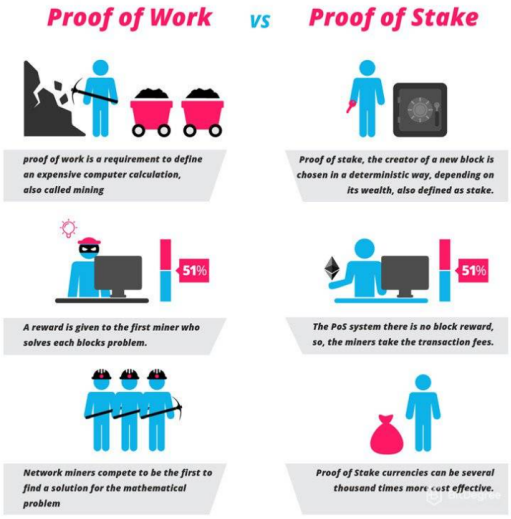


Figure 5: PoW vs PoS

Source: Here

In proof of stake a miner become a validator chosen in a deterministic way and the invested money is called a stake which is a not spendable amount of cryptocurrency. Moreover, the validator don't gain a reward but earn the transactions fees, so there is no creation of coins during each validation.
In order for the process not to favor only the wealthier nodes in the network, more unique methods has been implemented. There are multiple variants of the algorithm which a blockchain network can use, especially two of them: Randomised block selection and coin age selection systems but regardless of the approach, the more "staked" users have biggest chance to publish a new block. Randomised block selection is a method mixing proof of stake and proof of work, the validator is chosen according to the best combination of the lower hash value and the bigger stake in the network. In regards to coin age selection, each node has a coin age score which is computed by multiplying the stake and the times he has not validated a block (generally in days). Once a node has "forged" a block the time is reset and an interval is defined in which the node can't forge another block to favor a fair distribution on the selection. All this parameters specific to each cryptocurrencies must be set carefully to ensure the best performance of the system.
When a node is chosen, he verifies the transaction in the block, sign it, add it in the blockchain and receive the transaction fees. If this node decides not to forge blocks anymore, his reward and staked coin are frozen giving time to the network for verifying that a fraudulent block hasn't been added to the blockchain. If the network detects a falsified block, the validator will lose a part of his stake and his right to participate, so the security is assured so much so that the cost of attack is bigger than the potential reward.

3.2 Security and properties

To be able to control the network and validate the fraudulent transaction, a malicious entity must have a stake of more than 50% of the total amount of coin in circulation on the network which doesn't really make a financial sense. The bad actor would need to purchase the coins on the market and by buying it, the "real" value of the coin would increase and he would end up spending significantly more than he could gain from the attack. Moreover, if the network detects it, he would lose his stake.
Also proof of work has scalability issues, the maximum amount of transaction processed is significantly lower than the needs of the network due to the time to validate the transactions. With proof of stake, it is potentially possible to validate thousands of transactions per second thanks to the low complexity of the algorithm. This is good for an economic stability as well, the validators doesn't need a bigger motivation to participate in the process of forging blocks, because they don't need a big investment to have a chance to make a profit.

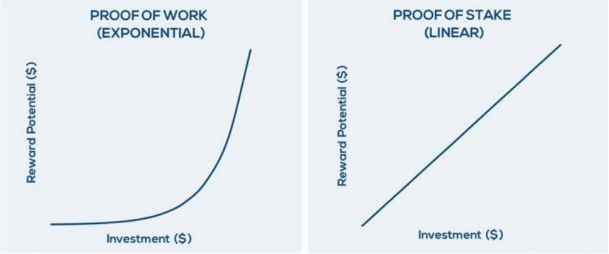


Figure 6: Reward over investment

Source: Here

The chance to be selected as a validator follows a linear evolution over the investment. In contrary, to have a good chance to find a correct hash with proof of work, an entity need a very strong computation power to be competitive. Therefore, validators don't need to earn a block's reward which doesn't add raw coins to the market. Thus, the system doesn't need to use mechanism like bitcoin-halving to slow down the creation of coins which have a big impact on the price of the cryptocurrency and isn't favorable for a long term stability. Although, proof of Stake alone doesn't improve the scalability, this architecture allows solution like sharding without reducing the security. This mechanism is a database scaling partitioning the blockchain into multiple shard chains allowing to process the blocks all at once. The sharding isn't used in proof of work because it would have a huge impact on the security as the hash power to compromise each chain will be decreased. However, in proof of stake it is possible to implement a random algorithm which ensure that the chosen validators on different chains are random which avoid a validator to take control over a chain.

