PoolChain - A Consensus Protocol for Blockchain

Final Report
of the Project in

Advanced Blockchain Technology Course

by

Smruti Ranjan Behera, Saurish Darodkar, Hrithik Mohan
(Roll No. 213050077, 213050046, 180050079)

Supervisors:
Prof. Vinay J Reibero

Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
2022

Contents

Chapter 1

Introduction

1.1 Abstract

PoolChain is a new consensus protocol for blockchain that is a hybrid of Proof-of-Work and Proof-of-stake that inherits the robustness of the former while still saving few resources like the latter without compromising with the mining rewards and security of the system. It uses the stake to reduce the difficulty of Proof-of-Work mining thus making the protocol energy efficient and scalable. Our protocol tries to address the issues that are present in PoW and PoS like Energy wastage, Long Range Attack, Stubborn Mining, Selfish Mining and Initial Distribution Problem. Our main objective is to come up with a protocol which is energy efficient while keeping the mining rewards similarly competitive as with PoW while also addressing major attacks on pure PoS. We introduce a new concept of blockchain miners bidding to create the next block in the blockchain and test its effectiveness as an alternative for the existing protocols.

1.2 Problem Statement

Blockchain is a decentralized database containing linked data blocks that are resistant to modifications. Blockchain's agents or nodes must arrive at a consensus to decide the final state of the system eventually using consensus protocols. Permissionless Blockchain Like Bitcoin uses Proof-of-Work for consensus whereas various others use Proof-of-stake or a variant of it. POW has been notorious for being unsustainable with high ever increasing wastage of energy and electronics. Whereas POS has been marred with security issue right from the beginning. Design a protocol that requires less energy and simultaneously more scalable, secure and robust.

1.3 Background Work

A blockchain is a growing list of records, called blocks, that are linked together using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree). The timestamp proves that the transaction data existed when the block was published to get into its hash. As blocks each contain information about the block before it, they form a chain, with each additional block reinforcing the ones before it. Therefore, blockchains are resistant to modification of their data because once recorded, the data in any given block cannot be altered retroactively without altering all subsequent blocks.

Blockchains are typically managed by a peer-to-peer network for use as a publicly distributed ledger, where nodes collectively adhere to a protocol to communicate and validate new blocks. Although blockchain records are not unalterable as forks are possible, blockchains may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance.Various Blockchain system uses different consensus protocol to arrive at a consensus which will be the next block in a permissionless setting. Proof-of-Work and Proof-of-Stake are the popular ones. (1)(2)

Proof of Work : It's a consensus protocol where miners of a blockchain need to solve a puzzle i.e., nodes must generate a hash that is below a hash threshold by using nonce and previous block hash and some more data. It's the nonce that changes till the solution is achieved. Typically, in Bitcoin on an average $28x10^{18}$ hashes run every second for doing POW mining. That's a lot of energy and investment in ASC.

Proof-of-stake : Proof-of-Stake miners commit stake (money) to create blocks into existence. Probability of creating the next block is proportional to the amount of stakes a miner has in the system.(3)

Hybrid POW/POS: While most PoS protocols are a deliberate departure from PoW, some hybrid consensus mechanisms use elements of both PoW and PoS together to power on-chain operations. In most cases, these hybrid consensus mechanisms (HPoS) rely on PoW miners to generate new blocks housing transactions, which are then passed on to PoS validators, which vote on whether to confirm the blocks and record them to the blockchain's ledger.(5)

There have been various other protocols as well as hybrid of above like slasher but we will be focusing on these two only.

1.4 Objectives

1. **Energy Efficiency** : To come up with a protocol that doesn't waste energy a lot like Proof-of-Work. Since PoS addresses this, our protocol will be along the lines of PoS.

2. **Resistance to Long Range Attack** : Proof-of-Stake is a very good alterative for PoW. PoS is energy efficient. But PoS is prone to Long Range Attack. We want our protocol to be resistant to this attack.

