Hindawi
Security and Communication Networks
Volume 2021, Article ID 6693731, 22 pages
https://doi.org/10.1155/2021/6693731

Review Article

A Research Survey on Applications of Consensus Protocols in Blockchain

Sivleen Kaur, Sheetal Chaturvedi, Aabha Sharma, and Jayaprakash Kar

Centre for Cryptography, Cyber Security and Digital Forensics, Department of Computer Science & Engineering, The LNM Institute of Information Technology, Jaipur, India

Correspondence should be addressed to Jayaprakash Kar; jayaprakashkar@lnmiit.ac.in

Received 22 October 2020; Revised 25 November 2020; Accepted 24 December 2020; Published 22 January 2021

Academic Editor: Debiao He

The concept of blockchain, widely known as virtual currencies, saw a massive surge in popularity in recent times. As far as the security of the blockchain is concerned, consensus algorithms play a vital role in the blockchain. Research has been done separately, or comparisons between a few of them have been presented previously. In this paper, we have discussed widely used consensus algorithms in the blockchain. The consensus protocols covered in this paper include PoW (Proof of Work), PoS (Proof of Stake), DPoS (Delegated Proof of Stake), PoET (Proof of Elapsed Time), PBFT (Practical Byzantine Fault Tolerance), and PoA (Proof of Authority). For each consensus, we have reviewed the properties, applications, and performance in the blockchain.

1. Introduction

People have been involved in trade since the beginning of the era. An early trade form, *barter system*, saw the direct exchange of goods and services for other goods and services among people. As time evolved, the invention of physical currencies has greatly simplified and promoted trade, but these currencies had their own challenges involving a lot of fraudulent activities such as exchange of fake currencies. As a result of which, global economy inevitably started moving towards the digital ecosystem which involved electronic transaction and money transfer through banks, but even this advancement had its own shortcomings. The involvement of a third party such as banks in these transactions incurred a subtle amount of charges on users and still had chances for deceitful activities. Thus, people across the globe started making attempts to completely decentralize the process of value exchange. A major breakthrough was made, when a pseudonymous person named Satoshi Nakamoto presented the idea of *Bitcoin*, a cryptocurrency, in his paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" [1]. As a part of its implementation, Nakamoto devised the idea of blockchain to achieve decentralization for the peer-to-peer transfer of bitcoin and used consensus protocols in order to achieve agreement on various decisions between the entities of bitcoin blockchain network. Although the idea of blockchain was proposed in 2008 for bitcoin implementation, consensus protocols have been in existence since the 1970s [2]. In the 1970s, a group of computer scientists including Leslie Lamport and Barbara Liskov began thinking about the solution for the problem:

> *"Is it feasible for a set of machines to come to an agreement and how?"*

This simple question gave rise to a long and complex problem known as "consensus" and became fundamental to the study of decentralized systems. For many years after their invention, only one family of consensus protocols was known, namely, "classical consensus protocols." Protocols belonging to this family can be easily identified due to one property: all-to-all voting. In 2008, Satoshi introduced the second family of these protocols, known as Nakamoto, or longest-chain, consensus protocols. After this, a number of variations in the classical and longest chain protocols were proposed over the years. In 2018, another family of consensus protocol came into existence, called the Snow family. These algorithms use a novel mechanism for reaching an agreement: instead of all-to-all voting, these protocols randomly select participants and ask about the state of the network. They merge the best of both classical and Nakamoto: they confirm transactions in a few seconds and operate at thousands of transactions per second, while also enabling thousands to millions of participants in the network. A blockchain, in simple term a chain of blocks, is a growing list of records called *blocks*, which are joined together using cryptographic techniques. Blockchain, also known as the *DLT (distributed ledger technology)* [3], through the use of decentralization and the cryptographic hashing techniques makes the history of digital assets unalterable and transparent. It was primarily invented to serve as the public transaction ledger for the bitcoin cryptocurrency. Bitcoin transactions are performed and saved using this distributed ledger on a shared network and are open, anonymous, and oftentimes public. But over the years, this technology has found varied applications in different fields.

The blocks of a blockchain specifically consist of two parts: block information and block header. Block information consists of the list of *transactions*, i.e., a record of events, and also the information regarding these transactions like the value transferred and time of transaction along with the information about who is participating in these transactions using a unique "*digital signature*" without revealing their actual identities. The block header fundamentally consists of the timestamp when the current block gets created, *nonce*, and *Merkle root*. Merkle root is a hash of all the nodes of a *Merkle tree*. Merkle tree is a mash-up of the binary tree and linked list data structure with some hashing for security, which provides efficient and secure verification of transactions in the blocks of a blockchain. A reference to the previous block to which the current block has to be added is also included in the block header unless it is a *genesis block* (Figure 1). A genesis block is the starting block in any blockchain network and is hardcoded by its creators when the blockchain is first started [4, 5].

1.1. Properties of a Blockchain Network. The three main pillars [6] of blockchain technology which helped it in gaining widespread acclaim are as follows:

(i) Decentralization: no single entity has the rights and power over the information stored in the blockchain. Each of the network entity owns this information, and any changes in this chain of information can only be made after achieving an agreement between all.

(ii) Transparency: anyone inspecting a basic *public blockchain* and participating in a *private blockchain* network is capable of viewing every transaction and its associated details.

(iii) Immutability: once the block containing a set of transactions gets added to the blockchain, it is nearly impossible to alter these transactions. Blockchain achieves this property by using the cryptographic hash functions.

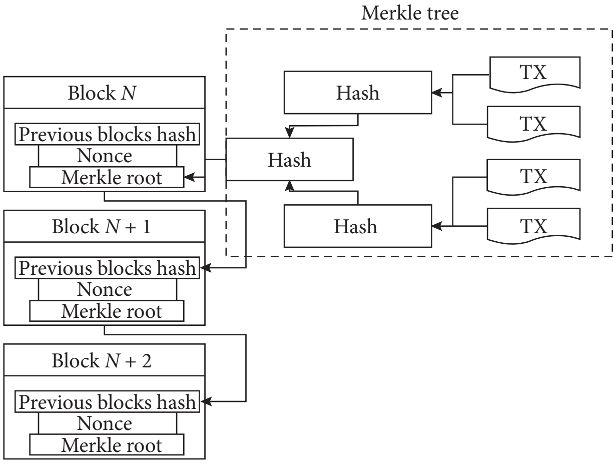

FIGURE 1: Hash generation of a block in a blockchain.

Apart from these, few other features which are fundamental to every blockchain network are as follows:

(i) Distribution: blockchain is a type of distributed ledger; i.e., all the information which is stored in the blockchain is distributed across the nodes of the network.

(ii) Security: since blockchain technology gets rid of the need for any central authority, no group can directly change any of the network characteristics for their benefit. Moreover, the use of encryption and other cryptographic techniques ensure another layer of security for the system.

(iii) Scalability: scalability defines the ability of the blockchain to maintain smooth operations ruling out the possibility of slow processing time, system bloating, lags, etc. even for an enormous network. A blockchain gets bigger with its popularity and demand, so it should be able to handle millions of transactions per second.

1.2. Types of Blockchain Networks. The content stored in the blocks and the activities performed by the various participants in the blockchain network are accessible and controllable depending upon how we configure the blockchain and how we expect it to meet the desired purpose. Broadly speaking, based on this, we can categorize the blockchain network into three types listed subsequently [3, 7]:

(i) Public blockchain: a public blockchain is an open-to-all network. Due to its permissionless nature, any willing individual can join this network and view, read, and write data in a block and add it to the blockchain. The information stored in blocks of such a blockchain is available in the public domain. These blockchains operate in a truly decentralized and authority-free manner.

(ii) Private blockchain: a private blockchain which is also known as consortium blockchain allows only selected entries of verified participants. It is usually implemented for private businesses. A private blockchain is not truly decentralized and is simply a distributed ledger that operates as a private, secure database based on cryptographic concepts.

(iii) Permissioned blockchain: a permissioned blockchain stands somewhere between the public and private blockchain and provides a lot of customization options, which could include allowing anyone to join the permissioned network after appropriate verification of their identity. These network participants are then allocated with selected and designated permissions allowing them to perform only certain activities on the permissioned network.

1.3. Working of a Blockchain. Nodes present in a blockchain network either act as *miners* and create new blocks or act as *block signers* who validate and digitally sign the transactions [4]. When a new transaction is made or records are created, they must be stored in a block and added to the blockchain. However, in order to create and add a block to a blockchain, the following steps are used:

(i) A transaction must occur, i.e., some information must be transferred from one party to another.

(ii) The transactions to be added in a block must be verified.

The actual working of a blockchain network varies depending on the application for which it has been developed. However, the basic steps involved in adding a new block to blockchain have been explained in detail taking reference from the *bitcoin* cryptocurrency, for which it was originally developed, in the subsequent paragraph.

