Novel Consensus algorithm for Blockchain using Proof-of-Majority (PoM)

Gorla Praveen*, Siddharth Pratap Singh*, Vinay Chamola and *Senior Member*, IEEE and Mohsen Guizani *Fellow*, IEEE

***Abstract*—The popularity of Blockchain is rising on account of its far-reaching applications in diverse industries. However, in the recent past, the blockchain has seen a rise of energy extensive mining pools which is leading to centralization, contradicting the basic blockchain tenet of decentralization. This letter proposes a novel consensus algorithm using Proof of Majority (PoM), to increase decentralization and to eliminate resource-intensive tasks. We have evaluated the proposed algorithm in terms of latency and throughput. The proposed consensus algorithm outperforms popular existing consensus algorithms.**

***Index Terms*—Blockchain, Proof of Majority, Decentralization, Consensus algorithm and Peer-to-peer computing**

I. Introduction

Since the introduction of blockchain as a conceptual framework in enabling decentralization, it has gained momentum as futuristic technology [1] that can enable public participation, trust and consensus. In attaining such participation, public blockchain networks are seen as an approach to improve their accessibility to majority of the citizens to achieve transparency, reliability, immutability and security [2] in public governance [3], [4]. However, there exists multiple issues with the current form of the technological development and deployment of blockchain as a decentralized systems. One of the major issues is that chains being more resource intensive and other is non-Democratic architecture of the system. The most popular blockchain protocols such as Proof-of-Work (PoW), Proof-of-Stake (PoS) and Practical Byzantine Fault Tolerance (PBFT) suffers from one or more problems of resource intensiveness, high stake based prioritization and imposing heavy network loads [5], [6]. In addition, with the constant increase in computational intensiveness to mine the blocks, mining tends to increase the complexity leaving the individual participation of the nodes with moderate resources. In support to this, individuals joining the mining pool and forming mining server farms establishes the control on majority of the blockchain operations by a group of individual nodes. In overall, due to inefficiency and lack of incentives with high variance in mining for single entity nodes, there has been an active decline in both the participation and number of transactions performed by them. This is seen as trend of imposing the centralization in blockchain in contrary to its fundamental concept of being equal opportunity aware decentralized system [7], [8]. The studies in [9], [10] had put forth significant effort in proposing the majority based consensus through establishing the trust based validation and consortium of private chains. However, the scope for decentralized and majority consensus based blockchain based transaction is partially achieved or limited to the consortium specific. And in most of the applications of the blockchain, as a decentralized system, it poses a major Byzantine Generals Problem. In such systems, the peer-to-peer communication with malicious node attacking the system of nodes leads to change of information, both in time and value. In such scenarios, mining nodes needs to distinguish the tampered information and be in synchronous with the other nodes in the chain. However, this needs a corresponding algorithmic design to achieve consensus algorithm. Some of the existing blockchain protocols solves the Byzantine Generals Problem using the Proof of Work consensus algorithm and blockchain [11], [12]. They establishes a single source of truth using Proof of Work consensus and considers the longest chain as truth [6], [13]. The longest chain represents the sequence of events and most work done. A powerful adversary who controls the majority of the hash power of a blockchain network can outpace honest nodes and create the longest chain.

Our work tends to democratise blockchains with each node having equal eligibility to vote. The conceptual framework behind the Proof-of-Majority (PoM) consensus algorithm lies in the effectively considering the majority of the decision as the true to the transaction decision. PoM removes the entry barrier for potential new miners by significantly reducing resource requirements. PoM based blockchains allow nodes to become miners without providing any resource-intensive task.

II. Methodology

This section presents our proposed consensus algorithm for Blockchain using Proof-of-Majority (PoM) with improved decentralization and democratization of network participation. We first discuss the process of a PoM based blockchain network; after that, we present the block structure in the PoM network, then we introduce the hash voting mechanism and finally miner reward in the network. The Mining in PoM based chains constitutes of the following processes.

* G. Praveen and P. Singh have equal contribution

The work of Vinay Chamola was supported by DST-SERB [Science and Engineering Research Board (SERB)] funding under Grant ECR/2018/0001479. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro P5000 GPU to Dr. Vinay which was used for this research.