3. **Resistance to Selfish and Stubborn Mining** : We intend to make our protocol resistant to Selfish Mining and Stubborn Mining.

4. **Initial Distribution Problem** : We also want to address this problem which can be seen in cryptocurrencies like Ethereum that uses PoS protocol.

5. **Scalability** : We intend to reduce inter-arrival time between blocks while keeping the forks minimal.

1.5 Report Outline

- In the Protocol Chapter 2, we have explained the our protocol in detail.

- In the Experiments Chapter 4, we described the simulator that we built to check the effectiveness of our protocol, the method of our simulation and presented the Observations we have made in the simulation. This chapter contains all the **Github code** for our protocol simulator.

- In the Vulnerabilities Chapter 5, we listed out the vulnerabilities that are very critical and how our protocol deals with it.

Chapter 2

Protocol

2.1 PoolChain - High Level

- Reduce Difficulty of POW on the basis of stakes
- These stakes aren't on the basis of total stakes but calculated on the basis of bids per block. Otherwise who have large stakes will get inaccessible lead in cornering more blocks forever creating a reinforcing loop.
- Bids for a block is made before some N blocks (say 1000) earlier.
- Bid ratio or stake for a block is decided by bid/ total bids for a block.
- More Bid Ratio -> Lower difficulty -> Easier to Mine
- Less Bid Ratio -> Higher Difficulty -> Harder to Mine
- After Block Creation Bids and rewards blocked for S blocks(say 1000) .
- Any tampering leads to confiscation of bids and rewards.

2.2 PoolChain - Details

- From Genesis Block first N+ blocks would be done by pure POW with a threshold decided upon in genesis block. This N is a system parameter and is depended upon the total currency distribution in the system.
- Let's say N = 1000.

- Assuming there's enough currency distribution by 0th block
- Within first 0-100th block away from 1000th block, miners need to bid on 1000th block for mining against their public key.
- Bidding is a continuous process. Similarly, for 1001th block miners need to bid within 1st – 101st block.
- When 1000th block needs to be mined. All the total bids are calculated.
- For a miner his difficulty is reduced by a factor of bids/total bids.
- When a miner finds a block then they its added to blockchain.

2.2.1 Variable Hash Difficulty

We make use of the bids by the miners as a kind of security to reduce hash difficulty. Miners with higher can be seen as ones with more hashing power which means they will mine the block with high probability.To reduce the energy wastage, we bring in variable hash difficulty instead of same difficulty for all miners. Higher the bid, lower the hash difficulty and easier the block creation. This in turn also decreases the average block generation time.

Difficulty Reduction Ratio (*DRR*), denoted as f, is the fraction of bid by the miner in the total amount (*Jackpot*) bid by all miners for mining that particular block.

$$f = \frac{\text{Amount bid by miner}}{\text{Total bid for target block}}$$

$$h = hash(hash(B_{prev}), Pk, MerkleRoot, Nonce, t) \tag{2.1}$$

where h is Block Hash, B_{prev} is Previous Block, Pk is the public key of the miner/account holding the bids, Merkle Root is Root of all Transactions, Nonce is Random Number and t is timestamp.

$$h \leq Threshold_0 + (Threshold_{max} - Threshold_0) * f_{miner} \tag{2.2}$$

where $Threshold_0$ is Threshold Constant and $Threshold_{max}$ could be any $d \times Threshold_0$.

Similar to PoW, T_0 is time-adjusting. If the hashing power of the network increases, then T_0 decreases. T_0 is adjusted after every n (say 2016) blocks

$$Threshold_0^{new} = Threshold_0^{old} \times \frac{timestamp_{k+2016} - timestamp_k}{2016 \times t_B} \tag{2.3}$$

where t_B is Expected Block Inter-arrival Time. k is kth block.
Note: This above formula is from Bitcoin and this is right now under scrutiny because if inter-arrival time decreases due to reduced POW difficulty it will automatically update threshold to increase the the difficulty in every 2016 block which is undesirable as again miners need to bid more to save on electricity costs and subsequent profits.