As soon as a transaction occurs between two blockchain users, it gets added to a pool of unverified transactions. These unverified transactions are then broadcasted to all the participating validator nodes in the blockchain network where they are checked and validated against some validation rules which are setup by the creators of that blockchain. However, in the bitcoin blockchain, the process of transaction verification is performed by the mining nodes only. The mining nodes then put the verified transactions into a block, which is then sealed up using a hash. Bitcoin hashes the block data into a 256 bit number using the SHA-256 cryptographic hash algorithm. All transaction data inside the block along with the previous block's hash to which this block has to be added are hashed together by the bitcoin miners to generate a 256 bit number that uniquely identifies the block (as depicted in Figure 2). However, a bitcoin miner cannot just generate any hash for the block and add it to the blockchain network; i.e., the hash should meet certain conditions in order to get the block accepted by the network. A critical decision that every blockchain network has to make is to figure out which node will commit the next block to the blockchain. This decision is taken up using a *consensus mechanism*. To be able to add the next block to a bitcoin blockchain, a miner has to win the competition to find a *correct hash* by solving a complex mathematical problem. The math problem requires miners to produce a hash with a certain amount of leading 0s. The first miner to achieve this for the block will be the winner and would be able to add the created block to the existing blockchain. Since the transaction data and previous block's hash inside the block are unalterable, in order to get the desired hash, miners need to add a variable number to the block which miners can change continuously until they find a variable that solves the above-stated math problem. This variable number is called a *nonce*. After a miner creates a block hash successfully, the miner broadcasts the block to all its peer nodes which receives and validates the new block. This block is then added to the blockchain and is sent out across the network where each of the network nodes adds this block to their copy of the blockchain, extending the blockchain's height by 1. When mining nodes receive and validate the block from some other nodes, they stop their efforts to find a block at the same height and immediately start computing the next bitcoin block in the blockchain [8].

2. Consensus Protocols

In a centralized setup, only one entity dominates the system, i.e., exercises power over the entire system. In most scenarios, they are allowed to make the changes according to their choice—a complex governance system is not required to reach consensus among the administrators. But in the case of a decentralized setup, it is an entirely different story. In a distributed system due to the lack of a central governing authority, all the participants in that network should collectively decide and agree upon what is best for the network. Achieving this in an environment where strangers do not have confidence in each other was probably the most important development that paved the way for blockchain and led to a set of algorithms to achieve consensus amongst the participating nodes. Consensus algorithms are protocol sets which provide a technique with the help of which the users or machines can coordinate in a distributed and decentralized setting. It is needed to ensure that all entities in the system collectively agree upon one thing (single source of truth) even if some entities fail individually. This mechanism aims to make the system fault-tolerant. This mechanism is devised in order to achieve reliability in a network consisting of unreliable nodes. While voting works on the majority rule, neglecting the well-being and the sentiments of the minority, a consensus ensures that an agreement is reached which might benefit the complete network. Thus, consensus algorithms do not merely agree with the majority votes but also agrees to one that profits all of them. So, it is always a success for the network.

Various types of consensus algorithms have been devised over time for varying applications, but all of these algorithms must hold these properties for tolerating halting failures:

(1) Termination: the process of achieving consensus on a given data value should come to an end; i.e., eventually, every correct node must decide some value [9].

(2) Agreement seeking: each consensus protocol should try to bring about as much agreement as possible from the network [9].

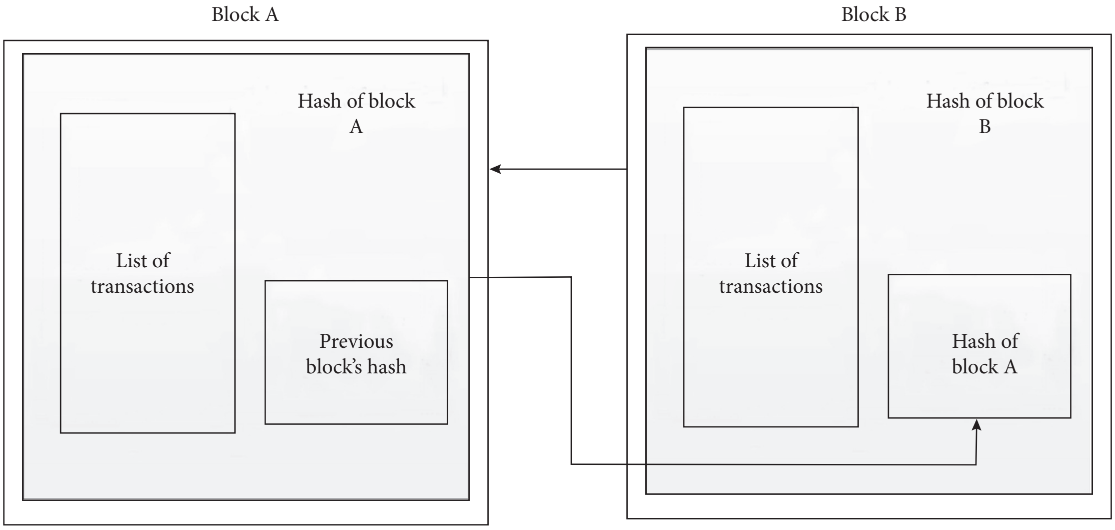

Figure 2: Hash generation of a block in blockchain.

(3) Collaborative: the aim of the participants of the system should be to work in unison for the welfare of the group by achieving a result that favors the best interest of the group.

(4) Cooperative: the participants should not put their interests first and work as a team more than individuals.

(5) Egalitarian: a system that is trying to achieve consensus must be as *egalitarian* as possible; i.e., weightage of every vote should be equal. In simple words, one vote cannot be less important than another.

(6) Inclusive: for a system to reach consensus, it should try to involve as many entities as possible in the process. It should not be similar to normal voting; i.e., it should not be the case that certain entities do not vote because they feel that it is not worthy to cast their vote as it will not have any weightage in the long run.

(7) Participatory: everyone should participate actively in the entire process of consensus.

(8) Integrity: if v is the value that is decided by the majority of correct processes, then that same value (v) must be decided by any correct process [9].

Different applications follow a different definition of integrity. For instance, systems with the decision value equal to a value proposed by some correct process (not necessarily all of them) follow a weaker type of integrity. Over the years, consensus protocols have found applications in various fields like state machine replication, state estimation, load balancing, control of UAVs (and multiple robots/agents in general), smart power grids, clock synchronization, opinion formation, and others where they have been modified according to their use. But one of the most important and evident applications of the consensus protocols can be seen in the blockchain. As the consensus algorithms maintain the security and integrity of distributed computing systems, they are essential elements for blockchain networks. Blockchain, an open decentralized system, has each node acting both as a host and as a server and they need to share information among all the nodes of the system to reach a consensus to carry out transactions. Anyone can be a node and remain anonymous in a *public* blockchain network, as no permissions are required to become a part of the network and to contribute to its upkeep. Therefore, a node can alter transactions and include them in a new block. Thus, the blockchain can end up with a fork. For instance, in the chain, one fork contains only valid transactions, while the second one contains the tampered transaction. Public blockchain protocols need to deal with this issue independently to maintain the decentralization of the network. Transactions cannot be declared valid or invalid unilaterally by a single participant. To avoid forks and tempering of blocks so that everyone agrees to a single version of the truth, various consensus algorithms are used. Different sorts of blockchains have different application scenarios. Thus, the consensus algorithms adopted by blockchain need to be suitable and should fit the demands of its specific application. In the subsequent sections of this paper, we will see various consensus protocols and their applications in the blockchain.

3. Proof of Work

Proof of Work (PoW) is a mechanism to achieve consensus in a blockchain network and is the underlying consensus model of various cryptocurrencies like bitcoin and Ethereum [8]. It describes a system that requires a considerable amount of effort to be done for mining a new block in order to prevent malicious uses of computing power and other possible attacks on the system such as denial-of-service attacks and other service abuses like spamming by making the service requester do some demanding work. In a blockchain network, the PoW consensus mechanism requires the network mining nodes to prove that the work done and submitted by them qualifies them to receive the right to add new transactions containing blocks to the blockchain. In Proof of Work, nodes that will add the next block to the blockchain are selected in proportion to their computing power; i.e., the nodes compete with each other using their computing power [10].

As discussed earlier, miners in the blockchain network create blocks by calculating the solution to a complex mathematical problem, and the only way to solve this problem is via costly guessing, i.e., *Proof of Work*; hence, this problem is also known as Proof of Work challenge. Since miners freely enter and leave the network, in a bitcoin blockchain, the difficulty in this challenge is adjusted every 2,016 blocks to keep a 10-minute interval between the mining of two blocks by the same miner that ensures the decentralization in the verification process across the entire network. This adjustment is done automatically by the protocol by increasing or decreasing the target hash based on the number of miners. Thus, in a distributed consensus, based on the Proof of Work, miners require a lot of energy and they have to incur a substantial amount of cost in hardware and electricity to carry out such heavy computations.

It is also possible that two mining nodes generate blocks at the same time. This situation occurs because the process of acceptance of blocks in a blockchain network is not instantaneous. This time lag in accepting a block may lead to another miner finding the correct hash for a block at the same height in the blockchain leading to a short-term fork in the blockchain network. In such a situation, for mining new blocks, nodes try to decide which of the two newly identified blocks in the two forks it wants to consider. Later, for security, the protocol behind blockchain considers the longer of the two forks valid as it has a larger share of Proof of Work (POW) and hence more confirmations. Blocks present on the other fork are discarded and referred to as orphan blocks [11].

Another way a fork is created is when a hacker with sufficient computational power to dominate the network attempts to reverse a few of the transactions or tries to double spend a coin. PoW provides an efficient mechanism to deal with such double-spending problems. Suppose two transactions are made simultaneously in order to spend a single coin, and both of these get into the unconfirmed pool of transactions. Let us say the block containing the first transaction got validated by the miners before the block containing the second transaction. In this case, the second transaction will not get validated because the miners will consider it invalid, and it will be pulled from the network. But what if both the transactions are taken simultaneously by the miners? In this case, blocks containing both the transactions are added to the blockchain creating a fork. If the blockchain network is completely decentralized, then this fraudulent fork will become impractical over time because a miner has a low probability of consistently winning the next block and add it to the fraudulent fork to increase the number of conformations on it, and if the adversary tries to get an adequate amount of conformations, then he will have to go back and reverse transactions in all the blocks added after the fraudulent block which would require a lot of work and thus is computationally and financially infeasible as discussed below.

