Survey of the Different Consensus Mechanisms in a Blockchain and the Difference Between Proof Based and Vote Based Algorithms and a Projection of the Future of Consensus Algorithms in Blockchain

Vaibhav Chopra
19BCE0932
mailvaibhavchopra@gmail.com

Abstract

Since the inception of Bitcoin in 2009, the different blockchain-based cryptocurrencies that followed have been largely powered by Proof of Work (PoW). PoW enjoys over 85% of the total market capitalisation when it comes to digital currencies. But with time, various new mechanisms for ensuring security guarantees have been introduced which have largely gone unnoticed but might be much more superior than the classic PoW mechanism in some aspects. In the current scenario, Proof of Stake seems to be a much more viable option based on long term sustainability and eco-friendliness. The various advantages of a PoS-based blockchain are analysed and the limitations and shortcomings are also addressed. Other options aside from PoS and PoW are also touched upon. Security and performance implications of different consensus schemes are explored and adequate adversarial strategies are discussed wherever needed. Figuring out the right consensus mechanism for each use case can improve efficiency and make blockchains much more feasible for newer and unprecedented applications. How different altcoins like Ethereum and Chia are pursuing non-PoW mechanisms for better performance will also be highlighted. The main aim of this paper is to find a consensus mechanism that can provide good performance without sacrificing security.

Introduction

First introduced by Haber and Stornetta, Blockchain has recently been considered one of the most powerful technologies. He drew the only major attention after the introduction of Bitcoin by Nakamoto. This is because Bitcoin solves the problem with the traditional payment method: that of trusting a third party. Usually, when people pay, they have to trust a third party, who will verify the authenticity of their payment, before using it. Unfortunately, this middle class is skeptical, whether or not it can be exploited to cheat its users. The problem arises from formal integration, where everything depends on one organization, which makes trust insufficient. The solution to this problem is using multiple private entities to validate the transaction, which transforms the view from the medium to the power partition. Motivated by this idea, there is a ledger in Bitcoin and other recent Blockchain models that record all transactions successfully verified. For example, Alice sends Bob 5 dollars, Bob sends Carrie 10 dollars. So, depending on this ledger, you may know how much money a person has. To make the system more reliable, the couple is managed by many organizations, which can be called nots or groups. Satoshi suggested the formation of this article informing us of the so-called block area, containing successful transactions verified. The tree mentioned above is rebuilt, and contains many blocks of interstate commerce. This is the reason for the term “Blockchain”. The first block of metal is called the genesis block, which contains the first Bitcoin transaction. When any transaction is proposed, its performance is guaranteed by other locations. If applicable, which means that the sender has enough money to send (verified past transactions in old blocks), and also the shipper confirms this transaction by signing his digital signature within the transaction, will be entered in a block. From here, in order to make the transaction truly valid, the block containing this transaction must be added to the chain, which should be viewed by all other locations. The node will attempt to insert a block containing multiple transactions by spreading it across, which it proposes to add to its current network. However, if a transaction is validated in this way, it can be confusing when the rest of the community tries to broadcast their received block. To prevent this situation, an agreement should be called, called a synchronization agreement, between all nodes as to which blocks to apply, and which locations are allowed to enter their proposed blocks. So far, most of the algorithms for compatibility have been proposed.

Since its inception in 2009, the blockchain of Bitcoin has fueled new capabilities and many novel systems, such as smart contracts, have been designed to take advantage of blockchain. Bitcoin has been set up several times to optimize compliance (e.g., timeframe and hash performance), and network parameters (e.g. block size and data streaming protocol) and to increase blockchain efficiency. For example, Litecoin and Dogecoin - the most prominent Bitcoin forums - reduce the block's production time from 10 to 2.5 and 1 minute. In line with these efforts, other blockchain-based networks (such as Ethereum) are driven by the desire to increase the consistency and limits of the network and to

facilitate the deployment of applications used for blockchain over-denial. Although a number of consensus agreements have been proposed (PBFT, Proof of Stake, Proof of Elapsed Time), most of the existing blockchains have access to the most expensive Proof of Work (PoW) process - 80% of total market capitalization digital channels are available. While Bitcoin security provisions were once th