Praveen Gorla, Siddharth Pratap Singh and Vinay Chamola are with the Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science Pilani, Rajasthan-333031, India. (e-mail: p20190028@pilani.bits-pilani.ac.in, f20180392@pilani.bits-pilani.ac.in, vinay.chamola@pilani.bits-pilani.ac.in)

Mohsen Guizani is with Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi 51133, (e-mail: mguizani@ieee.org).

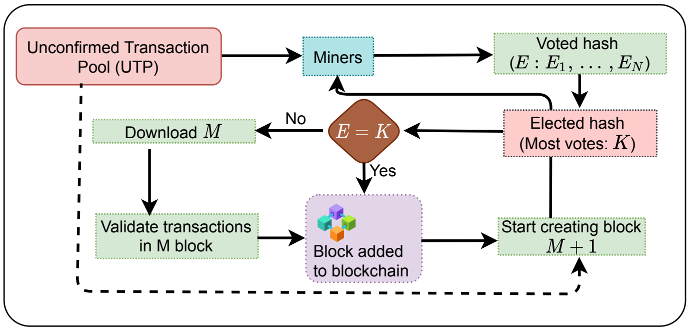

Unconfirmed Transaction Pool (UTP)
Miners
Voted hash $(E : E_1, \ldots, E_N)$
Download M
No
$E = K$
Elected hash (Most votes: K)
Yes
Validate transactions in M block
Block added to blockchain
Start creating block $M + 1$

Fig. 1: Overview of the framework

- Listening for transactions and adding them to the unconfirmed pool of transactions.
- Pre-validating the transactions for forming the block.
- Miners in the network share the final block hash (candidate hash).
- Listening to the candidate hashes from other miners.
- Finding the hashes with the most votes, with a scope of representing it as an elected hash.
- Validating if the elected hash and voted hash are the same and adding the block to the chain.
- If validation results in the voted hash and the elected hash being different, the miner starts requesting the elected block from a node on the network.

The conceptual framework behind the PoM consensus algorithm lies in the network consideration of the majority of the miners voting for the candidate block. It eliminates the conceptual framework of the resource and stake intensive mining taking control of the network. In our framework, we consider that a minimum of 90% of the participating nodes should vote for the candidate block in each cycle. $i.e$ at least 90% of the nodes vote for the candidate hashes for a successful mine cycle. [14]

A. Block structure

PoM chains have a similar block structure as the general-purpose blockchain consisting of the hash of the previous block, ensuring the authenticity of the block chain in the event of alteration. The block structure of the PoM includes Height-index of the block, Version number - the version of the chain, Chain ID - ID which uniquely identifies the chain, Previous block, Hash, TimeStamp - Mining epoch time of the block and Transaction data with a size of 133 Bytes. For the PoM chain, the fundamental block transaction structure is the adaption of the bitcoin structure with varied consensus in the network.

In the network, if block size reaching to the $x\%$ of the capacity within the encapsulated block interval time, then the $100 - x\%$ of the nodes on the network would be unable to receive the blocks as they arrive leading to the effectively disabling of the node. Considering this fact, it is desired to maintain nearly the current degree of decentralization, as measured by the active functioning nodes in the peer-to-peer overlay network. For our work we target a minimum threshold of block reaching to the 90% of the total nodes. The threshold minimum achievable probability of the block reaching to nodes is 0.9. Considering the 5G and Beyond use cases, and its achievable speed across the networks, a minimum cycle time of 12s [14] is evaluated to be needed for a block with a size of 800 KB to be propagated for 90% of the nodes. Considering the general purpose block transaction structure, the blockheader size is 80 bytes, the size of blocksize(size of the block) is 4 bytes and transaction counter size of 3 bytes. And the Size of coinbase transaction is 65 bytes and rest other transactions have average size of 61 bytes [15]. Then the transaction space T_s of the blockchain network and Total number of transactions in each block N_b is as follows:

$$T_s = 800\,\text{KB} - (80 + 4 + 3)KB = 7,99,913\,\text{bytes}\,\text{(SI)},$$

$$N_b = \frac{T_s - 65}{61} + 1.$$

The transaction carried out per second (T_{ps}) is $\frac{N_b}{12} = 1092$.