Since each block in a blockchain contains a reference to its previous block to which it has been added if miners attempt to alter the blockchain or subvert the mining process, damaging bitcoin's credibility in the process, their expenditure on the computing resources such as hardware and electricity would comparatively be costlier than the *Return of Investment* (ROI) obtained in form of block mining reward. Thus, it becomes extremely difficult to alter any aspect of the Proof of Work consensus-based blockchain ensuring that no one can intrude in the system or temper the data saved in the blocks because such an alteration would require remining all succeeding blocks. Apart from this, PoW also makes it extremely hard to monopolize the network's computing power by a user or group of users since the generation of hash requires very expensive hardware resources and a considerable amount of electricity.

3.1. Applications. Proof of Work consensus algorithm is widely applicable in a lot of cryptocurrencies and other blockchain systems. We have already seen the application of and working of PoW in bitcoin above. In this section, we will look upon a few more PoW-based blockchains and cryptocurrencies.

(i) Litecoin: Litecoin is a cryptocurrency that enables fast, near-zero cost transactions across the world. Litecoin was originally developed as an improvement over bitcoin. Although both Litecoin and bitcoin use the Proof of Work concept for mining, the algorithms behind it used by them are very different. Litecoin uses *scrypt*, a memory-intensive algorithm. The main idea behind using this algorithm was to ensure anyone could participate in the network mining process by ruling out the dependency on high computing resources such as those used in bitcoin and replaces it with memory-intensive CPUs. Litecoin also reduces the amount of time required to confirm a new transaction to 2.5 minutes from 10 minutes in bitcoin, making itself capable of handling higher transaction volume than its counterpart—bitcoin [12]. However, there is a fixed supply of Litecoin; thus, there will ultimately be only 84 million of these in circulation.

(ii) Ethereum: Ethereum cryptocurrency's mining process is almost the same as bitcoin. However, the PoW algorithm used in the Ethereum frontier network, called *Ethash*, is somewhat different from that of bitcoin and was created specifically for Ethereum [13]. The foremost reason behind developing a new Proof of Work algorithm rather than using the existing one was to eradicate the problem of *mining centralization* caused because of the dependency on hardware resources and to create a mining system which could easily be embedded on common hardware. Technically speaking, with Ethash, developers aimed to create a network resistant to application-specific integrated circuit (ASIC), specialized chips particularly designed to outperform standardized computer hardware by many orders of magnitude in hashing performance, and use of which is the only profitable way to mine bitcoin blocks now [14]. Ethash achieves this by providing a PoW algorithm for which the commodity hardware to the miners is already highly optimized, and hence, the addition of an ASIC to it will give very little advantage over simply using the latest commodity hardware. Such hardware is designed to achieve *memory hardness* as one of its properties. Memory hardness primarily means that the performance of a computer system is defined as its ability to the data around in memory rather than by how fast and efficient it is in performing calculation operations. Graphic processing units (GPUs) tend to provide such mining hardware.

Apart from these, several other cryptocurrencies including bitcoin cash, Zcash, and bitcoin SV make use of the Proof of Work consensus algorithm for their implementation.

3.2. Performance. PoW consensus has several advantages and downfalls surrounding it that have been discussed in this section.

3.2.1. Advantages.

(i) Most battle-tested: being the oldest one and implemented in the first cryptocurrency, this consensus mechanism has weathered several challenges to its security and stability. Although, in theory, other mechanisms can be considered superior, they are at a disadvantage in the sense that they simply have not been in active use long enough to definitely prove it.

(ii) Reaches consensus quickly: the key characteristic of PoW is difficult to find a solution to the complex math problem but extremely easy to verify. Hence, once a hash is created, it can be verified easily and the consensus is reached quickly.

(iii) Deter spammers: since PoW requires a considerable amount of work to be done for each process say sending an e-mail, most spammers will not have enough computational power to send a number of unsolicited emails. Even if a spammer has enough computational power, the cost associated with it is likely to exceed the profit made from spamming.

3.2.2. Disadvantages.

(i) Electricity dependency and wastage: PoW requires a lot of computing power and a huge amount of electricity is wasted in this process as all mining nodes attempt to solve the complex problem, but only one is able to mine a block. Moreover, access to electricity is not uniform all across, thus allowing miners in regions with cheaper available electricity to monopolize the mining industry.

(ii) Centralization: due to the dependency on electricity and mining hardware, the Proof of Work consensus mechanism is tending towards centralization. The PoW network's hash rate is heavily concentrated in areas having inexpensive and plentiful hydroelectric power and mining hardware from local suppliers. This is threatening the PoW network and all of the data locked in it.

(iii) Less secure for small networks: PoW-based blockchains provide adequate security only if there are a large number of miners in the blockchain network competing to mine the next block. But if the network is small, the possibility of a hacker gaining a simple majority of the network's computational power and mining a fraudulent block goes up.

3.3. Security. In the Proof of Work consensus mechanism, network centralization makes the network vulnerable to the 51% attack which in turn breaches the network to numerous other security problems. 51% attack is a potential attack on the blockchain network, where a group is able to concentrate the majority of hash rate of the network, consequently gaining the power to falsely validate transactions and controlling the network. The attackers would also be able to halt payments between some or all users, by preventing the new transactions from gaining confirmation resulting in *denial-of-service*, and prevent some or all other miners from mining new blocks leading to a *mining monopoly*. A successful 51% attack could also pave the way for the attacker to double-spend coins by reversing the completed transactions, while they are in control of the network leading to the problem of *double-spending* [15].

3.3.1. Working of 51% Attack. Suppose a fraudulent miner makes a transaction that is included by the trusted miners in their block and added to the blockchain. However, at the same time, the attacker also mines a block that does not contain his transaction and gets it validated accounting for the domination over the network. This block also gets committed to the chain creating a fork. This fraudulent node is not broadcasted across the network by the attacker, and the attacker keeps on adding new blocks to this fork until it becomes bigger than the other branch. Once it gets bigger, the attacker broadcasts it to the entire network and all the trusted miners are forced to abandon the previous branch and accept this fraudulent branch that does not contain the transaction made by the attacker. Thus, the attacker still has access to the coins and can spend it again. The visual working of this attack is depicted in Figure 3.

Although it is believed that owing to the financial resources needed to perform a 51% attack and the size of the bitcoin network, it is unlikely to be performed because once a blockchain grows large enough, the likelihood of a group obtaining adequate computing power to overwhelm all other participating nodes drops rapidly and altering the previously confirmed blocks also gets more difficult because the blocks are all linked with each other through cryptographic hashes

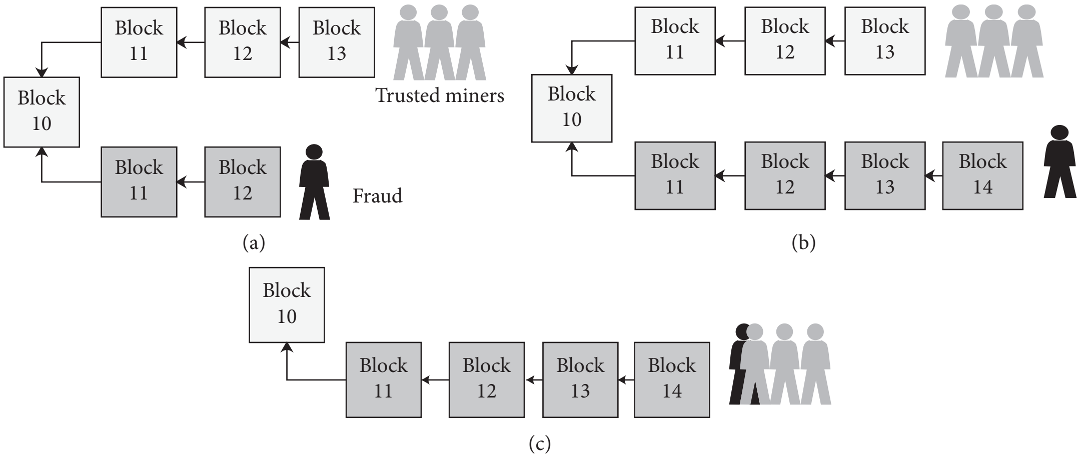

Figure 3: 51% attack in a blockchain: (a) the attacker starts creating its own private branch of the blockchain; (b) once the fraudulent chain is bigger, attacker broadcasts it to the entire network; (c) the actual chain is abandoned because its blocks had lesser conformations.

and changing a few of these will result in an imbalance in the chain. Hence, a successful attack if occurs would likely only be able to alter the transactions of a few recent blocks in the chain and only for a short duration of time. However, the possibility of the occurrence of this attack can never be ruled out.

A PoW-based system is also open to selfish mining attack where a mining node, instead of publishing the generated block after adding it to the blockchain, keeps its blocks private, and publishes it gradually. Researchers have proved selfish miners exercising with approximately 33% mining power can effectively earn 50% of the mining power as a consequence of this attack [16].

Apart from these, a PoW-based framework is vulnerable to eclipse attack. Although a PoW-based blockchain is scalable and can easily handle a large number of nodes due to the limitations on node bandwidth, an average network node is unable to connect directly to the entire network. Bitcoin software only permits a maximum of 125 connections. This limitation paves the path for an eclipse attack where the malicious attacker establishes all connections possible with the target node. Thus, the target node is completely isolated from the rest of the network and could easily be fed with incorrect transaction data [17, 18].