Overview of Blockchain

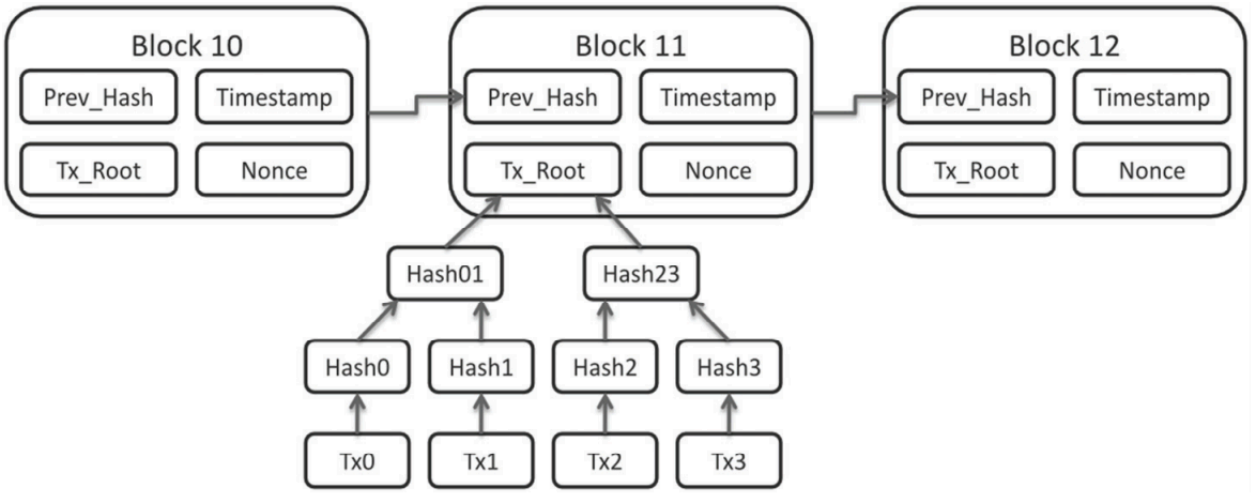
In this session we shall supply an overview of Blockchain, which includes transaction verification and block architecture inside Blockchain. These fundamentals shall provide a solid ground for the elaborate concepts and proof mechanisms that we will discuss in the subsequent sections.

Transaction Verification

The first work that every system has to do is to verify the identity of the sender, to make sure of one thing: the transaction between the sender and the receiver is requested by the sender, and not by anyone else. For example, when Bob sends Alice 10 dollars, then the request of this transaction comes to a third-party verifier; this middleman has to make sure that this message does indeed come from Bob. Fig. 1 shows an overview of this verification. In order to do this work, two definitions are employed: public and private keys, which are known as digital signatures. When any user makes a transaction, he or she has to use his or her private key to “sign” the transaction, which could be understood that the transaction and his or her private key are used as inputs to a sign function to create a signature. In Bitcoin, the algorithm used for creating a signature for each transaction is the elliptic curve digital signature algorithm. Then the transaction combines with this signature to become a request transaction. The verifier will check if this entity belongs to the correct person or not, by using the sender’s public key, which is known by everybody. Afterward, the transaction, sender’s signature, and his or her public key will be inputted together to a verify function to get the result: true or false. If the result is true, then the requester is the true sender in the transaction, and vice versa. Because the signature in each transaction contains 256 bits, if anyone wants to fake this signature to make a fraudulent transaction, he or she has to guess 2^256 cases, which is practically impossible. Besides checking the validity of the sender, the verifier also has to check the validity of the transaction as to whether the sender has enough money to send to the receiver, or not. This work could be done by looking at the ledger, which holds information about every past successful transaction.

Blockchain Architecture

As mentioned before, the key phenomenon giving rise to the name Blockchain is the



series of blocks, which connect sequentially to each other like a chain. Each block contains many transactions, which are validated, as shown in the previous section.

The above figure describes the architecture inside a block. Besides the list of transactions contained inside the block, the block contains some fields in the block header:

• Prev_Hash: this field can be known as a reference to parents, which is a link of a block to its previous one in the chain. All the information inside the previous block will be inputted to a hash function to get a value, then this value will be assigned to the field Prev_Hash in the new block. In Bitcoin, a 256-bit hash function is used to get this value.

• Timestamp: the time when the block was found.

• Tx_Root: this field, which is also known as the Merkle root, contains the hash value of all validated transactions of the block. As seen from the example in Fig. 2, all the transactions are hashed into a hash value; then they combine with each other pair-by-pair, and are inputted to another hash function. This work is repeated, until there is only a single entity, which stands for the Merkle root.

• Version: this field contains the protocol version used by the node proposing the block to the chain.

• Nonce: this field is used in PoW, which proves the efforts that a node has paid for getting the right to append his block to the chain. This field will be presented in the next section.

• Bits: this field indicates the difficulty level of the PoW, which will be introduced in the next section.