B. Consensus Voting Mechanism

Each miner broadcasts the candidate hash to all the other nodes in the proposed consensus voting mechanism. While this happens, this particular node would continue to listen to the candidate hashes from other miners. The miner would vote a candidate hash at the end of the fifteen-second time cycle. Here

Algorithm 1 Voting mechanism in the blockchain

```
Input: $N$, $M$, and $K$
▷ $N$ is the total number of nodes in the network.
▷ $M$ represents the $M^{th}$block that is currently under the process of mining.
▷ K is the elected hash.

Initialization: voteMap = new Map
i = 1
while $i \leq N - 1$ do
▷ $h_i$ is the candidate hash received from a peer node i
    voteMap[$h_i$] += 1
    i++
end while
i = 1
maxCount = 0
electedHash = NULL
while $i \leq lengthof(voteMap)$ do
▷ $g_i$ is the candidate hash in the voteMap in the $i_{th}$ position
    if $voteMap[g_i] \geq maxCount$ then
        maxCount = voteMap[$g_i$]
        electedHash = $g_i$
    end if
    i++
end while
K = electedHash
if K == E then
    Add the candidate block to the chain
else
    Request for $M^{th}$ block from a peer in the node
end if
Start Mining $M + 1^{th}$ block with the elected hash
```

in this step, to be more clear, the miner forms a block after validating all the transactions and then sends this block hash to all other nodes. If the elected block matches the miner's voted block hash, the miner adds the block into the chain. If the elected hash does not match the voted hash, the miner proceeds with the mining process with the previous hash for the new block as the current elected hash. Meanwhile, the miner would simultaneously request and download the block contents from another node if the voted and elected hash did not match.

Let there be total N nodes in the network with E and Krepresenting elected hashes and candidate hashes, respectively, with $M-1$ number of blocks in the current chain. Then, the Algorithm 1 refers to the mechanism of voting in the proposed blockchain to attain the consensus.

As shown above if the elected hash is not equal to voted hash then the miner would request for the new block from a peer in the node. The miner meanwhile would continue to listen for transactions and continue to mine the next block with elected hash as the previous block hash.