A few notable examples of cryptocurrencies that have fallen victim to the 51% attack include monacoin, bitcoin gold, and zencash.

4. Proof of Stake

Proof of Stake (PoS) is an alternative to PoW as it is more energy efficient. The objective of both being the same, i.e., to reach a consensus within the blockchain, method of achieving it is completely different. The Proof of Stake consensus algorithm uses a selection process that is pseudorandom in nature to pick the validator of the subsequent block from the existing nodes. The process is based on a mix of several factors which include randomization and staking age along the node's wealth. In Proof of Stake consensus mechanism, blocks are said to be *"forged"* rather than being termed as mined [19]. While in PoW, the block which first solves complex problem mines the next block and receives rewards; in PoS, the individual node which creates the next block is selected based on how much they have "staked" in comparison to other competitor nodes [20]. The stake is usually based on the number of coins the network node has for the particular blockchain it is attempting to mine. In these systems, the transaction fee is generally the reward, and users who want to be among the participants in the forging process need to lock their stake (a certain amount of coins) in a network. The chances of a node to be selected to forge the next block as the validator depends on the size of their stake, which means that the chances of the node to win the next block increases as its stake increases. But these selection criteria are biased, as the network will be dominated by the single node with the maximum stake. To overcome this issue, more methods are added to the selection process, two of them being *"randomized block selection"* and *"coin age selection."*

(i) In the *randomized block selection* method, the next forger is selected based on a combination of the hash value and stake, and the node with a combination of the highest stake and the lowest hash value is selected. But in this case, generally, the nodes can predict the next forger, the reason being that the size of the stake held by network nodes is public [21].

(iii) In the *coin age selection* technique, the next forger is selected based on how long it has held the stake along with the size of the stake, which is termed as the coin age. It is calculated by performing the multiplication of the number of staked coins by the number of days they have been held at stake. Once a block has been forged by the node, its coin age is again set to zero. And to prevent the blockchain from being dominated by large stake nodes, after forging a block, the node must wait for a particular period before being able to forge another block [21].

When a node gets selected to forge the subsequent block, it checks whether the transactions that are present in the block are valid or not, and if the transactions are valid, the block is signed by the node and finally committed to the blockchain by the node (the same can be visualized, as shown in the 4th and 5th steps of Figure 4). And the transaction fees which are associated with these transactions (that are present in the block) are received by the node as a reward (the same can be visualized, as shown in the 6th step of Figure 4). If a node does not want to be a forger anymore, its stake and the earned rewards are released after a definite period, giving the network time to verify that the node has not added any fraudulent blocks to the blockchain. Moreover, the forger stays motivated for not validating or creating fraudulent transactions because if a fraudulent transaction is detected by the network, then in the future, the forger node will not be able to participate as a forger and will also lose a share of its stake. So till the time the stake is more than the reward, in case of doing any fraudulent activity, the validator would bear a loss, as he will be losing more coins than gaining. In an ideal scenario, the working of the PoS consensus mechanism as discussed above can be visualized as Figure 4.

4.1. Application. Two of the most important platforms using the Proof of Stake consensus mechanism have been discussed below.

4.1.1. Peercoin. Peercoin, proposed in 2012 and formally based on core bitcoin technology, is the first hybrid blockchain that utilizes the PoS mechanism to provide security for the network and PoW mechanism for the distribution of new coins. The mining process is initially handled by the PoW algorithm, but as the mining process becomes more difficult and reward decreases over time, the process of mining starts moving towards centralization. To overcome this problem, the platform in later stages accepts the PoS algorithm for the generation of new blocks. This block generation process in Peercoin is known as minting. The PoS algorithm is also responsible for securing the platform against 51% attack. Peercoin allows any network-connected computer to participate in the blockchain security process because of the nature of efficiency influenced by PoS consensus rules. Thus, by growing the number of security providers, Peercoin is strengthened, and this ensures long-term security [22, 23].

4.1.2. Ethereum 2.0. Ethereum, the second-largest blockchain platform after bitcoin, has planned to shift the protocol, for reaching the state of consensus in this open-source and public ledger, from PoW to PoS, as PoS comes with several improvements over PoW such as it makes Ethereum more secure and energy-efficient as compared to counterparts of Proof of Work [24, 25]. This new form will be known as *Ethereum 2.0* and will require validators to stake 32 ether coins for running a validator node on the network and need to deposit the same to the official *Ethereum 2.0 deposit contract*. Validators will then have to download the Ethereum 2.0 client software and run the same, and while running it, a minimum of 4 to 64 random committees that will be consisting of 128 validator nodes will be selected for proposing and attesting the block, from the pool of validators. Validators who correctly do so will be given a reward of ether coins in respect of the percentage of their stake. If a validator node does not stay online and fails to execute its computational responsibilities' share, then as a result, its block reward will decrease moderately, thus incentivizing validators to stay online as frequently as possible. Moreover, if a validator attempts to compromise the network maliciously, then all or some of its 32 staked ether coins will be slashed [26].

Apart from Peercoin and Ethereum, several other cryptocurrencies make use of PoS such as Nxt, ShadowCash, Qora, and BlackCoin.

4.2. Performance. PoS consensus has several advantages and downfalls surrounding it which have been discussed in this section.

4.2.1. Advantages.

(i) Reduced electricity consumption: as the PoS system does not need users to solve the complex energy-consuming algorithms, it reduces electricity consumption by 99%, as mentioned in 1. Moreover, the network validators do not have to stress about a cheap electricity source while using the energy source from wherever they are, thus increasing the playing field for validation of the network.

(ii) Makes staking easy: Proof of Stake encourages mass participation and reduces the stress on participants of the network; therefore, it makes staking easy. As it does not place excessive demands on stackers for hardware, so the necessity for a mining rig is eliminated. Because of lower demand and strain on stackers, the rate of participation increases. As a result of which, the network becomes more flexible and decentralized.

(iii) Environmentally friendly: as the architecture of the PoS mechanism is simple, the requirement for resources that strain the environment is very less. And for the network to function and issue new coins, mining farms are not required, thus making it more environmentally friendly.

(iv) Decentralisation: Proof of Stake solves the centralization problem of Proof of Work to some extent, as its electricity dependency is very less and it does not require a large amount of hardware. Moreover, it becomes more accessible and environmentally friendly because of the abovementioned reasons.

4.2.2. Disadvantages.

(i) In the PoS mechanism, the network can be influenced by users having a large amount of coins.

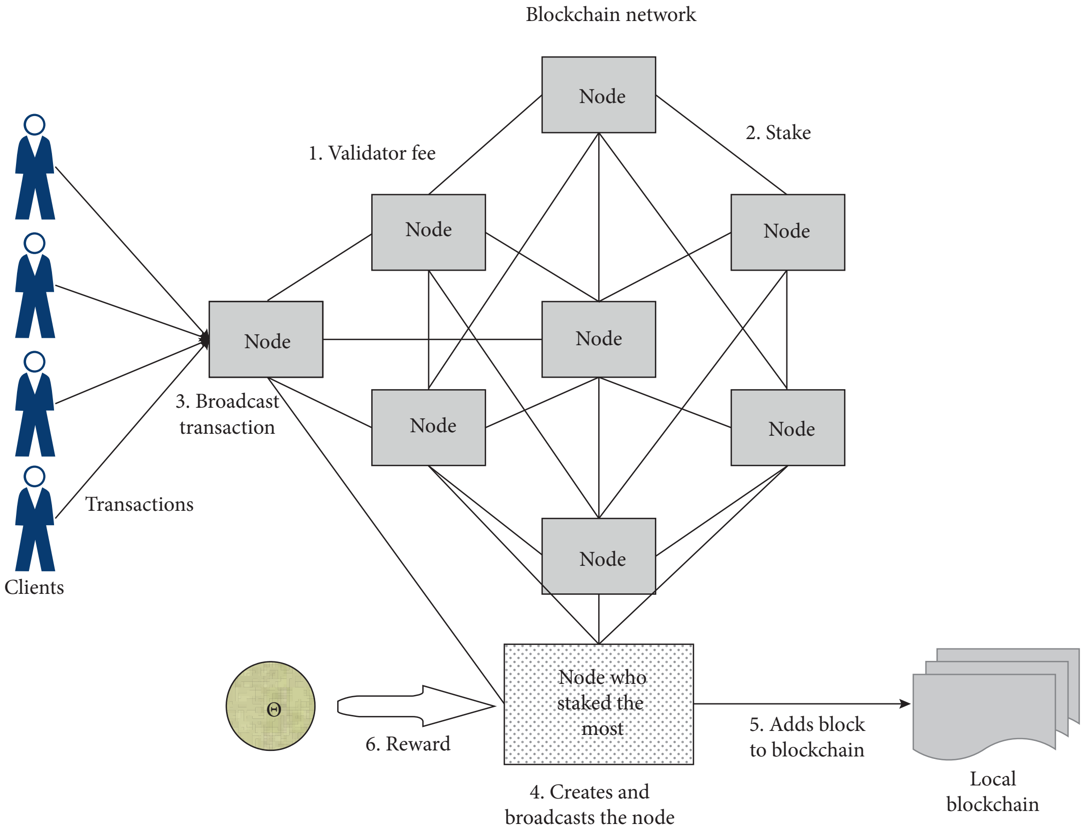

Figure 4: Steps depicting working of PoS.