Proof Based Consensus Algorithm

This section introduces the proof-based consensus algorithm. The original work is PoW, proposed by Nakamoto. To date, many variants of proof-based consensus algorithms have been proposed, which are based on PoW, PoS, their hybrid form, and other variants that are made independently from these two major ones. The basic concept of proof-based consensus algorithm is that among many nodes joining the network, the node that performs sufficient proof will get the right to append a new block to the chain, and receive the reward.

PoW-based consensus algorithm

As mentioned, in the Blockchain network, if every node tries to broadcast their blocks containing the verified transactions, confusion could possibly arise. For example, consider a transaction that is verified by many nodes, who will then put it into their blocks, and broadcast to other nodes. If the broadcasting work is free, this transaction could be duplicated in different blocks, then the ledger is meaningless. In order to get agreement between all nodes about the newly added block, the PoW requires each node to solve a difficult puzzle with adjusted difficulty, to get the right to append a new block to the current chain. The first node who solves the puzzle will have this right. Specifically, before solving this puzzle, all the verifying nodes would have to put their verified transactions, as well as other information like Prev_Hash and Timestamp, into a block. Then they start solving this puzzle, by guessing a secret value, which is the nonce field , then put it into the block. All the information inside the block header will be combined together, and inputted to an SHA-256 hash function. If the output of this function is below a given threshold T, which is designated by the difficulty, the secret value is accepted. Otherwise, the node has to make another guess of the secret value, until he gets the answer. The difficulty of the puzzle will be adjusted after every 2016 blocks are appended, so that the average speed for adding a new block in the chain is 1 block per 10 minutes. Also, the more difficult the puzzle is, the smaller the threshold T is. Thanks to the usage of SHA-256, guessing this value is extremely difficult, which makes every node guess many times to get the answer, unless they are lucky enough. Because of the efforts paid for guessing the right value, this work is called the PoW. Also, the node joining the network using PoW can be called a miner, and the action of finding a suitable nonce is called mining. When a node finds the secret value, he broadcasts his proposed block with this value to other nodes, to notify them that the answer has been found. Right after that, all the miners receiving this message, who have still not found the secret answer for their

puzzles, will stop guessing. Instead, they check the broadcasted block for whether all the transactions are valid; if the block contains the Prev_Hash value is the hash value of the last block in their chain. If all the verifications are correct, then these nodes will append the proposed block to their current chain, and re-start guessing the secret value, by repeating the steps above again. However, there is a rare case when more than 1 miner finds the answers for the puzzle, before it being noticed that another miner has also found another suitable answer. At that time, these miners will still broadcast their block with the found nonce. Then, other miners who receive the first coming block will ignore the others coming later. This work leads to the forking problem: in the verifying network, there are different chains of blocks (they should be the same). Satoshi proposed that those miners will keep mining a new block on their forks, until one fork is made longer than the others. So at this time, all nodes have to follow this longest fork. Whenever a block is recognized in the chain by all the nodes, the miner appending this block will receive some bitcoins as a reward.

The Problem with PoW

The Proof-of-Work consensus mechanism has some issues which are as follows:

- **The 51% risk**: If a controlling entity owns 51% or more than 51% of nodes in the network, the entity can corrupt the blockchain by gaining the majority of the network.
- **Time consuming**: Miners have to check over many nonce values to find the right solution to the puzzle that must be solved to mine the block, which is a time consuming process.
- **Resource consumption**: Miners consume high amounts of computing power in order to find the solution to the hard mathematical puzzle. It leads to a waste of precious resources(money, energy, space, hardware). It is expected that the 0.3% of the world's electricity will be spent to verify transactions by the end of 2018.
- Transaction confirmation takes about 10–60 minutes. So, it is not an instantaneous transaction; because it takes some time to mine the transaction and add it to the blockchain thus committing the transaction.

Mining pools and the centralising effect of PoW

Thanks to modern hardware, the block appending speed increases day-by-day, which makes the difficulty of the puzzle harder. Therefore, in order to be the first puzzle solver,

the miners have to invest more in the hardware. This investment results in other miners, who do not have good condition, being unable to compete effectively. Due to this, lots of smaller miners often join up in pools to combine their resources in hopes of improving their chances of winning the block reward. This strategy has bore fruit in some cases but also has its own drawbacks.

Following this pooling principle opposes the initial vision for Bitcoin and other cryptocurrencies that aimed at decentralising the whole money system. These strategies also heighten the chances of a 51% attack, something which is highly unfavourable for any cryptocurrency.

