Nat. Volatiles & Essent. Oils, 2021; 8(5): 1794 - 1812

NVEO
Natural Volatiles & Essential Oils

Understanding and Analyzing Consensus Algorithms for Blockchain

Nipun Bansal, Mrinal Singhal, Mohak Rastogi, Lakshay Arora

1Department of Computer Science & Engineering, Delhi Technological University, Delhi, India
nipunbansal@dtu.ac.in

2 Department of Computer Science & Engineering, Delhi Technological University, Delhi, India
mrinal_bt2k16@dtu.ac.in

3Department of Computer Science & Engineering, Delhi Technological University, Delhi, India
mohak_bt2k16@dtu.ac.in

4Department of Computer Science & Engineering, Delhi Technological University, Delhi, India
lakshayarora1998@gmail.com

Abstract – Over the past decade, ever since the advent of Bitcoin, the first cryptocurrency to enter the market and amid growing privacy concerns in the digital space, blockchain as a technology has grown immensely in popularity as a means of decentralized networking. A consensus algorithm, which in a decentralized network acts as a means for the nodes to arrive at an agreement on the state of data and any changes made to it, forms an integral part of the blockchain technology. However, with the increasing number of these algorithms, there is an urgent need to study them in a systematic way, which will enable us to select an algorithm suited to our needs. In this paper, we study and analyze 17 consensus algorithms and compare them on various parameters like energy requirement, scalability, specialized hardware requirement, etc. and present our findings in a tabular form. We further present our observations on the suitability of the consensus algorithms.

Index Terms – Blockchain technology, consensus algorithm, cryptocurrency

1. INTRODUCTION

The blockchain technology has emerged as one of the biggest disruptive innovation in the field of computing since its inception in 2008. As the difference between the real and digital world shrinks at an unprecedented rate, personal data has become a highly valuable commodity in recent times. Therefore, to ensure its privacy and prevent its misuse by the government or private organizations, blockchain has emerged as a leading solution.

The blockchain is essentially a distributed and decentralized ledger consisting of immutable records. As the name suggests, it is designed as a chain of blocks where each block holds a list of verified transactions and a block header containing a hash value. The links in the chain are maintained by storing in each block's header, the hash value of the preceding block. When any node on the blockchain network has to make a new transaction, its validity is checked by other nodes. Only if it is a valid transaction is it inserted into the block. Any update to a block must be communicated to other nodes so that their local copies keep up to date. However, problems arise when two or more nodes try to broadcast the update

simultaneously. This is the problem that the consensus algorithms address. A consensus algorithm is a method for the nodes in a distributed network to arrive at a consensus or agreement on the change of data.

The consensus algorithm constitutes an essential component of any blockchain network. However, currently, there are over 50 individual consensus algorithms with more are being added every year. Moreover, each has its own sets of requirements and particular characteristics like energy requirements, speed of consensus, etc. which makes them applicable in very specific scenarios. This makes it very difficult to study the algorithms in a systematic way and for individuals or organizations to select a algorithm that will suit the needs of their particular blockchain project. This is the problem that we seek to address, and through this paper, we present a comprehensive review and analysis by studying and evaluating some of the common consensus algorithms.

The rest of the paper is organized as follows: In section II, we give a brief overview of 17 consensus algorithms along with their merits, demerits and applications. In section III, we analyze the algorithms and present our observations. In section IV, we compare the algorithms on the basis of certain criteria. Finally, section V presents the conclusion of our paper.

2. RELATED WORKS

Mingxiao *et al.* [34] reviewed several consensus algorithms along with their key principles, characteristics and performance. They also analyzed the different application scenarios of these consensus algorithms. They also put forward the point that with the research on these consensus algorithms being still in its infancy, it will take some time for them to be specifically designed for different application scenarios.

Bach, Mihaljevic, and Zagar [36] provided a comparative analysis on some common consensus algorithm and their present-day versions that are presently being used in current blockchains. Their analysis primarily focused on the steps involved in each of the consensus algorithms, their scalability, their method of rewarding validators as well as the security risks involved in these algorithms. They came to the conclusion that the Proof of Work consensus algorithm would ultimately be replaced by modern consensus algorithms[37,38].

Shahaab, Lidgey, Hewage, and Khan [17] reviewed and mapped around 65 consensus algorithms for both public as well as private distributed ledger technologies (DLTs). Their paper focused on the public sector and brought attention to potential consensus algorithms. They listed these algorithms across some basic properties. They also proposed that there is not one consensus algorithm that fits perfectly for every business requirement.

Alsunaidi and Alhaidari [27] also performed a survey on Blockchain Technology and typical consensus algorithms, while identifying their features, performance and security. They also gave an analysis of the main aspects influencing these algorithms.

3. CONSENSUS ALGORITHMS

In this section, we discuss 17 consensus algorithms. The reason behind selecting these particular 17 algorithms comprises of a number of factors.

- We wished to research and review consensus algorithms from all periods. Therefore, we start with Proof of Work, the first consensus algorithm developed, and then analyze algorithms that build on the Proof of Work model. As time passed, Proof of Stake took center stage; therefore, Proof of Stake and its major editions have been taken up.
- We discuss some consensus algorithms that are based on entirely different concepts. For instance, we decided to review algorithms such as Proof of Burn and Proof of History, based on different underlying principles altogether, and Byzantine Fault tolerant algorithms as well. This is done to incorporate all underlying principles of consensus algorithms in the literature review to conduct an adequate analysis.
- Another criterion which has been taken into consideration while selecting these algorithms is to take those algorithms which are a part of the most widely used or are being used in upcoming blockchain networks and cryptocurrencies.