2.2.2 Mining Rewards

The mining reward will be proportional to the difficulty of the computation done by the miner who created the block. Higher the creator's bid, lower the hash difficulty and hence lower the mining reward. This reduces Initial Distribution Problem which we analyze in the coming chapters. Block Reward is a function of Block creator's Difficulty Reduction Ratio, f_{bc}

$$\text{Block Reward} = R_0 \times function(f_{block_creator})$$

where f_{bc} is block creator's Difficulty Reduction Ration, R_0 is Reward Constant that halves for every few years.

2.2.3 Block Inter-arrival Time

In comparision to pure PoW, the hashing difficulty is reduced for miners, which reduces the HashRate (Total Number of hashes computed per block created) of the network. This results in reduced Block inter-arrival time.

2.2.4 Release of bids

After a block is mined the mining reward + original bid is blocked for y blocks, after which the bid transactions and rewards are released. Higher the bid/total bids ratio, longer the bid will be locked. This will ensure that attackers can't hijack the blockchain with ridiculously high bid amount and DDoS attacks on other bidders to prevent them from bidding. If bid ratio is 1 it effectively means the bid is blocked for infinite time ultimately means burning the bid.(Under Theoretical Scrutiny but implemented in simulator)

2.2.5 Invalid Blocks

Anyone in the network, if finds out an invalid block, can release a confiscation transaction that makes the block invalid. If the transaction is accepted by the network, then the whistle blower would get the entire bid amount.

2.2.6 Fork Resolution

- Fork Resolution Will be on basis of the heaviest chain rather than longest chain.
- We use the concept of anchors by Ovia Seshadri et al(4).
- Each eligible miner who is also mining on a particular block, manage to create a block that has hash within a particular target space becomes an anchor. This target space definitely has lower difficulty than the original difficulty that the miner is supposed to bid.
- The anchor blocks only points to the previous blocks. The anchor blocks txn are dropped except the coinbase.
- The weight of the chain = Weight of anchors + Weight of blocks.
- So miners mine on the chain that has the highest weight.

2.2.7 Corner Cases

- If there's no bids for a level. In that case an empty block is proposed without the coinbase and the mining process resumes. The empty block will have Zero as public key and remains same no matter who proposes.
- If there's no block creation or miners unable to mine even after bidding. In that case after a certain time interval that is corresponding to Network Time Protocol an empty block is created without any coinbase.

Chapter 3

Analysis

3.1 Optimal Strategy for miner

Among the window of 100 blocks that are allowed to bid at a given time, Miner will have the best chance to mine a block if he bid on a block that has the lowest amount of bids so far. Also, to maximize his chances to get that block, he should put his entire stake on a single block. If all miners bid on the block having a minimum total stake, at equilibrium, stakes of the network are uniformly distributed over the available blocks. Also, it is wise to bid on a block as soon as you have money to bid on, to maximally utilize the resources.

Since the set of miners bidding on the same block has their stakes locked up for almost 1000 blocks and all their stakes are released at the same time, (except for the winner, The winner's stake is released a bit later in the network), there are implicit pools of mining are created. Each pool will mine every 1000_{th} block hence there will be 1000 pools in the network. Each pool will have almost the same amount of stake. Due to the fact that the winner's stake will be released a bit later, there will not be a dominance of any single party in a single pool. This helps to reduce the pool hijack issue.