PoS based consensus

The proposed PoW is supposed to be unfair: while some miners owning modern and powerful equipment could find the suitable nonce easier, others with poorer condition could find it very difficult to be the first one to find a suitable nonce. PoS is supposed to deal with this inequality. Being firstly introduced and discussed in Bitcoin forum from 2011, PoS has had some variants and research contribution to it. The basic idea of these consensus algorithms is using the stake to decide who will get the chance to mine the next block of the chain. Using stake as a proof has an advantage: anyone who owns much stake would be more trustful. He or she would not want to perform any fraudulent actions to attack the chain that contains much of his or her profits. Furthermore, using PoS would require any attackers to own at least 51% of all stakes in the network to perform a double spending attack, which is very difficult. There are currently two popular kinds of consensus using PoS: the kind using pure stake to make consensus, and the hybrid kind, which combines PoS and PoW.

Pure PoS based consensus

The pure PoS form could be seen in Nextcoin (aka. Nxtcoin, http://nxtcrypto.org/). In this platform, the more stake a miner owns, the more chance he has to mine a new block. Specifically, if there are a total b coins from all the miners, and miner M owns a coins ($a < b$), the chance for miner M getting the right to mine a new block is a/b. The lucky miner picking work is executed every 60 minutes, and this work is made randomly based on the stake of each miner, as mentioned. Once a miner gets the chance to mine a new block, he will verify the transactions, collect them into a block, then broadcast it to the other miners, and receive the rewarding fee.

PoS by Entropy Value

Kiayias et al. also employed the idea of Bentov, follow-the-Satoshi procedure to execute the PoS consensus. They claim that the leader election should be done randomly by calculating an entropy value. This calculation should be secure enough, so that it would be difficult to simulate the protocol to predict the calculation, and manipulate the leader election. Kiayias et al. considered the work of Bentov et al. for leader election as a way to calculate entropy based on the current status of the Blockchain. The first difference of Kiayias et al.'s work from Bentov et al.'s is that they snapshot the stake of each stake holder after an interval called an epoch. In each epoch, a collection of stake holders will be chosen to execute the so-called coin-flipping protocol. Based on these results, the leaders and the stake holders executing this protocol in the next epoch will be chosen, which makes sure that the entropy calculation is difficult to simulate. Also, in order to raise the equality, the leader will have to create the block only, while the work of adding transactions belongs to a group of stake holders called endorsers, who are voted like the leader. The rewards will be divided equally to those leader and endorsers.

Delegated Proof of Stake

Using the stake as a proof for voting, not for getting the chance to mine a new block, is the idea of Larimer. This kind of consensus algorithm is called delegated proof of stake. In this platform, there are many people who hold stake, and they will have to vote for a delegation, which includes some "witnesses", who are miners verifying the transactions and maintaining the chain. The more stake a person owns, the more powerful voting he has to assign the witness. After the verifying congress is made, the witnesses inside will verify the transactions, and produce blocks, including the valid ones. The list of witnesses is always shuffled. With the creating speed of 2 seconds per block, the witness in the list will have to produce blocks sequentially. If any witness fails to produce his block, he would potentially be removed from the delegation. Whenever a witness creates a block to append to the chain, he will receive a reward.

Hybrid form of PoW and PoS

Being officially introduced in a paper by King and Nadal, PPcoin could be considered as the first variant of PoS, but it still uses PoW. These authors have proposed a definition called the 'coin age' of each miner, which is calculated by his stake multiplied by the time that the miner has owned it. For example, Bob receives 10 coins from Alice and keeps them for 10 days; consequently, he has 100 coinday, and if he sends 2 coins to anyone, the coin age of these 2 coins accumulated will be destroyed. In order to get the right to append a new block to the chain, the miner creates a kind of block called a coin stake, which like before, holds many transactions, but includes a special one from that miner to

himself. The amount of money spent on this transaction will provide the miner more chance to mine a new block. Afterwards, he will have to do a puzzle, like PoW. The more money he spends on the transaction, the easier the puzzle he has to solve. If any miner solves the puzzle first, he will get 1% of the amount of coins he has spent in the transaction, but the coin age accumulated by these coins will be reset to 0. Being dissimilar to King and Nadal, Vasin do not employ coin age with their Blackcoin, because they suppose that using coin age could provide the attacker the chance to accumulate enough value to cheat the network. Worse yet, there could be some miners who hold their stake until they have a lot of coin age, while staying offline from the verifying system. Therefore, Vasin propose using the raw stake instead of coin age for providing miners the chance to mine a new block, which could encourage more nodes to be online to get the rewards. Being noticeable with the offline miner, Ren propose using an exponential decay function with the coin age: the more the miner waits for the increase of the coin age, the less speed of increase it has.