(ii) In terms of long-term sustainability, it is not proven yet, as none of the largest three cryptocurrencies currently use PoS.

(iii) In the PoS mechanism, a cold wallet cannot be chosen as the next block creators, as it requires the users to sync their wallets to prove ownership.

4.3. Security

4.3.1. Secure against 51% Attack. PoS-based models are less prone to 51% attacks (as mentioned in 1), as to carry out this attack, 51% of the cryptocurrency needs to be obtained by the attacker. To avoid this attack, it is made disadvantageous for a majority stakeholder to attack the network. Although it is costly as well as difficult for a miner to accumulate 51% of a digital coin, it will not be advantageous for a miner with a 51% stake in the coin (a majority share) to attack that network, as a decrease in the value of the cryptocurrency is equivalent to the decrease in the value of his holdings. As a result, a miner with a majority share would not be incentivized to attack the network but will be motivated to maintain a secure network [27].

This can further be seen by taking the example of bitcoin PoS:

The bitcoin PoS is resistant to a 51% attack, as the bitcoin codebase is transitioned to the PoS consensus mechanism, which was designed specially to overcome the centralization problem. BTP also uses a variation in the Proof of Stake model, i.e., mutualized Proof of Stake consensus, that is not possible to be breached by an attacker, as it requires a lot of resources. At least 50% of the network's token supply is to be taken under control by the attacker if he or she wants to aim at bitcoin PoS, whereas in bitcoin, 51% of the network hash rate needs to be controlled by the attacker. This difference creates a recognizable change. By creating common interests for the heads of major mining cartels, hash rate can be secured. However, as tokens are distributed across a wider cast of actors that have varied network values, interests, and aims, they cannot be secured by using the same technique. It will be incalculably difficult to persuade the token holders for contributing or selling; therefore, it does not lie in the scope of threats to bitcoin PoS. In the PoS ecosystem, staking pools that lessen the requirement of technical knowledge by stackers and are beneficial for delegating stakes are accused as centralization sources. However, as staking pools are purely delegates of stake and they do not require to store the tokens that are being staked physically, it does not own the tokens in a saleable layout. Therefore, it further secures bitcoin PoS as well as PoS networks against 51% attacks.

Apart from solving PoW's 51% attacks security issue, PoS is at risk of the following attacks:

(i) Nothing-at stake attacks: the nothing-at stake attack means generating conflicting blocks on all possible forks to maximize the profits by putting nothing-at stake. For instance, by taking the reference of Figure 5, it can be understood as follows: in an ideal scenario when a fork takes place, the participants have to choose one among the two chains. But in the PoS mechanism, the participant is incentivized to follow both chains as his optimal strategy, as by picking only one chain at a time, he/she risks losing the transaction fees on the orphaned chain. However, by following both chains, the participant gets his reward either way, as one chain among them will be picked as a winner [23, 28].

Solution: as generating a signature consumes almost the same resources as generating a PoS block, in scenarios where conflicting blocks exist at the same height, a dedicated digital signature scheme can be used for enabling nodes to reveal the identity of the block leader. Alternatively, the rule of "three strikes" is suggested to be used for blacklisting the stakeholder who despite being allowed to create blocks fails three times consecutively to do so. Also, an auxiliary output needs to be signed by an elected mining leader that can be used as proof that it has provided the deposit, i.e., some extra amount of tokens. In this scenario, if the node comes out to be malicious and more than one block is broadcasted by it, then that auxiliary output can be used as evidence by any of the succeeding block creation leaders, to seize the deposit provided by the attacker. For disincentivizing a block forking, such a scheme is specially designed by the round leader.

(ii) Grinding attacks: a basic problem for PoS-based blockchain protocol that is faced because of its design is the simulation of the leader election process [19]. Although it is important to introduce entropy among shareholders, for achieving an unbiased randomized election in the group of stakeholders, mechanisms that can be used for introducing entropy could also be susceptible to be manipulated by the adversary. For example, by trying different stakeholder participant sequences to find a protocol continuation favoring the adversarial stakeholder, the adversarial stakeholder controlling a group of stakeholders can try simulating the protocol execution. This results in a grinding vulnerability (grinding attacks), where computational resources may be used for biasing the leader election by adversarial parties [23, 28].

Solution: including entropy based on the chain into target hash calculation is a solution to avoid these attacks.

(iii) Long-range attacks: long-range attacks target the voting mechanism of the PoS protocols. For these forms of voting-based protocols, the stakes of the committee members could also be sold by them right away at the start of the epoch for which they are selected. Still having voting rights, they are unaffected by the mechanism of incentives. As a result, without suffering from penality, they may behave in a malicious manner [28].

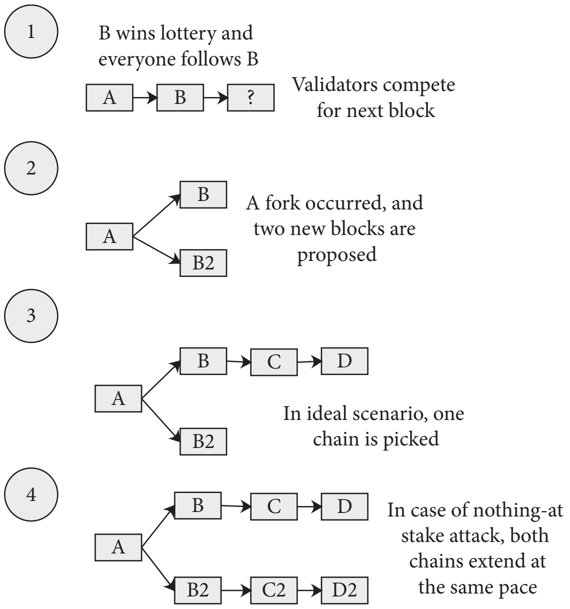
1
B wins lottery and everyone follows B
A
B
?
Validators compete for next block
2
A
B
B2
A fork occurred, and two new blocks are proposed
3
A
B
C
D
B2
In ideal scenario, one chain is picked
4
A
B
C
D
B2
C2
D2
In case of nothing-at stake attack, both chains extend at the same pace

Figure 5: Nothing-at stake attack on Proof of Stake models.

Solution: locking up of the committee member's stake for a predefined time at the end of the epoch is a way of managing such attacks adopted by some protocols. As a result of employing a committee for voting for every block, a block is finalized once it is appended to the chain. Thus, all the mentioned attacks are mitigated and it is impossible to alter the history of transactions.

5. Delegated Proof of Stake

The DPoS (Delegated Proof of Stake) agreement calculation is a variant of the PoS consensus algorithm that gives higher levels with respect to scalability and productivity at the price of reducing the number of validators to a limited count on the network. It was specifically designed with the idea of providing an answer to the phenomenon (known as scalability trilemma) [29] that indicates that a blockchain in which each transaction is validated by every node and can only possess two of the following three properties:

(i) Decentralized block production

(ii) Security

(iii) Scalability

The DPoS consensus mechanism executes a structure that has a fixed order of block production and thus predetermined block producers, also referred to as *witnesses*, as the sole actors of the given system that can produce new blocks and are liable for validating the transactions. By this implementation, much of the computational strain of the network is removed, as nodes are not required for verifying the work of the miner, which stands to increase the efficiency of the blockchain. For a block producer candidate to become a witness, it has to be voted in by that blockchain network's users, i.e., the *token holders*. Every token holder can vote for the desired block producer, and the weightage of their votes is based on their stake, i.e., the number of tokens owned by them; thus, it implies that the influencing power of the voters is directly related to their stake. Alternatively, token holders can pass on their stake to another voter, for casting a vote on their behalf in the election of block producer. Thus, for a system offering *liquid democracy* [30], the DPoS mechanism is incredibly optimizing for it. Each of the systems using the DPoS consensus mechanism has set its limit for the number of witnesses responsible for proposing blocks. Once all witnesses are elected, they can produce blocks after verification of all transactions that are cumulated for the last block time (generally around a couple of seconds), in the order in which they were selected. They get a reward if all transactions in the block are verified and signed by them. The reward is generally shared with users who voted for the witness. But if a witness is not able to do so in an allowed time, then the block is missed, and all transactions in the block are left unverified, and the reward is not given to the witness. Usually, these transactions are collected in *stolen*, an additional block, by the following witness.

In DPoS systems, voters can vote out witnesses and vote for other candidates that they feel are more suitable for the task; i.e., they have continuous elections. This mechanism of voting incentivizes the witnesses to act for the welfare of the network's stakeholders.

Voters can also vote for "delegates;" they are parties that are responsible for supervising the performance and governance of blockchain. These parties are liable for actually performing updates of the codes.

In an ideal scenario, the working of the DPoS consensus mechanism as discussed above can be visualized as Figure 6.

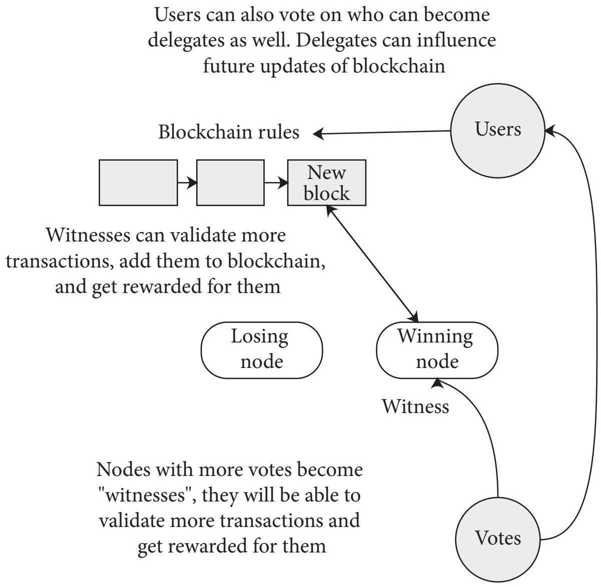
Users can also vote on who can become delegates as well. Delegates can influence future updates of blockchain
Blockchain rules
Users
New block
Witnesses can validate more transactions, add them to blockchain, and get rewarded for them
Losing node
Winning node
Witness
Nodes with more votes become "witnesses", they will be able to validate more transactions and get rewarded for them
Votes

Figure 6: Working of DPoS.