As mentioned before, proof of work has a big environmental cost, on the other hand proof of stake doesn't need a lot of computation power, so the electricity cost is significantly lower.

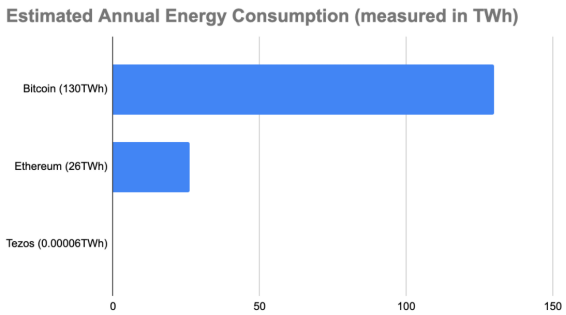


Figure 7: Consumption difference

Source: Here

Here is the comparison on ecological footprint between two of the most used proof of work cryptocurrency and Tezo's, one of the best cryptocurrency using proof of stake. Even if there is a lot much more users for the

proof of work systems and a raw difference in the cost of running these networks, we observe a difference by a factor of two millions.

However, proof of stake doesn't only have advantages. The biggest concern is fairness, obviously if the participant which staked the most coins has more chance than others of validating the block and receiving the fees, the system will help the rich getting richer. That is why it's important to manipulate the algorithm by adding some mechanism preventing this inequality. As mentioned before, these mechanisms can be the cold age selection or the randomised block selection.

4 Proof of concept

4.1 Proof of work

To have a better understanding of the functioning and on the main issues around these two consensus algorithms, I implemented two very simple proof of concept using python.
Originally, coins were intended to be mined on CPU, though we can get more hashing power from graphic cards or specific integrated circuits optimized for mining. Here we will simulate the solving process of proof of work with a simple CPU without using any parallelism. By doing this, we will try to perceive the implications of this method and understand the gap between the computational power of an ordinary user and the needed computation power to mine a block.
First, we generate a random set of numbers and letters of length 32 which represent the SHA-256 digest of a block with ASCII encoding. We suppose it contains all the information related to a block: transactions, timestamp, previous block hash, etc. This value is our challenge for which we want to find a hash value below a target hash. Then, we add a nonce (an answer) of 4 bytes to this challenge to obtain an attempt which is the concatenation of the challenge and the random nonce. After that, we simply check if our attempt verifies a given condition. This condition will define the difficulty to obtain a hash below a target. As discussed before, Bitcoin update this difficulty according to the available computing power, currently the last block has a difficulty of 25,046,487,590,083.27 = max target / current target. The max target is the target hash with difficulty 1 defined when the Bitcoin had his first run, it corresponds to the following hexadecimal numeration: 0x00000000FFFF0000000000000000000000000000000000000000000000000000.

In our implementation, we will simplify this notion of difficulty by only taking into account the numbers of "0" we have before having a non-zero value. For the above example, we would have a difficulty of 8 since there are 8 zeros before having a non-zero value. So, we will simply try to find a valid hash for the difficulties 1 to 8 and see how much resources we must use with a simple CPU. We can see an example of execution to have a better visualization.