Say there are N miners which mine on each block, say threshold for them be T. We state energy consumption as number of hashes calculated. In each round Pure POW will calculate $N \times \frac{2^{255}}{T}$ hashes. Say total bid on hash is S and bid of the i^{th} miner is S_i.
For miner i, Threshold = $\frac{2^{255}}{S} \times S_i$
Say total N' miners bid on a block and assuming uniform distribution among the bids, S = $N' \times S_i$

$$\text{Threshold T} = \frac{2^{255}}{N'} \text{ for each miner.}$$

$$\begin{aligned}\text{Total hash calculated for a round} &= N' \times \frac{2^{255}}{T} \\ &= N' \times \frac{2^{255}}{\frac{2^{255}}{N'}} \\ &= (N')^2\end{aligned}$$

The proposed value of N' = $\frac{N}{1000}$.
So in our protocol, total hash calculated are $\frac{N^2}{10^6}$.
Our protocol consumes less energy when $\frac{N}{10^6} < \frac{2^{255}}{T}$.
Currently bitcoin has around 10^6 miners. Hence our protocol is $\frac{2^{255}}{T}$ times less energy consuming as compared to bitcoin on the current bitcoin scale.

Chapter 4

Implementation, Experiments & Observations

4.1 Implementation

4.1.1 Simulator

We have created our Simulator in python with full functionality of Proof of Work and merkle tree implementation. We have tried to make it as realistic as possible. We have successfully implemented the code and have run simulation of this protocol without bug.
The github code is available here: **Github Code Repository**

4.1.1.1 Instruction to run Simulator

- main.py needs to be run to initiate the simulator. At the end of program runtime full results will be shown.

- global_functions.py contains all the system parameters.
  To run the simulator in PoolChain mode make POB=True whereas for pure POW POB=False.
  To adjust number of nodes NODES value should be changed. To create more blocks adjust TXN_NUM.
  To adjust N (the time till which the amount will be blocked after successful mining)
  Note: Here N is constant but in our protocol it varies as per bid ratio.

- blockchain.py contains all the files related to blockchain.

Table 4.1: Inter-arrival Time and Energy Consumption per block

Number of Nodes	PoolChain		Pure POW	
	IAT	*Energy Consumed*	*IAT*	*Energy Consumed*
10	8.25	0.14	11.74	0.16
20	6.78	0.183	9.65	0.29
50	6.6	0.38	7.52	0.711

Note: IAT is the inter-arrival time between two blocks

- block.py contains the block header formats.
- mkltree.py is the file for Merkle tree.
- proof.py is for Proof of Work.
- node.py contains code for node behaviour.

4.2 Experiments

We experimented with different values of nodes and also compared with pure POW and with PoolChain. Table 4.1 contains all the data output from simulator.

4.3 Observations

Our Experimental results in Table 4.1 show that there is significant saving in energy consumption and improvement in inter-arrival time. This improvement is quite proportional to the number of nodes in the network. More number of nodes meant huge savings in energy but not that significant improvement in inter-arrival time. But still inter-arrival time was better than pure POW.

Chapter 5

Study on Vulnerabilities

We have carried out following theoretical study on Vulnerabilities as reported in proposal. This hasn't been implemented yet as per proposal this will be executed by Checkpoint 2.

5.1 Nothing at Stake Problem

PoS has one serious problem. If there is a fork in the blockchain, the rational behaviour for all miners is to mine on both the branches which hinders the consensus in the network. This makes it easy to perform double-spending or other sorts of attacks relying on forking the blockchain. As long as users of the network believe the attack may succeed, the will support it my mining on top of attacker's branch. While forking attacks will be opposed by users with a large amount of currency who will fear to lose their money, if currency is evenly distributed among many users, the attack is more likely to succeed.

Study of NOS Problem on our Protocol: The addition of POW into our protocol forms effective protection against this type of attack as there will be always energy consumption for block creations. It can't be effortless like PoS. Even if the difficulty is reduced to create new blocks but still the reduction is not that much like in slasher protocol where difficulty is about 1% of original POW.

5.2 Initial Distribution Problem

In PoS, the coin balance determines the profit of the user. Thus, there is always a concern that the initial holders of the coins will not have incentive to release their coins to third parties. To deal with this, PoS implementations use additional algorithms to distribute initial wealth evenly.