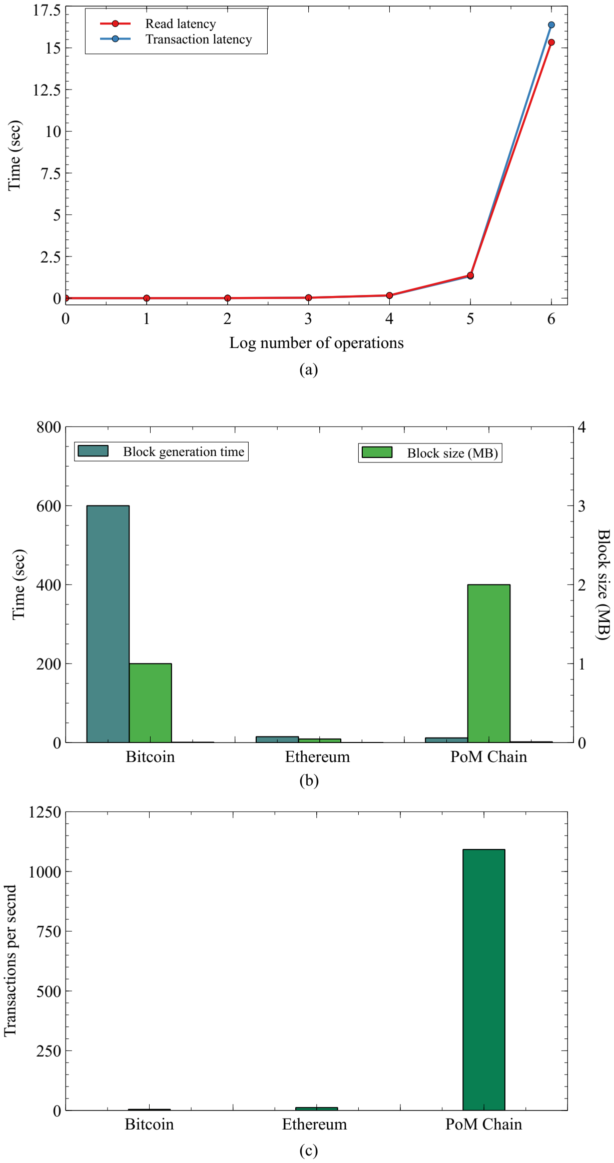
C. New miner entry to the network

Nodes in the network form a random graph network. When a node joins the network, it queries the number of DNS servers. These DNS servers are run by decentralized entities and return a random set of bootstrap nodes currently participating in the network. Once connected, the joining node learns about other nodes by requesting their neighbours for known addresses and listening for the spontaneous broadcasting of new addresses. When a new node wants to join the network, it is essential to identify the correct version of the chain that is agreed by the majority of the network nodes. To achieve this, a new node scans the network and keeps a record of the last block's hash. The last block's hash with the most frequency would be a correct hash. The miner can then request the full version of the chain from any node on the network with the correct version. Considering N number of current nodes in the network, the miner association framework as described in Algorithm 2 can be used to determine the correct version of the chain when a new node tries to enter the network.

Algorithm 2 Miner association in the network

```
Input: $N$            ▷ $N$ is the total number of nodes in the network.

Initialization: freqMap = new Map
i = 1
while $i \leq N$ do
    freqMap[i] += 1
    i++
end while
maxCount = 0
correctChainIndex = -1
while $i \leq N$ do
    if $freqMap[i] \geq maxCount$ then
        maxCount = freqMap[i]
        correctChainIndex = i
    end if
    i++
end while
```



(a)

(b)

(c)

Fig. 2: (a). Read and transaction operational latency,(b). Block generation time and Block size, and (c) Transactions throughput of blockchain protocol.

Miner Reward: Miners are rewarded for mining the correct block (elected block) after determining its truthiness of mining. Once the correct M^{th} block is mined, its rewarded information is included in the $M+1^{th}$ block. Ideally, all the miner who vote for the elected hash is rewarded equally in the network. The value of the mining reward R_{m+1} for the each miner is as follows:

$$R_{m+1} = \frac{R^b}{N}.$$

III. Performance Analysis

In this section, we present an experimental simulation setup to evaluate the performance of the proposed PoM framework for blockchain. The performance analysis of the setup comprises the system latency, transactions throughput and block generation time estimated over deploying the framework on multiple mining nodes. Transaction Latency is a network-wide view of the amount of time taken for a transaction's effect to be usable across the network, i.e. the amount of the time from the point that it is submitted to the end that the result is widely available in the network. This includes the propagation time and the settling time induced due to the consensus mechanism in place, performed on a eight nodes in a simulation environment. The metrics of the proposed work is compared with the existing Blockchain frameworks studied in the literature.

Figure. 2(a) plots the read operation latency and the transaction operation latency. As seen from the graph, it is observed that as the number of operations in the network increases, there is an exponential growth in the read and transaction latency. On closely examining, it is observed that transaction and read latencies are fairly increasing at a low rate for up to 10^4 operations and rises very steeply from 10^5 number of operations. This indicates the ultra-low latency and low latency performance of the proposed algorithm for network operations within 10^4 and 10^5 operational count, respectively.

Furthermore, we have shown the performance comparison between the existing blockchain protocols and proposed PoM based consensus algorithm in Fig. 2(b) and Fig. 2(c).
Comparison of PoM chain with different chains can be done on the basis of different parameters such as TPS, Block generation time and Block Size. Fig. 2(b) plots the comparison of the block generation time and the block size for the Bitcoin, Ethereum and the proposed PoM. The figure shows that the average block generation time of Bitcoin is significantly higher than Ethereum, which is relatively higher than the proposed PoM chain. However, the block size (*in MB*) of the proposed PoM algorithm is relatively higher than the Bitcoin and Ethereum due to the incorporation of the information related to the majority of the consensus of the accepted nodes in the network. Fig. 2(c) refers to the comparison on the Transaction performed in unit between the different blockchains. On evaluating the performance of the proposed PoM network, the transactions per unit time is steeply higher than the Bitcoin and Ethereum network.