The problem of double spending attack is again shown by Duong et al. The problem with miners owning more than 51% mining power raises a high alert for the security of Blockchain. They proposed a method for mitigating the double spending attack by combining PoW and PoS. The aim of this action is to make sure that even if a miner owns more than 51% of the mining power, he still does not have much chance to make a fraudulent action. These authors propose using the PoW first for choosing a winner, who is the first one getting the answer of the puzzle. Subsequently, this winner will not only append a block called PoW block to the chain as usual, but he will also provide a basis to choose another miner who owns stake. This basis is: if the return value of a hash function, whose input parameters are the newly appended PoW block and stake owner's private key, is below a threshold, the chosen miner will have the chance to append a so-called PoS block to the chain. Duong et al. claim that each PoS block is linked to a single PoW block, and each PoW block is linked to a previous PoS block. Consequently, it is difficult to make a double spending attack. The reason is that assuming that an attacker could own more than 51% of the computing power, then he could get the right to be the miner appending the illegal PoW block to the chain. However, right after that, there is still a PoS block appended by another miner chosen by the mentioned base. This stake miner will see the longest chain at that time, and realize that the latest one is a PoS block, which he could not append his PoS block to. As a result, the attack fails. The double spending attack is supposed to happen only when the attacker owns not only more than 51% of the computing power, but also more than 50% of all the stake holders. This kind of consensus is not based on the amount of stake, but only for miners owning stake.

A More fair hybrid PoS and PoW hybrid strategy

Bentov et al. proposed a solution called Proof of Activity. This combines PoW and PoS in order to not only solve the double-spending attack, but also handle some tragedies made by PoW called the tragedies of the common. The first mentioned tragedy is that only the miners solving the puzzle get the reward, while the others who have the role of preserving the ledger, update it and validate the new block; they do not receive anything. The second one is that miners can co-operate with others to raise the transaction fee to be high to charge the users. This action could make the Blockchain useless, because nobody would want to use it. In order to deal with these tragedies, these authors proposed creating an empty block by PoW first: all the miners will try to find a nonce with a block without any transactions. After finding a suitable nonce, the miner will broadcast the block containing it to other miners, who will verify the validation of their received ones. Also, they will check if they are the winner of another lottery, which is designed based on the block they receive. This lucky chance is like followthe-Satoshi procedure. A hash function is suggested to be used N times, which input parameters are the hash value of the received block, the hash of the latest appended block, and a random number, which is randomized N times. N miners who see that they own the N chosen Satoshi will continue the work: the (N – 1) first miners will sign into this empty block, their signature proving that they own the chosen Satoshi, then broadcast this signature. The last miner will include not only the required signature, but also as many transactions as he wants, then broadcast the block to other miners. The block creater and N chosen miners will get equal rewards. Besides preventing a double spending attack, this kind of consensus also encourages miners to stay online to collect the rewards.

Comparisons between PoW, PoS and their hybrid forms

Following table shows the comparisons between PoW, PoS and their hybrid form. As can be seen, using electric power to guess the secret value executing PoW, a lot of money should be spent on not only hardware devices, but also on the energy to execute them. In contrast, PoS does not employ any puzzle, so these two investments are much lower.

Criteria	PoW	PoS	Hybrid form of PoS and PoW
Energy Efficiency	No	Yes	No
Modern Hardware	Very Important	No need	Important
Forking	When two nodes find the suitable nonce at the same time	Very difficult	Probable
Double Spending Attack	Yes	Difficult	Yes, but less serious than in PoW
Block creating speed	Low, depends on variant	Fast	Low, depends on variant
Pool mining	Yes, but it can be prevented	Yes, and it is difficult to prevent	Yes
Example	Bitcoin	Nextcoin	PPcoin,Blackcoin

In PoW, forking can happen if two miners find a suitable nonce at the same time. Meanwhile with PoS, it is very difficult, happening only when a miner can own up to 51% of all stake in the whole verifying network. By employing both PoW and PoS, although their hybrid form can still make a fork, the probability of causing a double spending attack is much lower, compared to the pure PoW. Considering the speed of creating a new block in the chain, overall almost all variants of PoW require much time to append a new block to the chain, while with PoS, the block creating speed is lower, because no node has to solve any puzzle. What could make PoS less attractive is the work with pool mining: for PoW, the miners joining pool are trackable, and the fraudulent work is preventable. But in PoS, it would be very difficult to prevent the miners inside a pool delegating their stake to a single miner as pool operator.