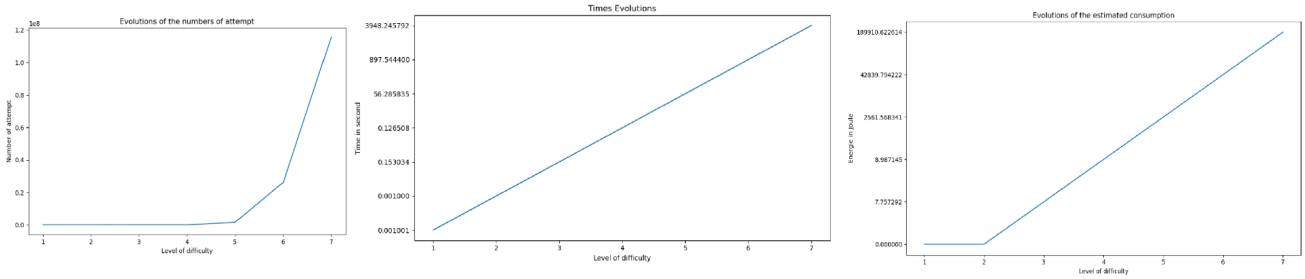
```
-----START DIFFICULTY:  4 -----
Solution Found:  00002f9de3e2496881165cb4ac5369e3a060a6c9dced48b943c69a2afe1b9e65
Elapsed Time:  5.912339
Challenge:  EPMxBgYVx8TaFkoYmYDlYnRWpXTazR4F
Answer:  MZg7
Attempt:  BljAR5nkkpHY2MNqL4R0SaLCxV7uTepCMZg7
Number of generation:  173759
Used Energy:  266.882970 J
-----END DIFFICULTY:  4 -----

-----START DIFFICULTY:  5 -----
Solution Found:  000005b1b513adcc806cd935d9776975800383c4711ea6bd5066a7fef1d01434
Elapsed Time:  7.417857
Challenge:  EPMxBgYVx8TaFkoYmYDlYnRWpXTazR4F
Answer:  9cCW
Attempt:  2Sc6DpuOZRSUzwsOzWa8rmQVwoxfJ6yU9cCW
Number of generation:  219623
Used Energy:  332.097476 J
-----END DIFFICULTY:  5 -----
```

Figure 8: Example PoW

For each difficulty, we can see the first valid hash found, the computation time, the number of nonce value tried (number of generations) and a rough estimation of the used consumption. Unfortunately, to compute the consumption I couldn't use some useful python libraries to have a better approximation cause of incompatibility

issues, thus I decided to take the total watt of my CPU and multiplied it by the current percentage of CPU-usage and the computation time to have an estimation in Joule. The CPU-usage doesn't only take into account the python execution, so the real energy consumption is probably a bit smaller, although it is enough to understand the general behaviour of the consumption.
Sad to say, but with my computation power I couldn't even solve the problem at difficulty 8 which is the lowest difficulty to mine a block with Bitcoin. After more than 10 hours of computation, I stopped the execution and here is my results.




```
Times at each difficulty:  ['0.001001', '0.001000', '0.153034', '0.126508', '56.285835', '897.544400', '3948.245792']
Attempt Number at each difficulty:  [5, 44, 4474, 3760, 1661335, 26158073, 115610062]
Energie at each difficulty:  ['0.000000', '0.000000', '7.757292', '8.987145', '2561.568341', '42839.794222', '189910.622614']
```

Figure 9: Results

Clearly, the number of iterations follows an exponential curve even in a very small scale. In regards of the computation time and consumption, they are both following a linear evolution, which is explained by the fact that the consumption strongly depends on the execution time. However, I don't doubt that this linear evolution become exponential at a bigger range because each time we increase the number of zeros by one bit, we divide the space of possible valid hash by two. In other words, it is very likely that in a bigger scale the curve will be more similar to a 2^x curve.
In this simple implementation, the amount of energy is "useful" because there is no competition, but in reality, I have "absolutely no chance" to be the first to mine a block without joining a mining pool. Therefore, all my computation power is a total waste and this is also the case for thousands of miners for each block at a much bigger scale, this waste is the main cause of this ecological issue with proof of work.