The selected 17 algorithms thus provide the right blend of the above factors. For the purpose of a systematic study, we have organized the algorithms on the basis of their underlying principle, namely – Proof of Work Based, Proof of Stake Based, Byzantine Fault Tolerance Based, Quorum Based, Randomization Based, and Authentication Based.

3.1 Proof of Work Based

3.1.1 Proof of Work (PoW)

The Proof of Work [2] consensus algorithm is perhaps the first instance of the concept of a consensus algorithm being introduced to the world of blockchain. In essence, it is the simplest consensus algorithm and is based on the principle of a “hard” computing problem. Basically, a problem that is difficult to solve, but easy to verify once a solution is provided, is the basis of the Proof of Work algorithm.

For the consensus algorithm to work, a mathematical problem relating to the cryptocurrency hash of the block which is to be added next to the blockchain. The only problem in finding a solution efficiently is that the hash function used to protect the block is cryptographically secure; therefore, only a brute force method can be used to solve the mathematical problem. Once a certain node or leader claims to have solved the problem, it is verified by all other nodes present in the blockchain. When they come to a consensus that the solution should be accepted, the block is said to be “mined” and the leader is rewarded with a certain amount of cryptocurrency. Bitcoin [2], the first cryptocurrency introduced, uses Hashcash [3], a Proof of Work based consensus algorithm.

Merits:

- Properties of having hard problems (easy to verify but hard to conceive solution).

Demerits:

- Extremely high amount of energy and computing power.

3.1.2 Prime Number Proof of Work (PNPoW)

An alternate PoW system aimed to provide a more efficient means of arriving at consensus, the prime number Proof of Work [4] algorithm uses the mathematical properties of prime numbers, beginning with Sophie Germain primes [4] (where two numbers are chosen such that p and 2p+1 are primes). If we extend this nearly doubled prime concept into a chain, it forms a Cunningham chain [4], where the prime number distribution forms a similar but rarer pattern than what is seen in traditional prime numbers. Figuring out these patterns in prime chains is the proof of work that this algorithm is based on.

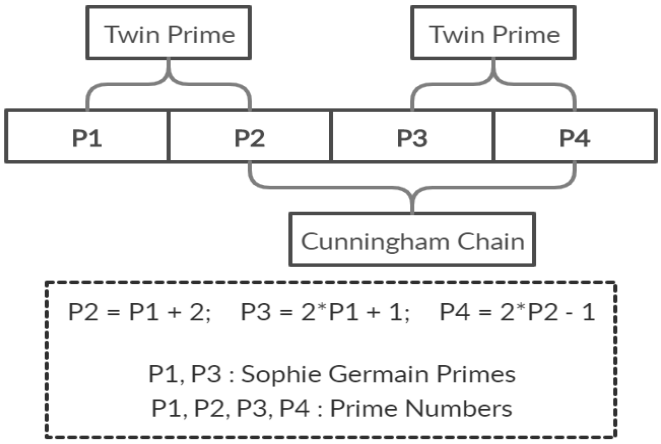

Figure 2 Sophie Germain Primes and Cunningham Chain

The tests for finding prime chains are the classic Fermat prime test with base 2, coupled with the Euler-Lagrange-Lifchitz test. Verifying pseudoprimality is enough since pseudoprimes of base 2 are much rarer than primes.

Merits:

- Non-reusability of proof of work, a much needed security measure.

Demerits:

- Tough to achieve difficulty adjustability, thus less control over mining.

3.1.3 Cuckoo Cycle Proof of Work (CCPoW)

Another variation of the Proof of Work concept, the Cuckoo Cycle PoW [5], aims to make better mining possible on commodity hardware and make it cost-effective. It works on the principle of a Cuckoo hashtable, which is made of two same sized tables, each with their own hashing function, mapping a key to a table location; therefore, each key has two possible table locations (one in each). When a new key is inserted, if the location is already occupied, it displaces the key already present in that location, which moves to the alternate location. If that is also occupied, then it displaces yet another key, forming a cycle until a vacant location is found or the maximum iterations are reached. The algorithm for detecting cycles and inserting edges into this Cuckoo graph is what forms the basis of the Proof of Work.

Solving the problem for proof of work consists of finding a Cuckoo cycle of length L. This proof can be recovered and verified by storing cycle edges in a set and enumerating nonces [5] to see which of these generate the edges of the cycle. When the verification yields a cycle of different lengths, the proof is ignored, and the graph is kept acyclic by ignoring the edge.

Merits:

- Memory boundness
- Energy efficiency

Demerits:

- Slowdown due to memory hardness

3.1.4 Proof of Burn (PoB)

Proof of Burn, invented by Iain Stewart, aims to provide a solution for the drawbacks of proof of work algorithm. PoB protocol employs the idea of destroying or burning the coins, which diminishes the need for high energy consumption resources while mining. Thereby, it reduces the PoW's dependency on powerful computational hardware. The coins are burned by sending them to a public, irretrievable and verifiable address. The coins that are sent, then become unspendable. On burning the coins, instead of waiting for months, the node immediately acquires the right to compete for the creation of new blocks. The more coins are burned by the node, the more are its chances to create the next block and get rewards. But, there is no guarantee provided by this algorithm that after burning a certain amount of coins, the node will be given an opportunity to mine. So, the node may lose a considerable amount of money before getting its reward. Also, if the number of miners increases, the odds of getting rewarded reduce.