5.1. Application

5.1.1. BitShares. BitShares blockchain, an open-source financial platform, uses a Delegated Proof of Stake consensus mechanism, which means that voting on consensus issues can be done democratically by stakeholders. Delegated Proof of Stake provides it with a greater speed, efficiency, and flexibility and allows for trading of arbitrary pairs without counterparty risk facilitating one of the available features. The BitShares network is one of the fastest in the industry as DPoS allows for ten transactions per second. BitShares, a decentralized network run by users around the world, keeps their databases synchronized as per the rules defined by the software [31]. This allows the BitShares network to run as long as there are at least two participants in the network communicating with each other over the Internet. Letting its shareholders decide on its future direction and products, BitShares represents the first decentralized autonomous company. Being a decentralized exchange on the blockchain, it cannot be easily compromised because the data in it are not stored at one central location only and it allows for trading of arbitrary pairs without counterparty risk. BitShares aims to improve certain aspects concerned with the financial system used currently like speed and privacy.

5.1.2. Steem and Steemit. Steem blockchain, an application based on the DPoS mechanism, is a social blockchain platform. It is the publicly accessible distributed database, which records all of the data and transactions, processes all of the events that take place, and distributes the rewards across the network. It is also a name for the system's value token. Based on social media, the Steem crypto provides a platform for decentralized application hosting and data storage.

Steemit is a front end web interface that views the blockchain data and interacts with the blockchain. It is the first completely decentralized social media or publishing platform. To protect the blockchain, Steemit uses the DPoS mechanism [32]. It is totally driven by community members and the contents are censorship-free. Steemit is an interface where a community coordinates without explicit coordination. Steemit is a transparent platform, as you can view all details in the blockchain. It also has a private wallet.

5.1.3. EOSIO. EOSIO, an application based on the DPoS mechanism, is a blockchain infrastructure that favors decentralization and copies the characteristics of computer hardware. It is a smart-contract platform that enables the development of industrial-scale decentralized applications [33]. The system is powered by the EOS coin, whose holders can access the distributed computing resources. This platform aims at simplifying the development and production of the decentralized application (DApps) of all types and supporting industrial and enterprise-scale DApps. With the help of asynchronous communication and parallel execution, it supports commercial and industrial-scale DApps, to continue scaling as new machines enter the network. This platform is envisaged as the solution to common blockchain development issues such as scalability, flexibility, speed, and transaction price. The EOS coin powers the EOSIO system, as the distributed computing resources can be accessed by the holders [34].

5.2. Performance. DPoS consensus has several advantages and downfalls surrounding it which have been discussed in this section.

5.2.1. Advantages.

(i) In DPoS networks, everyone earns despite being concerned with the concept of rich or poor, i.e., the amount/number of coins held by the nodes, thus making DPoS more decentralized and offering a better reward distribution.

(ii) A user can immediately vote to oust the offending delegate if he or she finds any malicious activity, thus keeping real-time voting secured.

(iii) As DPoS blockchains do not require high computational power, they are more scalable and in a given period can process more transactions. It also makes it environmentally friendly.

(iv) It offers digital democracy, as it gives the chance to decide the block producers to the one with more tokens.

5.2.2. Disadvantages.

(i) Giving the power of decision-making to the small group of holders moves the mechanism away from decentralization. It also becomes easier for a 51% attack, as only a limited number of people are responsible for keeping the network alive.

(ii) There are increased chances in this mechanism, that cartels can be formed by the witnesses, and ultimately can rule the network.

5.3. Security. The major attack vectors that make the DPoS blockchain system less secure can be divided into the following attacks:

(i) Exploit low voter turnout: in DPoS blockchain, a voting system with very little participation in voting, this is the most obvious attack. In such blockchain token voting systems, it is unlikely for the participant with a small stake to influence the direction of the platform in any form with the help of their vote. Therefore, in such a scenario, spending time on researching how to vote may not be worth it, as it will be more costly as compared to the value brought by their voting. DPoS attempts to resolve this issue by allowing proxy voting, i.e., voting power can be lent to another user who is considered to have more information, by a user with a small stake. In this scenario, the value gained is likely to be more than the value of the effort put in deciding to whom to transfer the voting power. Still, this results in an overall low voter turnout thus giving the voting power to wallet providers, whales, and exchanges. The attacks on the voting system are not good for the network, as a result of these attacks, there is a decrease in each token's price. The ones with a significant stake in the network need to be incentivized to vote so that the value of their tokens can be protected. Although a very significant stake needs to be purchased by the attacker, in order to take over governance, it is possible that token holder's percentage whose vote may be small might have the largest stake. As it is possible to incentivize large stakeholders to improve the network, it is not necessarily a bad thing [30].

(ii) Bribing attack: bribing attack, i.e., block producers paying for votes is possible in any implementation of DPoS. These attacks success and sustainability depend on other elements of the protocol as well; as a result, other features of DPoS make it much less likely to occur. In such systems, profit sharing becomes an easy option, where simply running a cloud instance to do the validation work is the only requirement for a block producer. As a result, we see block producers paying for votes as a bad thing, as it encourages voting based on who pays the highest returns rather than what is best for the network. This does not align the long-term incentives of token holders and of the network [30].

(iii) Block producers collude: in DPoS, the threat of colluding block producers is presented dangerously, the small number of validators being the main reason. Theoretically, organizing collusion among them should be easy [30]. In DPoS systems, the colluding block producers (more than 2/3 of all block producers) can launch three major attacks:

Censorship
Changing system parameters
Double spends

(iv) Distributed denial-of-service attack: in DPoS models, the attackers find it easy to spot the next block producer at a given time, as during each round the order of block production is fixed, and it becomes easy for them to launch DDoS (distributed denial-of-service) attacks on the producers. And in practice, it is difficult to pull off such attacks. Although it is likely for a single block producer to be targeted by an attacker and the network to suffer temporary delays because of this, it is unlikely for an attacker to simultaneously target multiple block producers. Furthermore, block producers can resist these attacks by using backup servers in other locations and various other means. Finally, a single or a few block producers can be voted out and replaced by their backup producers in a given round, if due to these attacks, they are consistently failing to produce blocks [30].

6. Proof of Elapsed Time

Proof of Elapsed Time (PoET), a lottery style blockchain network consensus mechanism, has a similar workflow like that of the PoW mechanism, but it requires far less computational resources, i.e., it eradicates the need for mining-intensive processes, thus preventing high energy consumption and resource utilization. In PoET, a separate random timer that operates independently at every node of the distributed ledger determines whether or not that node creates the new block for the blockchain and gets the reward. Thus, it mainly focuses on efficiency and ensures that the probability of every node to be the next block generator is equally likely.

The PoET concept was invented early in 2016 by Intel to solve the computing problem of "Random Leader Election" and is essentially used to achieve consensus in the *permissioned blockchain* 1.2 networks. As stated above, in PoET, each participating node within the network is given a random timer object; i.e., each node is required to wait for a randomly chosen amount of time. Each of the nodes within the blockchain network goes to sleep for the duration specified by their timer. The node which completes the designated waiting time first; i.e., the one with the shortest wait time becomes the next block generator and commits a new block to the blockchain by broadcasting the new block to the entire peer network. The process of choosing the block generator is depicted in Figure 7. For the discovery of the subsequent blocks, the method is repeated [35].

Before moving on to the working of the PoET mechanism, we need to go through a sophisticated technology that plays a crucial role in this protocol, namely, *Software Guard Extension* (SGX). It is effectively a set of security-related instruction code executed on CPU which is employed by applications to isolate certain trusted regions of code and data. These isolated regions are called *enclaves*. It primarily protects sensitive data and code from any outer interference or inspection, by providing a secure enclave for developers. Primarily, SGX functions to provide a trusted execution environment (TEE) which allows specific trusted code and data to execute independently of the application and system on which it runs [36]. Code that executes in a trusted environment created using SGX has the capability to generate a signed attestation from within the platform or application which is rooted in the CPU and provides authentication that the code has been correctly booted up in a trusted environment. SGX acts as a mechanism by which participant nodes hitch the network and confirm whether they are executing the trusted code needed for the PoET consensus implementation.

The working of PoET consensus protocol can be broken down into the following 2 stages:

(1) Verification and network joining: this is the initial phase of consensus and also the phase in which SGX plays a crucial role. In this phase, a node willing to join a permissioned network downloads the trusted PoET code. This downloaded code is then run on SGX which generates an attestation and a new private/public key pair. Using this key, the node signs the attestation and forwards this signed attestation (which also contains the public key of the trusted code which is executed in the SGX environment) along with a join request message to the rest of the network. The nodes which are already a part of the network verify this attestation. Based on this verification, the network nodes either accept or reject the signed attestation. If the attestation is accepted, then the node is welcomed in the blockchain network and here onwards is permitted to take part within the randomized next block generator selection process of the PoET mechanism.