Study of IDP Problem on our Protocol: By trying to accumulate more coins thus increasing its chances of block formation miner tries to hoard maximum coins possible. This is mostly eliminated in our protocol by calculating stake as per block bids basis rather than entire system. Because to create blocks miners must utilize their coins by biding on a block and if block is created then their coin is blocked for an extended period of time. To mine on a block electricity is consumed even if its less than normal POW.

5.3 Long Range Attack

In PoS protocol, attacker with considerable computational power can build an entirely new branch starting from the first block since creating a new block doesnt take much computation. To prevent a long range attack, the protocol can specify the maximum allowed depth of a branching point. But this restriction doesnt solve the problem for new users. When a new user connects to the network, he sees multiple blockchains with no prior knowledge of their authenticity. If attacker's blockchain is preferable to the valid blockchain, new users will adopt it instead.

Study of LRA Problem on our Protocol: By creating a separate chain where the attacker only includes its own bids , it gain complete advantage over others. Ultimately the miner catches up with main chain and reveals the longest chain and others start mining on the attacker's chain. In order to block this behaviour, we introduce a concept where the bids are blocked for x blocks where x is proportional to bid/total ratio and a lower bound on bid amount. If there is only one bid then bid/total ratio becomes one even if its just a small bid. This would give any attacker ease of creating new blocks. But in our protocol, this is prevented as too high ratio can effectively block bids for a long time thus preventing the attacker to gather any new coins to further create new blocks. To prevent attackers to get away with this we have a put a lower bound on the block bids so that attacker can't create new blocks with extremely small bids.

5.4 Double Spending Attack

In this the attacker offers bribe for building on top of truncated blockchain that doesn't include a payment transaction he has already done. Users participating in the attack lose nothing if the attack fails. For the attacker, it is profitable as long as total bribe is less the value of the transaction.

Study of DSA Problem on our Protocol: Our protocol incorporates variety of methods to deal with the issue. By blocking the bid for an extended period of time it gives natural protection against DSA. If miner creates inavlid attacks bid is forfeited. Also other nodes won't accept any blocks that has invalid transactions. Nakamoto-style protocols like our protocol are usually not vulnerable to this attack.

5.5 Large Bid Attack

In this attack, an attacker can bid with large amounts will get a high chance to mine the next block.Thus an attack with large currency in the network might create a large proportion of the blocks where he might include many malicious transactions thus bringing down the entire network.

Study of LBA Problem on our Protocol: In order to block this behaviour, we introduce a concept where the bids are blocked for x blocks where x is proportional to bid/total ratio and a lower bound on bid amount. Higher the bid in comparison to others longer the bid amount will be blocked. If there is only one bid then bid/total ratio becomes one even if its just a small bid or large bid. This would give any attacker ease of creating new blocks. But in our protocol, this is prevented as too high ratio can effectively block bids for a long time thus preventing the attacker to gather any new coins to further create new blocks.

5.6 Selfish Mining

In this attack, miner doesn't immediately publish a newly mined block. When the rest of the network is about to catch up with the miner, the blocks that are hidden would be released into the network. The selfish miner makes sure that their chain is longer which makes the rest of the network mine on their blockchain.

If the attacker will keep his chain secrets he would have to ensure that his bids remain higher for consecutive blocks which can never be guaranteed as others can bid on the same block. So selfish mining can't as straight forward as in Pure POW. Also higher bids mean bids get locked for longer time. This is done purely to stop these types of attacks.