Other Less Popular Consensus Algorithms

Blocki and Zhou pointed out the same problem as King : The PoW wastes too much energy in finding the nonce, and also it is meaningless in everyday human life. Worse yet, it is not fair for the miners having poorer condition to purchase modern hardware, which makes their chance of mining a new block very low. Therefore, Blocki and Zhou proposed using some kinds of puzzle for education and social activities, which would be easy for computers to solve, but difficult for human to solve. It is for the human to solve the puzzle to mine a new block, instead of using hardware. This difference would be fair for everybody, whether they could or could not invest in new modern equipment.

Proof of burn and proof of space are other kinds of proof-based consensus algorithms that do not employ the idea of PoW and PoS. In proof of burn, miners have to send their coins to an address to “burn” them, which means these coins could not be used by anyone else. The miner who burns the largest amount of coin during a duration will be the one getting the right to mine a new block. With Proof of Space, miners will have to invest their money on hard disk, which is much cheaper than computing devices for PoW. During the work, the proof of space algorithm will generate many large datasets called plots on the hard dish. The more plots a node has, the more chance he will get to mine a new block. Proposed by Intel, proof of elapsed time is used in a blockchain platform called Sawtooth Lake. This kind of consensus is executed in a special environment called the trusted execution environment (TEE), with a special device of Intel known as Intel Software Guard Extensions (XGS). In order to conduct the consensus algorithm, each miner will at the same time request a wait-time from a trusted enclave, which is also known as a trusted function inside XGS hardware. Subsequently, all the miners will receive their responded wait-time from the enclave, and together will wait until their received time elapses. When a miner waits enough time, but has found no one has finished the waiting match, he will broadcast to all other miners that he is the winner, which provides him a chance to mine new block. In short, the miner owning the shortest received wait-time will be the one to mine a new block. In order to support this kind of consensus, two functions are used: function CreateTime will inform the miners of the time they need to wait, and function CheckTimer will check whether or not the miner has waited enough time. Because this consensus is executed in TEE provided by XGS devices, it is supposed that cheating with the work of the two above functions is very difficult. Milutinovic et al. proposed a kind of consensus, which is also executed in TEE and with XGS devices, called proof of luck. To execute it, after all the ledgers from all the miners are synchronized, each miner will create for themselves a new block to append to their current chain, then a random number ranged from 0 to 1 will be assigned for each created block, which could be considered a lucky value. All of the nodes will have to agree that the chain with the total largest lucky value would be the main chain. As a result, proof of luck is considered to be fair for all the miners. Furthermore, it would be

very difficult to make attacks, like double spending attack, because the attacker should be very lucky to perform their illegal actions successfully. Multichain is a kind of consensus algorithm, which is quite similar to PoW with the work of mining and fork handling. However, multichain does not apply PoW for choosing the node to append block to the chain; instead, it applies a round-robin schedule to make nodes create blocks in rotation. Specifically, a parameter p ($0 < p < 1$) called the mining diversity is used. In each phase, all the nodes have to wait for a duration, then start checking their rights to append new block to the chain. If among p*N (N is the number of participating nodes) blocks that are created most recently, no block is created by a supposed node A, then node A could append his proposed block to his current chain, then broadcast this block to other miners. In the case that fork happens, like PoW, the longest one would be chosen.

Voting Based Consensus

In order to execute the voting based consensus algorithm, the nodes inside the verifying network should be known and adjustable, so that they can exchange the message easier. This is the main difference compared to proof-based consensus algorithms, which nodes are often free to join and withdraw from the verifying network. Also, in voting-based consensus algorithm, besides maintaining the ledger, all the nodes in the network would have to verify together the transactions or blocks. They will communicate with others, before deciding to append their proposed blocks to their chain or not. In almost all of these variants, nodes are required to see at least T nodes having the same proposed block with them to do the appending work (T is a given threshold). Executing voting-based consensus algorithms is very similar to traditional methods for tolerating faults used in the distributed system. Therefore, voting based consensus should be designed to resist some bad cases: • Some nodes are crashed. • Some nodes are not only crashed, but also subverted. In crashing cases, nodes will wait for the messages from other nodes. However, there are some nodes that do not run, then the normal nodes do not receive enough evidence to make the decision. Therefore, in order to prevent the crashing case with f nodes, there should be at least f + 1 nodes operating normally. In subverting cases, nodes could act strangely, which could make the results inaccurate. This problem can be presented by a classical problem called Byzantine generals proposed by Lamport et al. a group of Byzantine generals attacked an enemy's camp. They decided to divide their army into N groups led by N generals, which would attack the enemy from different sites. If they attacked at the same time, they would win; otherwise, they would lose. Consequently, they had to make an agreement with each other about the attacking time by exchanging messages, and following the decision of the majority. Unfortunately, there were some traitors inside the general group, and they wanted to cheat other generals by telling different decisions to the others. Therefore, the results could be made inaccurate, which made some generals attack, while others did not, leading to failure. Lamport et