Burning coins has a benefit though: since the number of coins on the blockchain decreases, the value of a coin increases gradually. In order to prevent the early users from having a significant advantage or from manipulating the system for their own benefit, the mining power acquired from burning the coins gradually decays over time with every block that is mined. So, to maintain the mining power, the node must keep burning coins regularly. This ensures that only the loyal nodes are rewarded.

Slimcoin [6] makes use of Proof of Burn protocol along with Proof of Stake and Proof of Work to improve security.

Merits:

- The miners have the incentive to be committed in the long term.
- Less centralized mining as the value of the burnt coins decreases over time.

Demerits:

- Prone to 51% attack [1].

3.2 Proof of Stake Based

3.2.1 Proof of Stake (PoS)

Proof of Stake [7] consensus algorithm was introduced as a viable alternative to the traditional Proof of Work consensus algorithm. The concept of this algorithm states that a person can mine more cryptocurrency based on the coins that he/she already owns. Therefore, the more coins owned, the more mining power an entity possesses. For instance, a miner who owns 4% of the cryptocurrency generated, can theoretically mine 4% of the existing blocks at the given time, and no more.

The first recorded cryptocurrency to use the Proof of Stake model is Peercoin [7]. Thin, Dong, Bai, and Jin Song Dong have discussed the security aspects of a PoS based blockchain in their research publication [8].

Merits:

- Solves the exorbitant energy consumption problem of Proof of Work based algorithms.

Demerits:

- Security issues due to forking both ends of blockchain by malicious actors, and possible mining in secret.

3.2.2 Leased Proof of Stake (LPoS)

Leased Proof of Stake is an enhanced version of Proof of Stake, yet, it is not as common [9]. It is similar to PoS, which works in such a way that the node that is chosen to mine the next block is chosen on the basis of the stake it holds in the network. This makes the nodes with low stake having very less probability of mining the next block. So, the whole network is essentially maintained by a limited number of participants, which not only creates a sense of centralization in the network but also has an effect of network security. LPoS can be seen as an improvement over the classic Proof of Stake in the sense that it solves its centralization problem and makes the network more secure by involving more participants. It does that by enabling nodes having low stake to participate in the process of choosing the new block by adding the option to lease their stake to other staking nodes. In this way, nodes with low balances can lease their funds to nodes having higher balances for a specific period of time.

The leased funds increase the overall balance of the staking node and thereby, increasing its chances of getting chosen to mine the next block. The mining reward is shared proportionally among the chosen node owner as well as all the leasers.

Waves [10] uses Leased Proof of Stake approach to achieve consensus.

Merits:

- Open participation.
- More decentralization of blockchain, as node operators may use the leased stake of other nodes to generate new blocks.

Demerits:

- Nodes may lease their stake to one particular node having the highest probability of being chosen.

3.2.3 Delegated Proof of Stake (DPoS)

To work on the security disadvantages of the Proof of Stake [11] algorithm, a delegated version was suggested, which seeks to reach consensus more efficiently. The major principle in DPoS is that of voting. All nodes on the network vote to select witnesses or users that they trust to validate transactions, and for a group of delegates, which are trusted parties responsible for maintaining the characteristics and security of the network. The witnesses that collect the most votes are now authorized to validate transactions that appear on the network, and delegates oversee the operation.

Yang, Zhou, Wu and Long [12] have proposed a modified DPoS algorithm with a downgrading algorithm in an attempt to achieve a higher degree of efficiency and decentralization.

Merits:

- Speed, due to reduced interference of nodes in transaction validation.

Demerits:

- Reduced security due to fewer number of validators in network approving transactions.

3.2.4 Proof of Importance (PoI)

Proof of Importance is a variant of PoS where an account's balance, as well as how much the account transacts with other accounts and who it transacts with, is considered to determine the node that will mine the next block. Each account in the network has an "importance" score which is calculated on the basis of how much the account uses the network. So those accounts which use the network actively will lead to the increase in their importance score. Essentially, accounts need to gain the trust of other accounts. Accounts having higher importance scores have more chances of being rewarded.

Proof of Importance is the blockchain consensus algorithm used by NEM [14].

Merits:

- Similar opportunities for mining to everyone, rewarding the legitimate miners.

Demerits:

- Vulnerable to Sybil attacks [13], where malicious or faulty entities try to gain control of the system.
- Loop attack vulnerability [14], where an entity controlling multiple accounts may improve their importance score by sending currency around through them in a loop.

3.3 Byzantine Fault Tolerance Based

3.3.1 Practical Byzantine Fault Tolerance (PBFT)

Byzantine Fault Tolerance is a characteristic of decentralized networks like a blockchain network that allows it to remain consistent and reach consensus even in the presence of faulty nodes, i.e., non-responding nodes or nodes which respond with incorrect information.

PBFT algorithm is designed to work in an asynchronous environment. It requires that the number of faulty nodes is not greater than or equal to one-third of the total number of nodes, i.e., $|R| = 3f+1$ where R is the total number of nodes or replicas, and f denotes the maximum number of nodes that can be fault. In this algorithm, there is one primary node and the other nodes are referred to as backup nodes. The client sends the request to the primary, which then multicasts it to the backup nodes. The nodes process the request and return the result to the client. Upon receiving f+1 same responses, the client is sure that the result obtained is correct [15].