5.7 DDoS Attack

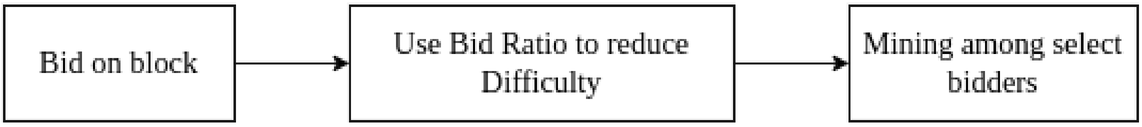
The main DDoS threat in blockchain is Transaction flooding. Attacker can send many transactions to the network and fill up the blocks with spam transactions while legitimate transactions aren't included in the blocks. But various forms of DDOS Attack can possibly occur in our protocol. For example the miners can form a pool to mine a range of blocks and deliberately not include any bidding transactions of other bidders denying access to mine block by non attackers. But this is extremely non-trivial to do because in order to prevent this we have made the window of bidding transaction blocks to be 100. Attacker have to consecutively mine for 100 blocks to successfully carry out this attack which can be near impossible to achieve given POW involved in this protocol. Also we plan to share the mining reward with the miner that has included the bidding transaction. This will incentivise the early miners to include bidding transactions hoping if one of that succesfully mines a block. And Network DDoS attack can be solved by using Tor Network or various IP hiding facilities.

Chapter 6

Discussion

- **Is this same as Vanilla POS except that they have only the bid amount per block is considered at stake rather than the entire stake of a user as in vanilla POS?**

  Ans: No , its not equivalent. The above statement itself can be categorised as another POS protocol for whose discussion is beyond the scope of this report. But this forms the foundation for our protocol on top of which we have added Proof of work onto it to make it more robust. **Detailed Explanation:** One of the biggest issues of Vanilla POS is the Nothing at stake problem; it is difficult to converge around a single chain when multiple competing chains exist. With PoW, the accepted chain is the one with the most work behind it, and miners are penalized for mining on a chain that will get rejected due to the wasted resources. With PoS, however, there's nothing to prevent nodes from staking on multiple chains. In PoS If a miner has a large number of coins, the miner will want to oppose attacks to preserve the value of their own coins; in an ecosystem with small miners, however, network security potentially falls apart in a classic public goods problem as no single miner has substantial impact on the result and so every miner will act purely "selfishly". By combining POS and POW we can get the best of both worlds but not necessarily inheriting the problems of both the Protocol. In Vanilla POS the stake of the miner in entire system contributes to block formation. This leads to two major problems (but not limited to) . Initial Distribution problem: and long term attack. By trying to accumulate more coins thus increasing its chances of block formation miner tries to hoard maximum coins possible. This is mostly eliminated in our protocol by calculating stake as per block bids basis rather than entire system. Because to create blocks miners must utilize their coins by biding on a block and if block is created then their coin is blocked for an extended period of time. Long-term attack : By creating a separate chain where the attacker only includes its own bids , it gain complete advantage over others. Ultimately the miner catches up with main chain and reveals the longest chain and others start mining on the attacker's chain. In order to block this behaviour, we introduce a concept where the bids are blocked for x blocks where x is proportional to bid/total ratio and a lower bound on bid amount. If there is only one bid then bid/total ratio becomes one even if its just a small bid. This would give any attacker ease of creating new blocks. But in our protocol, this is prevented as too high ratio can effectively block bids for a long time thus preventing the attacker to gather any new coins to further create new blocks. To prevent attackers to get away with this we have a put a lower bound on the block bids so that attacker can't create new blocks with extremely small bids. Nothing at stake Problem: The addition of POW into our protocol forms effective protection against this type of attack as there will be always energy consumption for block creations. It cant be effortless like PoS. Even if the difficulty is reduced to create new blocks but still the reduction is not that much like in slasher protocol where difficulty is about 1% of original POW.


Bid on block
Use Bid Ratio to reduce Difficulty
Mining among select bidders

Main objective of our Protocol: Reduce number of miners mining per block to save energy but still not compromising on mining fees or income Use stake/bids as an indirect tool to achieve lower computations per block.

- **As the merkle root can be changed arbitrarily? So h is easily made less than any threshold?**

  Ans: In PoS merkle root of current block or nonce is not included as the block creator can only mine with limited number of timestamps. Including merkle root or any nonce/random number will increase its chances of creating the block. In our case there is actual mining like in PoW but with reduced difficulty. Even if merkle root is changed it won't affect much other than serving the purpose of just another combination of nonce/random number just like in bitcoin. It's still takes fair amount of computations to achieve the solution even with reduced difficulty.