(2) Participation in the actual mining lottery elapsed time round: this is the eventual phase of the mechanism where the actual mining process occurs. The participants of the network receive a signed timer object from the trusted code which they had setup using SGX. This code ensures that the timer object is totally randomized in each round of consensus. The concept behind this randomization is to alleviate any possible malicious network node from be-fooling the network by attempting to continuously acquire the shortest timer object to be able to generate more blocks and get more rewards. Each participant node then waits for its randomized designated timer object to expire, and the first node to have an expired timer gets to be the winner. The winner node obtains a certificate that is signed with the private key of the trustworthy code and broadcasts it to the rest of its peer nodes in the network, indicating that it is the randomly selected block leader for that round. It needs to do so in order to prove that it had the shortest wait time and has

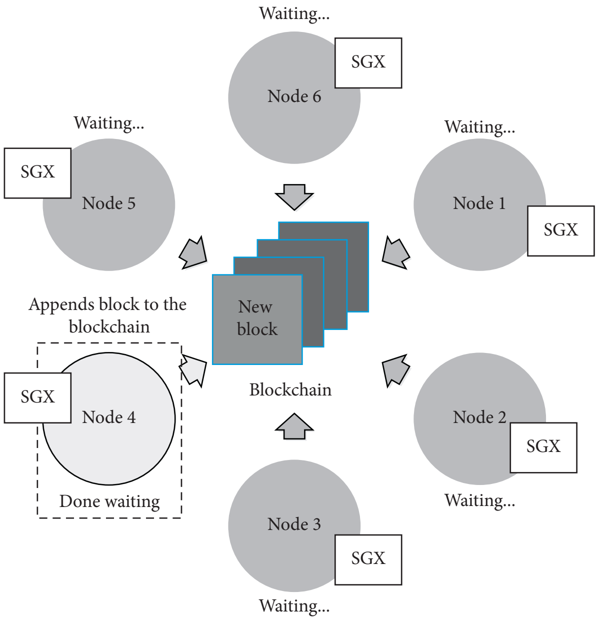

Figure 7: Committing block to blockchain by block leader in a round.

waited for the designated amount of time before starting to mine the next block. This node then creates and commits the next block to the blockchain, broadcasting required information to the peer network [25].

The process of signed certificate propagation is actually carried out by all the participants and not just the winner node to ensure that each of the participants obtained a randomized timer object from the trusted code and each of them waited for the time duration specified by the timer object associated with them. Moreover, the integration of SGX ensures that the trust code remains unaltered by any external entity.

6.1. Applications

6.1.1. Hyperledger Sawtooth. Hyperledger Sawtooth developed by Intel Corp. is an enterprise-grade blockchain platform that has been designed to develop distributed ledger applications and networks. It isolates the core ledger system from the application-specific environment. This simplifies the process of application development and keeps the core system safe and secure from outside interference. The core ledger system allows multiple applications to exist together on the same blockchain, selects the required permission mechanism for the chain, selects transaction rules, and defines the consensus protocols which are used to finalize the working of the digital ledger in a way that best supports the demands of the business enterprise which uses it (as shown in Figure 8). It uses the "Advanced Transaction Execution Engine" to provide multiprocessing of the transactions for fast creation and validation of blocks. This parallel transaction execution boosts the operating performance of the sawtooth network, which features superiority over the serial execution mechanism that is always a bottleneck while handling a large volume of transactions in many well-known cryptocurrency networks. Thus, the application developers can develop applications and host, run, and operate it without having to understand the underlying design of the core system [35, 37].

A sawtooth blockchain stands out from others owing to its following features:

1. Highly modular, thus allowing developers to develop applications and host, run, and operate it on system periphery without hindering the core blockchain system.
2. Scalable
3. Supports permissionless and permissioned infrastructure

Hyperledger Sawtooth blockchain uses the PoET consensus algorithm to implement a pacesetter election lottery system to select the next block generation unit. Sawtooth comprises of two versions of PoET [38] consensus:

1. PoET-SGX: PoET-SGX depends upon a trusted execution environment (TEE), like intel software guard extensions (SGX), to implement a leader-selection lottery system. As PoET-SGX is Byzantine fault-tolerant, it is usually called "PoET/BFT."
2. PoET simulator: PoET simulator provides PoET styled consensus mechanism on any type of hardware as well as in a virtualized cloud environment. As the PoET simulator is crash fault-tolerant, not Byzantine fault-tolerant, it is additionally called "PoET/CFT."

6.2. Performance. The advantages and shortcomings surrounding the Proof of Elapsed Time consensus protocol are discussed below.

6.2.1. Advantages

(i) PoET uses a randomized timer system to pick up the block leader which will produce the next block, thus eliminating the need for any resource-intensive mining process. This makes PoET more efficient.

(ii) PoET does not have scalability issues and works brilliantly even with a large number of participants.

(iii) Since the block leader is chosen by a randomized process, and thus, it rules out the possibility of the same node becoming the leader again and again, unlike in PoW where a group of nodes having adequate resources to mine a block are comparatively faster and dominate the network.

6.2.2. Disadvantages.

(i) One of the downsides of this protocol is its dependency on the trusted execution environment enabled hardware system. Although TEE-based hardware protects the system against malicious attacks by maintaining a monotonic counter which ensures that only one copy of the blockchain is running on one processor, it makes the utilization of this protocol very limited [25].

(ii) SGX, an important part of the protocol, runs against one of the three basic pillars 1.2 of the blockchain model, i.e., eradicating the need of any third party and trusting them. As SGX is a product of Intel, the consensus model relies on Intel, a third-party organization.

6.3. Security. This section deals with the security analysis and possible security breaches in a system using Proof of Elapsed Time consensus. Since PoET is a relatively new protocol, it is comparatively less tested than other consensus mechanisms; thus, not much is known about its reliability, and it is hard for us to predict the extent up to which it can tolerate failures of trusted computing components. However, the PoET protocol is open to *Sybil attack*, where the attacker gains influence over the network by creating a large number of fake identities and uses them to exploit the network [39]. Consider a system consisting of n nodes where each of these nodes keeps on generating blocks after waiting

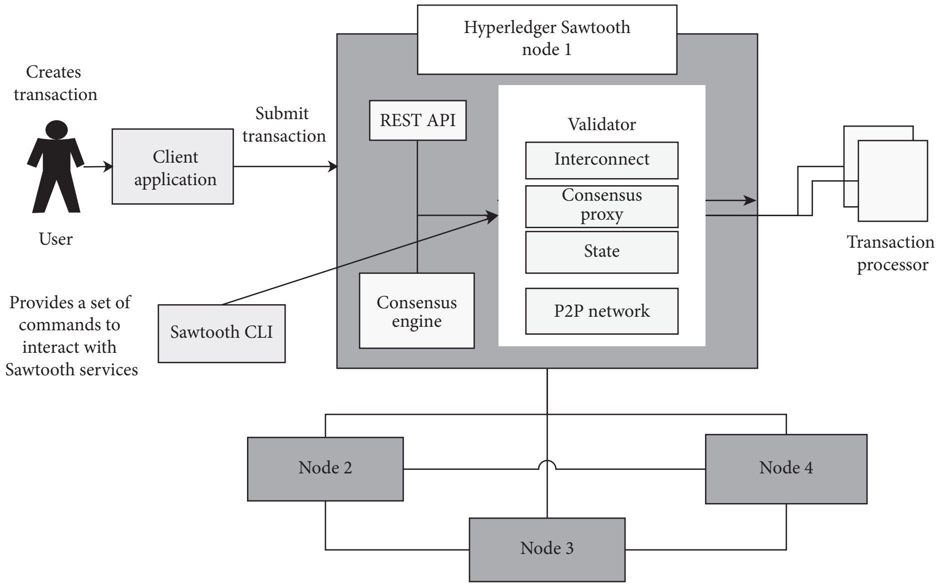

Figure 8: Working of Hyperledger Sawtooth.

for a certain amount of time, based on a probability distribution implemented by the trusted code, and adding it to the blockchain. In a given fixed time interval, it is probable that the fastest node in the network will generate a greater number of blocks than the average ones. A statistical test is performed in order to discover whether a node has created an unusually large number of blocks compared to the other nodes within a certain time period, and if any such block is found, it is treated as a malicious and is removed from the network. To mount a Sybilian attack, an attacker has to compromise several nodes, and each one of the compromised nodes should be able to create new blocks irrespective of the waiting time period as long as these nodes pass the statistical test. In this way, if the attacker is able to create new blocks using the compromised nodes and these newly generated blocks are able to pass the statistical test provided that the total number of blocks generated by all the compromised nodes exceed the total number of blocks which are generated by the remaining honest nodes in the network by a constant $H > 0$, the blockchain system is compromised. To exceed the number of honest blocks, the attacker keeps on adding new fraudulent blocks to his attack chain, while the rest of the network is busy adding blocks to the main chain. This ensures that the attack chain will eventually take over the main chain even if initially it is shorter than the main chain by H blocks. After this, the attacker would be able to make the compromised node to generate as many blocks as it wants by making the node to pretend as the fastest honest node in the network. The detailed explanation of this attack is beyond the scope of this paper, to know more refer [40].

