POW AS POS ALGORITHM

YAACOV KOPELIOVICH

1. introduction

Since the invention of Nakamoto concensus there is an explosion of concensus algorithms that are divided into two main categories. The first one is PoW(Proof of Work) implemented in the first crypto asset(BTC) and the other one is PoS (Proof of Stake). PoS emerged is a concensus algorithm that emerged in a later stage due to dis-satisfication with the traditional PoW because of transaction settlement speed and environmental concerns. While these algorithms are closely related we haven't found in the literature an explanation or discussion how these algorithms related to each other. In this note we propose to carry such comparison and show that with a slight modification of PoS algorithm we can obtain a PoW like algorithm that will not require to spend energy yet it maintains all the PoW properties that make it superior in the eyes of many cryptp pundits. This note has 3 parts in the first section we explain what are PoS and PoW algorithms than in the second section we propose a PoS modification that incorporates the main features of PoW. The final section is a short philosophical discussion.

2. PoS and PoW - a survey

2.1. **PoS- Proof of Stake.**

The fundamental problem of any concensus algorithm is to arrange transactions chronoligically into a linear list (or a directed graph which is called blockchain) in a trustless (decentralized) way. Decentralized in this setting implies that anybody who will download this graph will be able to verify the order of transaction independently of any external source according to rigid rules set up by the system. The most natural solution for this problem is to have more than one entity(we refer to these entities as miners) who has the privilege to arrange transactions. In each round of arranging transaction a miner is selected randomly where the probability is weighted by the amount of crypto assets any of these miners pledged towards the selection process in the current round. Once the leader is selected he arranges new transactions in a block and then adds the block in a cryptographically secured way (by using a cryptographic function to serialize the data.) Any user can (the validators) verify the correctness of this block and the entire blockchain by applying the same cryptographic function. If the block is confirmed the miner will benefit by earning fees on the block that was just created. For example if we have two miners one that has 60% of the crypto assets and the other has 40% of the total pool of the crypto assets the probability of the selection will be 60% for the first user and 40% for the second user. As noted in the introduction the current PoS algorithms employ more sophisticated versions of selecting the leader who will arrange the transactions in the current round. For example Cardano blockchain overlaps this simplistic rule of choosing a miner by introducing some path dependency into the process such that selecting the probability of a leader in the current round depends not only on his crypto stake but whether or not he was selected previously (up to prior n steps of organizing transaction.)

Date: July 17, 2022.

For example in the example above we can decide that if miner A was selected in the current round the probability will drop to 0.5 rather than 0.6.

2.2. **PoW.** PoW was the first concensus algorithm based on a competition between miners that arrange transactions by performing a work solving a hash puzzle. More specifically the algorithm for transaction sequencing into a block is performed:

1. Miners get a pool of transactions from the Web
2. Miners arrange those transactions in a cryptographic way
3. after arranging the transactions Miners solve a cryptographic puzzle that has a fixed average time for its solution.
4. The miner who solves this cryptographic puzzle published it's suggested new block together with a cryptographic solution that can be validated by any user that runs the Blockchain software (it is called PoW client). If the puzzle is verified the new chain is accepted by the PoW clients. (usually after 6 confirmations)
5. The first miner that created the puzzle wins a reward by collecting the fees and crypto asset that are awarded to him.
6. As the miners can see different pool of transactions coming from different users of the system it may be that we have conflicting chains of blocks. The chain of blocks (or a linked list) that has the most electric energy (while solving the puzzle) spent on it is the one that is chosen by the PoW clients if they see two conflicting chain on the Web.