4.2 Proof of stake

Now, we will implement a simple proof of stake algorithm using coins age selection. The idea here is to simulate the behaviour of a simplified blockchain with 100 nodes to understand the impact of various parameters of the system. I made two main classes, the first one represents a block which contains a couple of information, especially the name of his validator, the previous block's hash, the transaction volume and the transaction fees which the validator will obtain. The second class implements each participant, they have a number of staked coins, a timer and a coin age score, which is computed by multiplying their stake and the time they hadn't been chosen as a validator. In this simulation, the time will be the number of times we perform a coin age selection. During a run, a participant is "randomly" chosen according to weighted probabilities given by the coin age of all participants (initially the coin age is the staked coins). This chosen participant can either verify the current block and add it in the chain correctly or try to modify its value, then another node is chosen to verify the entire blockchain, if an incorrect hash is detected the validator of this block lose his stake and can't be a validator during a long period. Afterward, the validator can't forge any blocks during 50 iterations and his timer is reset while that of others is increased.

Let's visualize a simulation with 4000 good run, 1 bad run and 4 initial genesis blocks.

```
-------Start of the Simulation---------

!!!!!!
Participant 17  Tried to falsify a block !!!!!
Participant 17  has lost all his stacked coins and his right to participate during a long period
!!!!!!

-----End of the Simulation------

The blockchains is now of size: 4005 .
There is 100 participant(s) with 32 to 100000 coins,
Before the run, the wealthier participant was Participant 50 with 99184 stacked coins.
The wealthier participant is now Participant 86 with 115264 stacked coins with 53 participation(s).
The most chosen participant is Participant 50 with 114841 coins and 56 participation(s).
```

Figure 10: Example

In this special case, we can note that the wealthier participant isn't the wealthier after the simulation even though he has validated more blocks. It's because the transaction fees are given by 1% of the transaction volume of a block which is randomly generated between a range of 10000 to 50000 coins, so he validates "cheaper" blocks. This parameter causes a kind of "unfairness" in the system, thus we can conclude that the smaller is the difference between transaction fees the better it is.
Because fairness is the biggest issue in proof of stake, it is important to analyse our results before and after the simulation specially on the distribution of the selection and the coins over our 100 participants.