Hyperledger Fabric [16] is an example of a private permissioned blockchain network which uses PBFT.

Merits:

- Effective functioning in asynchronous environment.

- Achieves consensus without complex mathematical problems, and no multiple transaction confirmations, therefore energy efficient.

Demerits:

- Works effectively only when the number of nodes is small. This is because as the network scales, the communication overhead increases and performance drops significantly.
- Vulnerable to Sybil attacks, wherein a large number of nodes in the network are controlled by a single party which could potentially compromise the security.

3.3.2 Delegated Byzantine Fault Tolerance (DBFT)

The DBFT algorithm is an adaptation of the Byzantine Fault Tolerance and is designed to handle the issue of scalability, which PBFT cannot handle. The working of this algorithm is similar to the governing system of a nation. There are ordinary nodes (citizens), bookkeepers (elected delegates) and a speaker, all of which help the network (country) to remain in consensus.

The ordinary nodes vote for bookkeepers, and it is these bookkeeper nodes that take part in the consensus. Hence as compared to PBFT algorithm, the number of nodes participating in the consensus is reduced. In each round, one of the bookkeeper nodes is selected as the speaker and gets to decide the next block in the chain. It creates and transmits a proposal block to all the bookkeepers. Each of them examines the block and its transactions. If the block is validated by more than two-thirds of bookkeeper nodes, it is added to the blockchain [17]. The role of the speaker shifts to another bookkeeper node in the next round.

NEO [18] blockchain uses this algorithm.

Merits:

- Quick generation and addition of new blocks.
- Energy efficient.

Demerits:

- A potential security issue is the lack of anonymity as the delegates need to operate under their real identities for them to be elected.

3.4 Quorum Based

3.4.1 Proof of Approval (PoApr)

Proof of Approval protocol, like Proof of Work protocol, employs consensus in a permissionless setting where any node may join or leave the network as and when they deem fit. But this protocol, unlike Proof of Work protocol and other protocols, does not require the use of physical resources [19]. Alternatively, it uses network randomness and requires the stakeholders to give explicit approvals in order to reach consensus. In this protocol, blocks are published by the network periodically at a predefined interval. This predefined interval is called a slot, which may create not more than one block. A large interval is called an epoch when it contains a predefined number of slots. Any node having a minimum stake in the network is allowed to compete with other nodes to create blocks and get rewarded. Since, for every slot, there are many creators competing for block creation, it results in high liveness of the network.

For a block to be valid, it has to be approved by the stakeholders who hold a quorum of stake. The creators of the block put forth their block, for the receiving nodes, on the network. The nodes can then validate and approve the blocks. Approval is given in the form of an explicit message to the creator of the block. In practice, though, the approvals are most likely to be given by the bigger stakeholders as their scores have more influence. When the number of approvals surpass the minimum requirement, the block creators broadcast the list of approvals on the network. All of these approvals are then placed inside the next block that is created. The blockchain essentially builds on the block having the most approvals. If there are any communication difficulties in the network, the blocks will be created as normal as long as a quorum can be reached. If a quorum cannot be reached, then the blocks will not be created until the communication improves [19].

Merits:

- No consumption of physical resources, hence near instant finality.

Demerits:

- Communication difficulties in network may lead to issues in reaching quorum of stakeholders.

3.4.2 Proof of Reputation (PoRep)

Proof of Reputation, as the name suggests, is a consensus algorithm based on the reputation or trustworthiness of the participants to ensure the validity of the blockchain network. Hence an important consideration is that reputation be significant enough so that if any participant defaults or acts maliciously, they are subjected to serious consequences.

The algorithm is an extension of the Proof of Authority in which the validators are chosen on the basis of the reputation. Once the validator nodes are finalized, the algorithm proceeds as a standard PoAuth algorithm [17] in which the block is proposed by a randomly selected mining leader and is accepted if it is signed by a majority of the validators.

GoChain [21] is an example of a blockchain network which uses this consensus algorithm.

Merits:

- Cost efficient and fast in nature as no hash power is consumed while competing for the block as in PoW [20].

Demerits:

- Use limited to private permissioned blockchains.

3.4.3 Proof of History (PoH)

Owing to the distributed nature of blockchain technology, an algorithm is required to maintain a consistent order of sequence of transactions. While traditional centralized networks use timestamps, a blockchain network has no single central clock. This is the problem that PoH addresses by providing chronological ordering, whether a transaction occurred before or after an event.

In this algorithm, there is a leader node and others are verifier nodes. The leader is responsible for ensuring the global ordering of transactions in the system. The leader sequences the transaction and

executes them on the current state and also produces a signature of the final state. These are then broadcasted to the verifier nodes, which then run the transaction on their current states and also publish the signature of output. If the output matches that of the leader, it serves as a vote in consensus [22].

Solana [22] is one of the main blockchain networks which uses this algorithm. It uses a Verifiable Delay Function (VDF) which is a hashing function with collision resistance property [17] and involves a series of computations on a single core, which can help ascertain the duration and passage of time in between events [22].

Merits:

- Independent from the reliance on the local clocks of each node.

Demerits:

- The security and credibility of the leader node is a potential vulnerability in this algorithm.