Remarkably it can be shown that this sequence of steps results in convergence and an asymptotic stable chain emerges. Let us summarise the previous discussion and tabulate the main differences between PoW and PoS:m

Property Name	Proof of Stake	Proof of Work
Money Supply	Generated on each block transaction	Generated at inception.
Block Generation	Block leader selected based on stake	Miner emerge solving a puzzle
Chain competition	One universal from inception	emerges from local competition

The main difference between the current state PoS and PoW is point (3). Indeed in PoS there is one chain from inception and thus the question whether I am on the right chain or not is solved by quering one of the stakers to verify whether or not we downloaded the correct chain. While in PoW the chain emerges from a local competition based on a rigid rules and hence the correctness of the consensus chain can be verified independently of any stake holders or miners that are creating the chain. While this is clearly a desirable property so far none of the PoS algorithms was able to meet this requirement of trusless verification. In the next section we propose a simple modification for PoS algorithm that will enable us to create a PoS with local competition thus eliminating the need to consume a physical resource like electricity as prevalent in the current PoS algorithms.This also shows that PoW can be considered as a subclass of PoS algorithms.

3. PoW adjusted algorithm

In this section we produce a modified PoW algorithm which implies that it's basically a subclass of PoS algorithms with certain twists. We start from the observation that we can estimate how long it will take each miner to produce the solution to the cryptographic puzzle based on the average performance time it can take to solve the cryptographic puzzle which is going to be essentially the miner who is going to arrange the transactions in a block. Indeed assuming that the amount of bitwise operations it takes to solve the cryptographic puzzle is n steps and assuming we have k miners that can do m_k operations per unit of time (say per second). The miner who is able to do m_i operations per second will solve the puzzle with a probability of approximately: $\frac{m_i}{\sum_{i=1}^{k} m_k}$. Further it's clear that the i-th miner will take the time of $\frac{n}{m_i}$ and therefore we can estimate what how much he will pay to solve the puzzle (if he solves it first) based on the current price of electricity available in the market. These two observations immediately lead to the following modifed PoW algorithm:

1. In the beginning there are k miners with:
   (a) p_i the probability of winning the selection lottery to arrange transactions. In reality this will be based on the miners ability to perform m_i operations per second i.e. it's hashing power.
   (b) initial payment h_i that the miner will pay regradless whether he wins the selection lottery or not. This is the initial price miners have to pay to enter into a competition
2. A randomly selected miner based on probability p_i will arrange the transactions and will send them cryptographically secured to PoW clients. (**He will not solve the puzzle**, he will just arrange them apply the cryptographic function and will send them to the PoW clients)
3. PoW clients will perform the confirmation and once it's confirmed it will be added to the blockchain
4. The total price of the chain(total payments of the fees paid by the winners) is recorded and maintained to the blockchain.
5. The fees of the transactions and any rewards minted by the system will go to the miner that arranges the transaction.

3.1. **Local Selection.** The PoW algorithm described above is equivalent to Satoshi PoW where we replace the actual physical work with a montetary transaction which depends on the miners ability to perform the cryptographic puzzle. It allows for local competition i.e. we don't need to know all the miners in the world to organize a lottery. We can do a local list of miners as if we have conflicting chains the most expensive chain (i.e. those who paid the most money) will be selected to be the unique Blockchain available to all the participants of the system.

4. Summary

In this note I explained a simple method how to implement a PoW alogrithm within Pos. Thsi is just an initial suggestion as the bolts and nots should be worked out to implement a fully formal proposal. Nevertheless it looks like the algorithm proposed maintains the best of the both worlds. On the one hand PoS enable rapid settlement of transactions while PoW allow more decentralization because of local competition. Combining bith we can overcome the BTCV trilemma obstacle (you can have two of decentralization, scalability and security but not all 3. Thsi clearly requires further research that we will pursue in future works. The author thanks Alexander Russell and Alex Kravets for their keen interest and discussion on topics related to this note.

References

[N] Nakamoto, S. Bitcoin: A Peer to Peer Electronic Cash System, //web.archive.org/web/20140320135003/https://bitcoin.org/bitcoin.pdf

[R] A.Russell, Private Communication

[ZoSo] A.Zohar, Y.Sompolinsky. Secure High-Rate Transaction Processing in Bitcoin, https://www.semanticscholar.org/paper/Secure-High-Rate-Transaction-Processing-in-Bitcoin-Sompolinsky-Zohar/728b60c04afb5b87853b59265e49f430dbf631db

Department of Finance, School of Business, University of Connecticut, Storrs
E-mail address: yaacov.kopeliovich@uconn.edu