al.have proved that in order to tolerate f subverted generals, there should be at least another 2f + 1 normal generals to accompany them. Coming back to the Blockchain, when each node executes the consensus work, some nodes can be subverted, which sends different results to other nodes. Then the worst case is the network could not resist them, causing the ledger to be different in different nodes. Considering the crashing cases, if nodes are crashed, then they could not send their results to other nodes, which makes it difficult to make the final decision. Consequently, based on these bad situations, the voting based consensus algorithms could be classified into two main kinds:

• Byzantine fault tolerance based consensus: a kind of consensus that could prevent the cases of crashing nodes and subverted nodes.

• Crash fault tolerance based consensus: a kind of consensus that could only prevent the cases of crashing nodes.

All consensus algorithms in these two sub-categories will have to make a trust assumption: among N nodes, there should be at least t nodes (t < N) operating normally. While in crash fault tolerance-based consensus, t is usually set equal to [N/2 + 1], in Byzantine fault tolerance-based consensus, t is usually assigned equal to [2N/3+1].

Comparison Between Vote-Based Consensus and Proof-Based Consensus Algorithms

Criterion	Vote-based consensus algorithms	Proof-based Consensus algorithm
Agreement making basement	From majority of the node decisions	Following nodes performing enough proof (PoW, PoS, etc.)
Nodes can join freely	No	Mostly
Number of nodes executing	Limited	Mostly unlimited
Trust	Less trustful	More trustful
Node identities are managed	Yes	No
Security threat	Less serious	More serious
Award	Mostly no	Yes

Moreover, vote-based consensus is often conducted in private and consortium Blockchain, in which the decentralization degree is lower than in public Blockchain with proof-based consensus. Consequently, trust in the proof-based one would be higher. Clearly, the more freely that nodes can join the verifying network, the more decentralized the verifying network is. In contrast, a tradeoff is recorded between the freedom and security issues. The more nodes that join the verification network, the more risk the verification network has. In vote-based consensus, all the nodes are manageable, which makes it safer to control the security risk. While with the proof-based consensus algorithm, some threats, like double spending attack, could always possibly happen. Finally, with proof-based consensus algorithm, in order to encourage nodes to maintain the ledger, award is often given to any nodes who mine a new block. Following Satoshi, the more blocks that are in the chain, the fewer awards a node can receive after mining a new block in the future. We suppose that lately, if the mining award and the incentive for verifying transactions are too small, nodes will have no motivation like current time to maintain the ledger anymore. Meanwhile, with vote-based consensus, there is no necessity to give any award to a node. Using decentralization to increase the reliability is the only aim for this kind of algorithm.

Conclusion

We can derive the following conclusions from our discussion in the previous sections:

1) We discussed the limitations of a Proof of Work (PoW) based blockchain which is currently used by a large number of cryptocurrencies.

2) We also introduced a new approach to solve problems of consensus using Proof of Stake (PoS). The merits and demerits of both the systems as well as the different possibilities associated with either were listed.

3) Lesser known consensus strategies like Proof of Delegated Stake and Proof of Burn were also touched upon.

4)To enable readers to see the bigger picture, we categorised consensus algorithms into two main kinds: proof-based and vote-based consensus algorithms.

(i) In the former, nodes have to show they have performed sufficient proof to get the right to do the appending work, and get the rewards.
(ii) Meanwhile, in the latter, nodes will exchange messages with others to make an agreement about the blocks or transactions to be appended to the ledger.

We also make comparisons between these two types based on some of their highlighted characteristics, which illustrates the advantages and drawbacks of each category.

5) It could be observed that besides the original public Blockchain with proof-based consensus algorithms, the newly developed consortium and private Blockchain has much potential with vote-based ones at this time.

6) But in the end it is more likely that a hybrid system or something close to it will prevail in the long run that provides ample opportunities to people with low resources but also provides sizeable returns to people willing to invest large sum of resources in the blockchain.

7) If properly utilised and implemented, these consensus algorithms will not only lead to a better platform for cryptocurrencies, but also for other use cases for blockchain like loan tracking and decentralised DNS provisions.