3.4.4 Predictive Proof of Metrics (PPoM)

Predictive Proof of Metrics consensus algorithm employs a prediction based approach to achieve consensus. It encourages performance, CoS (Cost of Service), and QoS (Quality of Service) [23]. Clients, Miners and Providers are the main entities of a PPoM blockchain network, where each entity has a reputation value, between 0 and 1, stored in the blockchain. All the miners have a copy of the blockchain and are connected to each other. They transmit requests for services from clients to a set of relevant providers that are registered with them.

The provider does not provide the services to the client directly to prevent attacks and instead determines if it can fulfil the request. It can do so either by a static method, where a service can be provided with guarantee, or a predictive method, where no such guarantee can be given. It then broadcasts the transaction consisting of the values required by the service to its associated miners. Each miner prepares a block from the offers it receives from various providers and broadcasts it to other miners. The block with the highest reputation, calculated by summing the individual reputations of all the providers in the block, wins. After the block is written, the client needs to confirm and transfer payment to the provider. The provider can then perform the requested service.

Merits:

- It uses metrics that are time series based and, therefore, distinctively identifiable, and de-duplicatable [23].
- DAG based structures can be used in place of list based blockchain.

Demerits:

- Suboptimal selection of blocks having the same highest reputation value.

3.5 Randomization Based

3.5.1 Proof of Elapsed Time (PoET)

This algorithm is based on the concept of a fair lottery wherein each node is equally likely to be a winner irrespective of their computing power, resources, etc. It requires a Trusted Execution Environment (TEE)

such as Intel's Software Guard Extensions (SGX). Each one of the validating blocks requests the TEE for a random waiting time and goes to sleep for that specific duration. The first node to wake up, i.e., the node with the shortest waiting time, gets the right to commit the block to the network [24].

Hyperledger Sawtooth [25] is an example of a blockchain network that uses this algorithm.

Merits:

- It is a fair algorithm in terms of deciding who gets to commit the new block.
- Leader selection is not resource-intensive.

Demerits:

- Reliance on a third party and specific hardware makes its use limited.

3.5.2 Proof of Luck (PoL)

Proof of Luck [26] consensus algorithm aims to cut down the use of high computational requirements of PoW and achieve increased transaction throughput. PoL utilizes Trusted Execution Environments (TEEs), such as the Intel SGX. This requirement of specialized hardware often turns out to be a disadvantage. This protocol makes use of the random number generator, provided by a TEE platform, to assign a random number ranged in the interval [0, 1) to each block that is created by the miner. Higher numbers are treated as luckier and lower as unluckier. The miners add the created block to their chain and broadcast it to the network. Miners would prefer their block to be added to the chain having the highest luck score. This luck score is calculated by adding the luck values of each block present in the chain.

To enhance network communication, the broadcast is delayed by a specific amount of time based on the luck values, with short delay times linked to higher luck values. If during the delay, a block is received, the miner moves to this chain if it has higher luck and broadcasts this chain to the peers.

Alsunaidi and Alhaidari [27] have discussed in their paper about this algorithm's main features, performance, scope and possible drawbacks. Luckychain [28] is a prototype blockchain based on Proof of Luck using Intel SGX capabilities of modern CPUs.

Merits:

- Protects the network from double spending attacks as the attacker needs to be very lucky in order to be successful in carrying out such an attack.

Demerits:

- Susceptibility to partition attacks, as two partitioned groups may confirm two different chains having each partition's largest sum [29].
- Requires specific hardware.

3.6 Authentication Based

3.6.1 Proof of Authentication (PoAh)

Proof of Authentication, as the name suggests, is a consensus algorithm that incorporates authentication process during the validation procedure of a block. Its main objective is to make the blockchain lightweight so that it becomes applicable to a network of resource constrained nodes [30].

In this algorithm, the nodes in the network are divided into two types: normal and trusted nodes. The trusted nodes are responsible for block validation and have a certain minimum trust value. With each successful authentication of a block, the trust value of the node is incremented by 1 while false authentication decreases the trust value by 1. Normal nodes also gain 0.5 trust value by identifying the authenticated block. This ensures that trusted nodes which fall below the minimum trust value threshold become normal nodes and vice-versa [30].

Initially, the network nodes generate transactions and combine them to form blocks. This algorithm follows the ElGamal method for encryption and decryption [30]. The block is signed by the private key PrK of the node while the public key PuK is made available to everyone. The block is then broadcasted. A trusted node, on receiving the block, verifies it using the public key of source and then performs secondary evaluation by checking the MAC address.

Algorithm 1: Procedure of the Proposed PoAh.

Inputs : All nodes in the network follow $SHA-256$
Hash Individual node has Private (PrK)
and Public key (PuK)

Nodes combine transactions to form blocks
$(Trx^{+}) \rightarrow$ blocks
Blocks sign with own private key
(S_{PrK})(block) $\rightarrow$ broadcast
Trusted node verifies signature with source public key
(V_{PuK})(block) $\rightarrow$ MAC Checking
if *Authenticated* **then**
    $block||PoAH(D) \rightarrow$ broadcast
    $H(block) \rightarrow$ Add blocks into chain
else
    DROP the block
GOTO $(Step-1)$ for next block

Figure 5 Proof Of Authentication Algorithm [31]

After successful authentication, the trusted node broadcasts the validated block to other nodes along with PoAh information. A node in the network, on receiving a validated block from a trusted node, simply checks its PoAh information to verify it and add the block in the chain. This is done by computing the hash value of the block and storing the hash value of the previous block in it for maintaining the links in the chain [30].