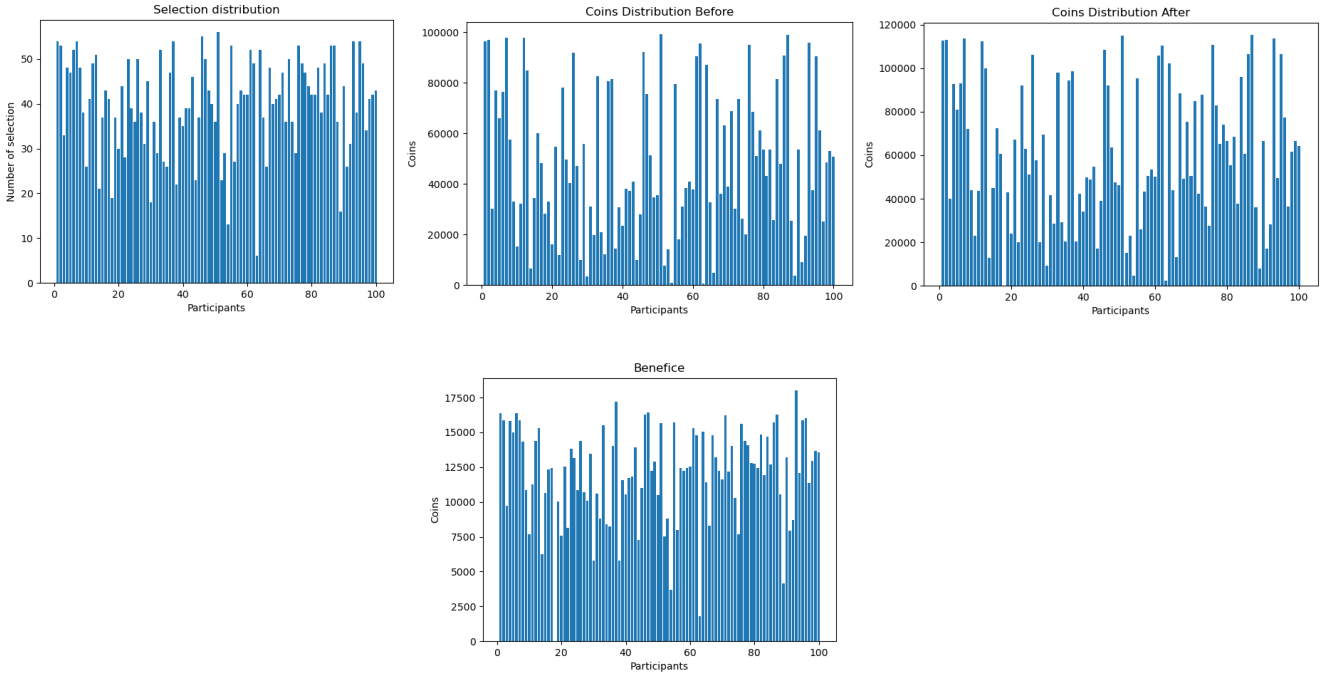


Figure 11: Results

Apart participant 17 who loses his entire stake, every node had won coins, moreover the coins distributions seems to be preserved before and after the simulation, which is good. However, my system seems to favor a little to much the participants who have a little stake, as we can see even with a very low stake a participant can make significant profit. For example, the participant 63 made a bigger benefice than he had coins before the simulation. It is probably due to the facts that the transaction fees don't depend on the stake of the validations and that the number of iterations within a participant can't validate another block is too big. This shows the complexity to find a fair solution which doesn't make the rich too much richer without making the poors too much richer. Proof of stake algorithm must take into account the coins disparity in the network to choose wisely the parameters which impact the distribution of the reward while ensuring the viability and the security of the system. In our simulation, these parameters are the minimum and maximum staked coins, the time a node can't forge after being chosen, the transaction fees and the global time management of the system.

5 Conclusion

In conclusion, we have discussed about the two main consensus algorithms used in blockchain and their advantages and disadvantages. Proof of work is based on solving a very complex problem which causes a huge ecological issue, but has shown the best result until now and dominate the cryptocurrencies market.
For its part, proof of stake is seen as the most common alternative by resolving some of the problems brought by proof of work. Instead of investing in expensive hardware, the validator invest in coins by locking up a part of them as stake. Their probability to add a block in the chain is given by the amount of their stake, which generate questions about the fairness of the system. We have seen that there exist a lot of different methods to implement a proof of stake algorithm and each of them require the system to be aware of their contributors and adjust carefully the parameters allowing the best fairness in the system. There are already some functional cryptocurrencies using proof of the stake, but soon we will witness the complete transition of Ethereum into proof of the stake. It will be an opportunity to see if this process can work in a bigger scale since Ethereum is the second most popular cryptocurrency.

6 Bibliography

References

[1] B. Academy, "Proof of stake explained," 2021.

[2] P. Basson, "Proof of stake explained," 2020.

[3] Euromoney, "How transactions get into the blockchain," >2016.

[4] ——, "The risks with public blockchains," 2020.

[5] J. Frankenfield, "Proof of work (pow)," 2021.

[6] A. Katrenko and M. Sotnichek, "Blockchain attack vectors: Vulnerabilities of the most secure technology," 2020.

[7] L. M., "Proof of work vs proof of stake: Which one is better?" 2021.

[8] B. Marr, "The 5 big problems with blockchain everyone should be aware of," 2020.

[9] E. Muzzy, "What is proof of stake?" 2020.

[10] P. R. Nair and D. R. Dorai, "Evaluation of performance and security of proof of work and proof of stake using blockchain," 2021.

[11] nirolution.com, "Proof of work vs proof of stake: Most important differences!" 2018.

[12] M. Patel, "Consensus algorithms in blockchain," 2020.

[13] S. Seth, "How do cryptocurrency mining pools work?" 2020.