On observing the performance analysis it is clearly evident that PoM chain has higher TPS than current chains with a fairly low block generation time. We can also compare PoM chain with existing blockchain networks on the basis of decentralisation, unlike other chains PoM is truly decentralised with each node having equal voting power. On the other hand, it also evident that the other existing blockchain networks using PoW or PoS, the controlling power on the network lies with the miner having dominant resources, and hence by virtue they cannot be a considered as true decentralised and democratic in node participation.

IV. Conclusion and Future Works

We presented a novel consensus algorithm to increase the decentralization of blockchains using Proof of Majority, with support for up to 1092 transactions per second. The inclusion of more miners will enhance decentralization leading to the democratized blockchain. Initial results from POC suggest higher throughput, increased decentralization, and reduced entry barrier than blockchains with Proof of Work or Proof of Stake algorithm. The future scope of this work lies in implementing and deploying additional nodes across various geographical locations to achieve very near real-time performance and exploring use cases of PoM in other domains.
However, the PoM consensus algorithm puts a significant network load when nodes are more than twenty. Hence, the future work considers the network architecture by creating a group of nodes that conduct elections to decide the leader, significantly decreasing the network payload. Multiple levels of leaders can be used to make their decision based on the majority of inputs received from nodes.

References

[1] D. Xenakis, I. Zarifis, P. Petrogiannakis, A. Tsiota, and N. Passas, "Blockchain-driven mobile data access towards fully decentralized mobile video trading in 5g networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

[2] B. Hamdaoui, A. Rayes, N. Zorba, L. Song, and C. Verikoukis, "Blockchains for scalable iot management, access, and accountability," *IEEE Network*, vol. 34, no. 1, pp. 6–7, 2020.

[3] V. Hassija, S. Zeadally, I. Jain, A. Tahiliani, V. Chamola, and S. Gupta, "Framework for determining the suitability of blockchain: Criteria and issues to consider," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 10, p. e4334, 2021.

[4] K. Lei, M. Du, J. Huang, and T. Jin, "Groupchain: Towards a scalable public blockchain in fog computing of iot services computing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 252–262, 2020.

[5] U. Bodkhe, D. Mehta, S. Tanwar, P. Bhattacharya, P. K. Singh, and W.-C. Hong, "A survey on decentralized consensus mechanisms for cyber physical systems," *IEEE Access*, vol. 8, pp. 54 371–54 401, 2020.

[6] M. Kaur, M. Z. Khan, S. Gupta, A. Noorwali, C. Chakraborty, and S. K. Pani, "Mbcp: Performance analysis of large scale mainstream blockchain consensus protocols," *IEEE Access*, vol. 9, pp. 80 931–80 944, 2021.

[7] A. Beikverdi and J. Song, "Trend of centralization in bitcoin's distributed network," in *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2015, pp. 1–6.

[8] D. Xenakis, A. Tsiota, C.-T. Koulis, C. Xenakis, and N. Passas, "Contract-less mobile data access beyond 5g: Fully-decentralized, high-throughput and anonymous asset trading over the blockchain," *IEEE Access*, vol. 9, pp. 73 963–74 016, 2021.

[9] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Transactions on Services Computing*, vol. 12, no. 3, pp. 429–445, 2019.

[10] J.-T. Kim, J. Jin, and K. Kim, "A study on an energy-effective and secure consensus algorithm for private blockchain systems (pom: Proof of majority)," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 932–935.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[12] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the Works of Leslie Lamport*, 2019, pp. 203–226.

[13] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 2567–2572.

[14] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, *et al.*, "On scaling decentralized blockchains," in *International conference on financial cryptography and data security*. Springer, 2016, pp. 106–125.

[15] E. Georgiadis, "How many transactions per second can bitcoin really handle? theoretically." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 416, 2019.