The PoAh algorithm was developed as an improvement to the PoW algorithm for the application of blockchain in the resource constrained IoT and edge computing systems [30].

Merits:

- Energy Efficient.
- Applicable in a resource constrained environment.
- Resistant to 51% attack [30].
- Low latency [32].

4. OBSERVATIONS AND ANALYSIS

As part of the research and review process, we made the following astute observations and inferences as far as consensus algorithms are concerned.

4.1 Energy Requirements

Most of the early work in consensus algorithms suffers from a specific downfall- energy requirements and inefficiency. Taking the case of Bitcoin [2], when the price of BTC touched the sky, enthusiastic miners set up mining rigs which used electricity equivalent to a country like Austria. All of this electricity was being used just to solve the cryptographic hash problem to mine more blocks, and nothing else. As progressions were made, the first order of the day was to reduce this energy consumption, which gave way to a different class of consensus algorithms. Another alternative was to direct this energy towards something useful, as seen in Coin.AI [33]. Regardless, it is safe to say now that energy requirements are becoming less and less of a problem as we proceed.

4.2 Security

Ensuring the security of consensus algorithms often revolves around placing a certain degree of trust in the participant nodes. In algorithms such as Delegated Proof of Stake, misbehaviour or malicious intent of validators and other entities can lead to a compromise of the blockchain, thereby violating the very essence of the blockchain (being a decentralized and trustless entity).

4.3 No Universal Best Algorithm

Even though many consensus algorithms tout themselves as the best in the business, applying a certain algorithm to a project simply depends on the level of energy, functionality, and security that the algorithm needs. There is no one-size-fits-all model, hence why most blockchain or cryptocurrency projects look to develop their own version of a consensus algorithm, which suits their application like a glove.

4.4 Cryptocurrency and Pricing Impact

As is the case with any currency, the price of cryptocurrencies also depends on supply, demand, and the cost of production. We observe that the third factor here, which is the cost of production, is the most relevant to our research since consensus algorithms are directly involved in the generation of cryptocurrencies. We discuss the following two main factors that influence the cost of production of cryptocurrencies, i.e., characteristics of consensus algorithms:

a) Energy requirements*:* High energy requirements for a consensus algorithm directly increase the cost of a single unit of cryptocurrency. Taking the example of Bitcoin, using the Hashcash Proof of Work algorithm [3], we can see that mining requires a great deal of energy, which is justified by the cost of BTC going upwards of USD 9000 at the time of writing. More energy expended in generating BTC will mean the final product costs more to obtain for a third party.

b) Hardware requirements: Even if consensus algorithms do not require high energy to function, the cost of hardware for mining can be a significant factor in cryptocurrency cost. Algorithms that use specialized

hardware, such as Proof of Elapsed Time (used in Hyperledger Sawtooth [25]), are bound to have a direct impact on the cost of production.

In conclusion, it can be stated that consensus algorithms (due to them being a means of production of cryptocurrency) are a direct indicator of the price per unit of cryptocurrency. The factors that pricing depends on are energy requirements, hardware requirements, and general characteristics of the consensus algorithm.

4.5 Throughput

The throughput, or number of items passing through an algorithm or system per second, is directly linked to the speed of transaction processing in consensus algorithms.

If we consider proof of work or proof of stake based systems, throughput lags behind other algorithms due to the purpose of reducing the amount of wait time for enough blocks to confirm a transaction. Increased throughput may put undue pressure on the system, leading to an overflow in wait times. In improvements of these algorithms, such as the delegated proof of stake, throughput is accelerated [34] due to a selected number of nodes being responsible for transaction verification. Similar case applies to quorum based algorithms as well.

When it comes to Byzantine fault tolerant algorithms, throughput is typically higher [34] than that in proof of work or proof of stake based algorithms, due to the fact that these algorithms function well in permissioned environments where a smaller number of nodes are present. The throughput limit is set by the maximum performance of a node. [34]

For algorithms based on other principles, throughput and transaction speed depend on intrinsic conditions of the algorithm itself, making it difficult to analyse them categorically.

5. COMPARISON

Table I provides a comparison between various consensus algorithms reviewed in this paper. This comparison covers numerous aspects, such as energy requirement, scalability, and network permission. These parameters have been selected keeping in mind the common considerations which arise when choosing a consensus algorithm for a new blockchain project.

CONSENSUS ALGORITHMS	Energy Requirement	Network Permission	Tolerated Power of Adversary	Scalability	Specialized Hardware Requirement	Example
Proof of Work	High	Permissionless	< 25% computing power	Strong	No	Bitcoin [2]
Prime Number Proof of Work	High	Permissioned	< 50% network mining power	Strong	No	Primecoin [4]