- **What Value t can take? Ans:** It follows the median past time (MPT) rule in bitcoin, the timestamp t must be higher than the median of the past 11 blocks. Also the timestamp cannot be more than 2 hours in the future. Unlike vanilla PoS it has little significance in achieving the solution of the block creation puzzle other than being just another random number. The very purpose of timestamp is to prevent Time Warp Attack and double spending attack in future. As per Nakamoto, "For our purposes, the last transaction is what counts, so we won't mind other subsequent double-spending attempts." The timestamp seals the fate of the transactions that is valid at the moment of block creation.

- **Is one Large bid equivalent say M equivalent to 'N' bids of $\frac{M}{N}$? If not is this a problem? Not clear why this will prevent DDOS attack?**
  Ans: Lets Say One bid amounts to 0.1 of total bids. And 5 bids of 0.1 equal to 0.5 of total bids. And initial threshold is 100 if there were no bids and 200 is the maximum threshold allowed as per protocol is 200. So a 0.1 bid will result in a threshold of 100 + (200-100) x 0.1 = 110. And 0.5 bid will result in threshold of 150. So all 5 small bidders must have hash lower than 110 and large bidder must have hash less than 150. As per our analysis getting a hash below 110 will require exponentially higher computations than getting a hash below 150. It won't be same as 5 small bids of 0.1 equal to one large bid of 0.5. So, no one large bid equivalent say M is not equivalent to 'N' bids of $\frac{M}{N}$. Also this will result in attackers pooling the bids and sidestep the variable bid-block time introduced earlier.

Chapter 7

Conclusion And Future Work

- We reformulated the blockchain consensus protocol by integrating Proof-of Work with Proof-of-Stake.
- We successfully demonstrated how PoolChain alleviates all the shortcomings of Proof of stake and more energy efficient than Proof of Work.
- This makes the blockchain far more secure, sustainable and robust than either of these consensus protocols
- We would like to see how it works in a Sharding setting in further works.

Chapter 8

Presentation Video And Code

Presentation Video :

https://drive.google.com/file/d/1_HQRoegV5aGc0mLI7EXwFV1p5zU9efNw/view?usp=sharing

Github Code:

https://github.com/SmrutiRanjan-Ai/PoolChain-Blockchain-Consensus-Protocol

References

[1] Wikipedia Article on Blockchain URL: https://en.wikipedia.org/wiki/Blockchain

[2] Nakamoto, Satoshi, and A. Bitcoin. "A peer-to-peer electronic cash system." Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf 4 (2008).

[3] Deuber, Dominic, Nico Döttling, Bernardo Magri, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. "Minting mechanism for proof of stake blockchains." In International Conference on Applied Cryptography and Network Security, pp. 315-334. Springer, Cham, 2020.

[4] Seshadri, Ovia, Vinay J. Ribeiro, and Shadab Zafar. "Securely Improving Stability and Performance of PoW Blockchains Using Anchors." In 2022 14th International Conference on COMmunication Systems NETworkS (COMSNETS), pp. 147-155. IEEE, 2022.

[5] Hybrid Proof of Stake URL: https://www.gemini.com/cryptopedia/proof-of-stake-delegated-proof-of-stake-consensus-mechanismsection-hybrid-proof-of-stake

[6] Sengupta, Jayasree, Sushmita Ruj, and Sipra Das Bit. "A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT." Journal of Network and Computer Applications 149 (2020): 102481.

[7] Neu, Joachim, Ertem Nusret Tas, and David Tse. "Two Attacks On Proof-of-Stake GHOST/Ethereum." arXiv preprint arXiv:2203.01315 (2022). Wesley, Massachusetts, 2nd ed.

[8] Wang, Xuechao. "Proof-of-stake longest chain protocols: security vs predicability." PhD diss., 2020.