References

[1] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” 2008 [Online]. Available: https://bitcoin.org/ bitcoin.pdf.

[2] Bitcoinwiki, “Genesis block,” 2017 [Online]. Available: https://en.bitcoin.it/wiki/Genesis_block.

[3] Ethereum [Online]. Available: https://www.ethereum.org.

[4] S. Popov, “A probabilistic analysis of the Nxt forging algorithm” ,2016.

[5] Bitcoinwiki, “Proof of Stake,” 2014 [Online]. Available: https://en.bitcoin.it/wiki/Proof_of_Stake.

[6] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, “Proof of activity: extending bitcoin's proof of work via proof of stake,” ACM SIGMETRICS Performance Evaluation Review, vol. 42, no. 3,2014

[7] “How to timestamp a digital document” by Haber and Stornetta, available at https://www.anf.es/pdf/Haber_Stornetta.pdf , 1991

[8] "A Digital Signature Based on a Conventional Encryption Function" by Ralph Merkle available at https://people.eecs.berkeley.edu/~raluca/cs261-f15/readings/merkle.pdf , 1979

[9] “A Survey Paper on Blockchain Technology, Challenges and Opportunities” by Sandeep Kumar, Abhay Kumar, and Vanita Verma, 2019

[10] “Bitcoin Mining and its Energy Footprint” available at https://www.researchgate.net/publication/271467748_Bitcoin_Mining_and_its_Energy_Footprint by David Malone and K.J. O’Dwyer, 2014

[11] “On the Security and Performance of Proof of Work Blockchains” available at https://eprint.iacr.org/2016/555.pdf by Arthur Gervais, Ghassan O. Karame, Karl Wüst, Hubert Ritzdorf ,Srdjan Capkun

[12] “Decentralized mining in centralised pools” available at https://www.researchgate.net/publication/323963835_Decentralized_Mining_in_Centralized_Pools by Lin William Cong, Zhiguo He, and Jiasun Li, 2018

[13] “Formal Analysis of a Proof-of-Stake Blockchain” available at “https://www.researchgate.net/publication/330030317_Formal_Analysis_of_a_Proof-of-Stake_Blockchain” by Wai Yan Maung Maung Thin, Naipeng Dong, Guangdong Bai, Jin Song Dong

[14] “Delegated Proof of Stake with Downgrade: A Secure and Efficient Blockchain Consensus Algorithm with Downgrade Mechanism” available at https://www.researchgate.net/publication/335186093_Delegated_Proof_of_Stake_with_Downgrade_A_Secure_and_Efficient_Blockchain_Consensus_Algorithm_with_Downgrade_Mechanism by Fang Yang,Wei Zhou, Qingging Wu, Rui Long,Naixue Xiong, Meiqi Zhou

[15] “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake” available at https://www.semanticscholar.org/paper/PPCoin%3A-Peer-to-Peer-Crypto-Currency-with-King-Nadal/0db38d32069f3341d34c35085dc009a85ba13c13 by Sunny King and Scott Nadal, 2012

[16] Pavel Vasin. 2018. BlackCoinąŕs Proof-of-Stake Protocol v2. (2018)

[17] “TwinsCoin: A Cryptocurrency via Proof-of-Work and Proof-of-Stake” available at https://dl.acm.org/doi/abs/10.1145/3205230.3205233 by Tuyet Duong, Alexander Chepurnoy, Lei Fan, Hong-Sheng Zhou

[18]“Proof of Activity” available at https://www.researchgate.net/publication/286247116_Proof_of_Activity by Iddo Bentov, Charles Lee, Alex Mizrahi, Meni Rosenfeld

[19] J. Blocki and H. S. Zhou, “Designing proof of human-work puzzles for cryptocurrency and beyond,” in Theory of Cryptography. Heidelberg: Springer, 2016, pp. 517-546.

[20] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, “Proofs of space” in Advances in Cryptology– CRYPTO 2015. Heidelberg: Springer, 2015, pp. 585-605.

[21] “Proof-of-Burn” available at https://eprint.iacr.org/2019/1096.pdf by Kostis Karantias , Aggelos Kiayias, and Dionysis Zindros

[22] M. Milutinovic, W. He, H. Wu, and M. Kanwa, “Proof of luck: an efficient Blockchain consensus protocol,” in Proceedings of the 1st Workshop on System Software for Trusted Execution, New York, NY, 2016.

[23] M. Castro and B. Liskov, “Practical Byzantine fault tolerance,” in Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, LA, 1999, pp. 173-186.