Cuckoo Cycle Proof of Work	Low	Permissioned	Cycle length between 20 and 64	Strong	No	Cortex [35]
Proof of Stake	Low	Permissionless	< 51% stake	Strong	No	Peercoin [7]
Delegated Proof of Stake	Low	Permissionless	< 51% validators	Strong	No	BitShares [11]
Leased Proof of Stake	Low	Permissionless	< 51% stake	Strong	No	Waves [10]
Proof of Importance	Low	Permissionless	< 50% importance	Strong	No	NEM [14]
Proof of Approval	Low	Permissionless	~ 50% stake	Weak	No	-
Proof of Burn	Low	Permissionless	< 51% hash power	Strong	No	Slimcoin [6]
Proof of Luck	Low	Both	< 50% processing power	Strong	Yes	Luckychain [27]
Proof of Reputation	Low	Permissioned	-	Strong	No	GoChain [21]
Proof of History	High	Both	-	Strong	No	Solana [22]
Proof of Elapsed Time	Low	Both	-	Strong	Yes	Hyperledger Sawtooth [24]
Practical BFT	Low	Permissioned	< 33.3% replicas	Weak	No	Hyperledger Fabric [16]
Delegated BFT	Low	Permissioned	< 33.3% replicas	Weak	No	NEO [18]
Predictive Proof of Metrics	Low	Both	-	Weak	No	-
Proof of Authentication	Low	Permissioned	-	Strong	No	IoT Applications

Table 1 Comparison of Various Consensus Algorithms

5.1 Energy Requirement

The consensus algorithms have been given a high and low rating based on their energy requirements. PoW has the highest energy requirements as the miners need to solve the mathematical problem using only the brute force method, which consumes a lot of electrical energy. Other algorithms such as PoB, PoL, PoRep, PBFT, and DBFT require reasonably low energy.

5.2 Network permission

Anyone is free to join a permissionless blockchain network, while only approved users can access permissioned network.

For permissionless blockchains and implementations of cryptocurrency with a global target audience, where speed is of the essence, Proof of Stake based algorithms are useful, including Leased Proof of Stake, Proof of Importance, and Delegated Proof of Stake. This is due to the nature of Proof of Stake algorithms relying on validators or witnesses to control transactions and achieve consensus.

For permissioned blockchains, where speed can be compromised for the sake of security, and efficient mining hardware is available, algorithms such as cuckoo cycle proof of work, prime number proof of work and proof of luck are adequate. Although these algorithms might provide a computing slowdown, it can be dealt with, due to a smaller number of participants and the private nature of the network.

5.3 Tolerated Power of Adversary

A suitable consensus algorithm must be able to thwart attacks from adversaries to some extent. This value denotes a certain percentage of control over the network that the adversary needs in order to attack the network security successfully. This value ranges from as low as 25% in the case of PoW to 51% in the case of PoS. However, these values may change in the future as more potential attacks are discovered against these algorithms. Some of these values are provided by [36].

5.4 Scalability

The algorithms need to be able to reach consensus in the face of ever-growing nodes in the network. As the nodes in the network increases, the transactions occurring between them increases. Scalability determines the number of transactions that can be handled at the same time. PoApr, PBFT, and DBFT are the only algorithms from those studied in this paper, which are not scalable as their communication overhead increases substantially.

5.5 Specialized Hardware Requirement

All the consensus algorithms reviewed in this paper do not require any specialized hardware except PoL and PoET, which require Trusted Execution Environments (TEEs), such as the Intel SGX, for their implementation.

6. CONCLUSION

Consensus algorithms are integral to the structure and function of any blockchain network. When looking at a consensus algorithm to be used by an organization for its applications, it is better for the organization to devise its own algorithm according to its needs. Organizations and firms looking to use this document as a reference for their blockchain applications must realize that only after analysing their requirements and demands, can they decide on what consensus algorithm to employ. We hope that with this review paper, our ideas behind the suitability and comparison can be extended to include other consensus algorithms in the future. This would act as a cornerstone for further research and development of the blockchain technology in the field of cryptocurrencies and new areas like education, banking, healthcare, public services, etc.

REFERENCES

[1] Q. He, N. Guan, M. Lv, and W. Yi, "On the consensus algorithms of blockchain/dlt for internet of things," in IEEE 13th International Symposium on Industrial Embedded Systems (SIES), 2018, pp. 1–10.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf [Accessed: 12-Jan-2020].

[3] A. Back, "Hashcash - A Denial of Service Counter-Measure," 2002. [Online]. Available: http://www.hashcash.org/hashcash.pdf [Accessed: 12-Jan-2020].

[4] S. King, "Primecoin: Cryptocurrency with Prime Number Proof-of-Work," 2013. [Online]. Available: https://primecoin.io/bin/primecoin-paper.pdf [Accessed: 15-Jan-2020].

[5] J. Trump, "Cuckoo cycle: a memory bound graph-theoretic proof-of-work." in International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2015, pp. 49-62.

[6] P4Titan, "Slimcoin: A Peer-to-Peer Crypto-Currency With Proof-of-Burn," 2014. [Online]. Available: https://github.com/slimcoin-project/slimcoin-project.github.io/raw/master/whitepaperSLM.pdf [Accessed: 17-Jan-2020].

[7] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," 2012. [Online]. Available: https://decred.org/research/king2012.pdf [Accessed: 19-Jan-2020].

[8] W. Y. M. M. Thin, N. Dong, G. Bai and J. S. Dong, "Formal analysis of a proof-of-stake blockchain," in 23rd International Conference on Engineering of Complex Computer Systems (ICECCS), IEEE, 2018, pp. 197-200.

[9] "Leasing Proof of Stake" [Online]. Available: https://docs.wavesplatform.com/en/blockchain/leasing [Accessed: 12-Feb-2020].

[10] "Waves Blockchain. Blockchain technology from Waves" [Online]. Available: https://wavesplatform.com/technology [Accessed: 12-Feb-2020].

[11] "The Bitshares Blockchain" [Online]. Available: https://www.bitshares.foundation/papers/BitSharesBlockchain.pdf [Accessed: 15-Jan-2020].

[12] F. Yang, W. Zhou, Q. Wu, R. Long, N. N. Xiong and M. Zhou, "Delegated Proof of Stake With Downgrade: A Secure and Efficient Blockchain Consensus Algorithm With Downgrade Algorithm," IEEE Access 7, 2019, pp. 118541-118555.

[13] J. R. Douceur, "The sybil attack," in International workshop on peer-to peer systems. Springer, Berlin, Heidelberg, 2002, pp. 251-260.

[14] NEM Foundation, "NEM: Technical Reference," 2018. [Online]. Available: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf [Accessed: 11-Feb-2020].

[15] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," In OSDI, vol. 99, no. 1999, 1999, pp. 173-186.

[16] "Hyperledger Fabric" [Online]. Available: https://www.hyperledger.org/projects/fabric [Accessed: 14-Feb-2020].

[17] A. Shahaab, B. Lidgey, C. Hewage and I. Khan, "Applicability and Appropriateness of Distributed Ledgers Consensus Protocols in Public and Private Sectors: A Systematic Review," in IEEE Access, vol. 7, 2019, pp. 43622-43636.

[18]"Neo White Paper" [Online]. Available: https://docs.neo.org/docs/en-us/basic/whitepaper.html [Accessed: 13-Feb-2020].

[19]S. Takahashi, "Proof-of-approval: A distributed consensus protocol for blockchains," Tech. Rep., 2018, pp. 1–21.

[20]F. Gai, B. Wang, W. Deng and W. Peng, "Proof of reputation: A reputation-based consensus protocol for peer-to-peer network," in International Conference on Database Systems for Advanced Applications. Springer, Cham, 2018, pp. 666-681.

[21]"GoChain" [Online]. Available: https://gochain.io/ [Accessed: 15-Feb-2020].

[22]"Introduction – Solana Architecture" [Online]. Available: https://docs.solana.com/book/ [Accessed: 15-Feb-2020].

[23]V. S. V. Bhamidipati, M. Chan, A. Jain, A. S. Murthy, D. Chamorro and A. K. Muralidhar, "Predictive Proof of Metrics – a New Blockchain Consensus Protocol," In 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), IEEE, 2019, pp. 498-505.

[24]L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu and W. Shi, "On security analysis of proof-of-elapsed-time (poet)," in International Symposium on Stabilization, Safety, and Security of Distributed Systems. Springer, Cham, 2017, pp. 282-297.

[25]"Hyperledger Sawtooth" [Online]. Available: https://www.hyperledger.org/projects/sawtooth [Accessed: 14-Feb-2020].

[26]M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in Proc. SysTEX, 2017, pp. 2–7.

[27]Alsunaidi, Shikah J., and Fahd A. Alhaidari, "A Survey of Consensus Algorithms for Blockchain Technology," in IEEE International Conference on Computer and Information Sciences (ICCIS), 2019, pp. 1-6.

[28]"Luckychain" [Online]. Available: https://github.com/luckychain/lucky [Accessed: 18-Feb-2020].

[29]X. Chen and S. Zhao, "Scalable, Efficient, and Consistent Consensus for Blockchains," arXiv preprint arXiv:1808.02252, 2018.

[30]D. Puthal, S. P. Mohanty, V. P. Yanambaka and E. Kougianos, "Poah: A novel consensus algorithm for fast scalable private blockchain for large-scale iot frameworks," arXiv preprint arXiv:2001.07297, 2020.

[31]D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos and G. Das, "Proof-of-authentication for scalable blockchain in resource-constrained distributed systems," in 2019 IEEE International Conference on Consumer Electronics (ICCE), IEEE, 2019, pp. 1-5.

[32]D. Puthal and S. P. Mohanty, "Proof of authentication: IoT-friendly blockchains," IEEE Potentials 38, no. 1, 2018, pp. 26-29.

[33]A. Baldominos and Y. Saez, "Coin.AI: A proof-of-useful-work scheme for blockchain-based distributed deep learning," Entropy, vol. 21, no. 8, 2019, p. 723.

[34]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), IEEE, 2017, pp. 2567-2572.

[35]Z. Chen, W. Wang, X. Yan and J. Tian, "Cortex-AI on Blockchain," [Online]. Available: https://www.cortexlabs.ai/Cortex_AI_on_Blockchain_EN.pdf [Accessed: 22-Jan-2020].

[36]L. M. Bach, B. Mihaljevic and M. Zagar, “Comparative analysis of blockchain consensus algorithms,” in 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2018, pp. 1545-1550.

[37]V. Sharma and N. Lal, "A Detail Dominant Approach for IoT and Blockchain with their Research Challenges," 2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3), Lakshmangarh, Sikar, India, 2020, pp. 1-6, doi: 10.1109/ICONC345789.2020.9117533.

[38]Vishal Sharma and Niranjan Lal “A novel comparison of consensus algorithms in blockchain” Advances and Applications in Mathematical Sciences (ISSN 0974-6803), VOLUME 20, ISSUE 1, NOVEMBER 2020, PAGES 1-13.