7. Practical Byzantine Fault Tolerance

The PBFT (Practical Byzantine Fault Tolerance) consensus mechanism was proposed by Miguel Castro and Barbara Liskov in 1999 in a research paper titled "*Practical Byzantine Fault Tolerance*" [41]. Practical Byzantine Fault Tolerance is primarily modeled to provide Byzantine state machine replication which is tolerant to the malicious nodes (Byzantine faults) of the system which fail or propagate incorrect information to its peer nodes. The main objective of such a consensus mechanism is to prevent the system from catastrophic system failures by reducing the influence of the compromised nodes on the network functioning and to reach the right consensus with the help of honest nodes. This protocol is outlined such that it works well for non-synchronized systems and provides high performance with a reduced runtime although it has a slight delay in data transfer. The participant nodes in a PBFT based network to act as follows:

(i) In a PBFT model, all the participating network nodes are arranged in an order where one node is the leader node (primary) and the rest of the network nodes are referred to as the backup ones.

(ii) All of the nodes within the framework communicate with one another and come to an agreement amongst honest nodes to achieve a system condition through the majority.

(iii) Nodes communicate to one another vigorously and not just need to verify that messages originated from a specific network node and additionally need to check that the message did not tamper during transmission.

For the PBFT model to work, the supposition that is followed is that number of malevolent nodes inside the system cannot all the while rise to or surpass 1/3 of the general nodes inside the network during a given pass of helplessness or vulnerability.

As the quantity of nodes increases in the network, it becomes all the more far-fetched for a number drawing nearer 1/3 of on whole nodes to be malignant. For whatever length of time that at most $n - 1/(1/3)$ nodes, where n speaks of total nodes, are vindictive or flawed simultaneously, the calculation gives both security and liveness. As a result, the answers received by customers from their solicitations are gratitude to linearizability. As all officers are equivalent, on account of the nearness of a pacesetter node, this model follows to a greater degree an "Administrator and Lieutenant" group than an unadulterated Byzantine general's problem.

Each round of PBFT consensus (called views) is divided into 4 phases [43]:

(1) A solicitation is sent by the client node to the pioneer (leader) node to summon an assistance activity.

(2) The request to the backup nodes is multicasted by the pioneer node.

(3) The remaining requests are afterward executed by the nodes, and an answer is sent to the customer.

(4) Then, the client anticipates $f + 1$ (f speaks to the most extreme number of nodes that might be broken) answers from various nodes (as shown in Figure 9) with a proportional result.

The prerequisites for the nodes are that they are adamant and begin with a synonymous phase. A definitive result is that every legit node goes to a concession to the request for the record and they either acknowledge it or reject it. The pioneer (leader) node is modified in a cooperative sort of round robin form position during each view and may even get supplanted with a protocol called view change if a specific measure of time has gone without the pioneer node, multithrowing the request. A supermajority of legitimate nodes may likewise choose whether a pacesetter is broken and evacuate them with the ensuing leader in line as the substitution.

7.1. Application

7.1.1. Hyperledger Fabric. Hyperledger, a blockchain consensus which is a part of *"The Linux Foundation,"* comprises a modular architecture that goes about as a foundation for making blockchain-based products, plans, and applications [25]. Fabric is one of its mainstream implementations [44]. It utilizes attachment and plays parts that are meant to be utilized inside private ventures. Linux Foundation hosts these projects employing a permitted version of PBFT for the platform. The permissioned group does not require decentralization like the public blockchains as they use small consensus groups. Also, Practical Byzantine Fault Tolerance offers high throughput transactions more effectively. The permissioned blockchain usually has a component of trust between parties, eliminating the necessity for the nontrusted environment and allowing the network to obtain sensible Byzantine Fault Tolerance (PBFT) advantages with no limitations. Inside private mechanical systems, the evident character of a member is an essential prerequisite. Hyperledger fabric bolsters enrollments based on permission, i.e., all network participants, must have verified existence [45]. Numerous business divisions, similar to medicinal services and funds, are limited by information assurance guidelines that order keeping up information about the shifted members and their individual access to fluctuated information focus.

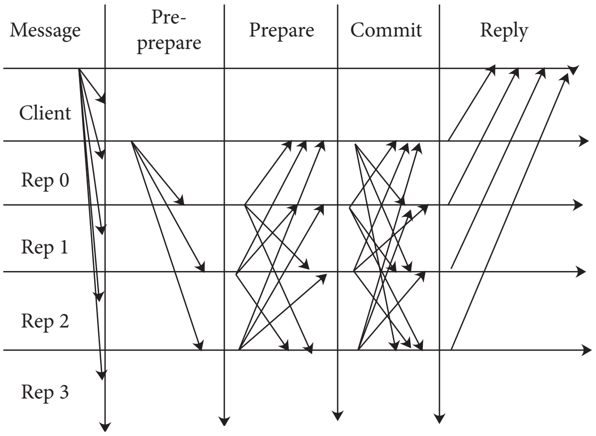

Figure 9: PBFT workflow diagram showing the 4 stages [42].

7.1.2. Zilliqa. Zilliqa applies the optimized version of classical PBFT combined with PoW consensus [46], in which about each of 100 blocks need to perform network sharding. Here, we divide the network into chunks called the shard. Every shard can process transactions in equal, prompting high throughput for the network. This network utilizes multimarks to downsize the correspondence overhead of old style PBFT. The application prefers a consensus to be less intensive in terms of computation so that it is energy efficient. In the Zilliqa network, the validators are chosen randomly in an egalitarian manner, as long as you have got the capital to amass the required hardware to run PoW and obtain a ticket of entry, you can participate. Instead of choosing just the standard 21 validators and let it rest, Zilliqa chooses to elect 600 validators per shard [47], as their research has shown it is a perfect number to stop the corruption of shards during this epoch before subsequent reshuffling of the group of validators. It is believed that the probability of getting 1/3 +1 of the shard, while using a 4-shard network, with 51% of the hash power is almost null [48].

7.2. Performance. PBFT consensus has several advantages and downfalls [42, 49] surrounding it which have been discussed in this section.

7.2.1. Advantages.

(i) Transaction finality: PBFT gives finality to the consensus decision; i.e., the proposed block is final and the agreement is exact if PBFT is used to reach consensus on the next block. As a result, the transactions can be approved and finalized without the need for confirmations like in the PoW models. This can be further explained as follows: suppose the nodes in a PBFT system agree upon the proposed block, then that block is finalized. It is possible only because after communicating with each other at a specific time, all legitimate nodes agree on the phase of the network.

(ii) Higher energy efficiency: since PBFT may or may not require energy-intensive computations to achieve network consensus, it diminishes the vitality impression for miners. Zilliqa utilizes PoW not for consensus but to forestall Sybil attacks and setup node characters. This implies a few rounds of PBFT convention can be pursued after identities have been setup, to concede to various blocks in succession. Therefore, PoW should be possible, just to state, after every 100 blocks.

(iii) Low reward variance: PBFT consensus protocol requires collective decisions through common agreement on records by marking messages, while in PoW, only the leader has a chance to propose the next block. Hence, the layer can be designed to incentivize every active node with the use of PBFT. Thus, every node is rewarded in the PBFT system and it lowers reward variance for miners.

7.2.2. Disadvantages.

(i) Each node should communicate with other nodes, to keep the network secure, as a result of the following: as the number of node scales increases, a huge communication cost is indulged. So, the PBFT mechanism works fine with small consensus group sizes.

(ii) PBFT models are vulnerable to Sybil attacks, in which a large number of nodes can be manipulated by a single party in the network, thus compromising the security [39]. This threat can be reduced by increasing the size of the network, but the PBFT mechanism does not support large networks because of the above reason. So, by using it in combination with another consensus mechanism, it can be optimized.

7.3. Security. The PBFT is secure against denial-of-service attacks, as it does not rely on any synchronous assumption to provide safety, and avoiding synchronous assumptions is the most important defense against these attacks. As a result, these attacks cannot cause incorrect replies to be returned by a replicated service. But by exhausting resources at the replicas, they can prevent the returning of replies by the service. The main idea is to prevent the monopolizing of resources by replicas or clients, by carefully managing the resources. This is achieved using the defenses, like bounding the rate of execution of expensive operations, using inexpensive message authentication, scheduling client requests fairly, and bounding the amount of memory used. Replicas immediately reject the messages that are authenticated by an unknown client and only accept those authenticated by a known client. This can be achieved because of the use of MACs by message types, which can be computed inexpensively. The recovery requests and new-key messages signed using public-key cryptography are the only exceptions to this. Since these messages are only sent periodically by correct replicas or clients, they can be easily discarded by replicas without even checking their signatures, but the condition is that the last message from the same principle was processed before, in timeless than a threshold time. This helps in achieving two important things that are expensive to process, which includes bounding the rate of signature verification and the rate of processing of authentic messages from faulty principals. The algorithm also uses a bounded amount of memory, as it retains information only related to sequence numbers between the low and high watermarks in the log. It also bounds the amount of information per sequence number. In addition to all these, the fraction of memory used on behalf of any single replica or client is also bounded. All these defenses ensure that the algorithm always has enough memory space to be provided for the service after an attack ends. A FIFO queue is maintained by the algorithm for the requests that are waiting to be processed, ensuring the fair scheduling of client requests; only the request with the highest timestamp from each client is retained in the queue. If the requests are not scheduled fairly by the current primary, a view change is triggered by the backups. By caching the last reply sent to each client and the timestamp, è, of the corresponding request, the algorithm defends against attacks that replay authentic requests. Requests that have timestamp less than è are immediately discarded, and the cached reply is used by replicas to handle requests with timestamp è efficiently [49].

But as discussed in Section 7.2.2, it is vulnerable to Sybil attacks.

8. Proof of Authority

The Proof of Authority (PoA), a consensus algorithm, gives the power of updating its registry, which is distributed in nature, and validating the interactions/transactions that are happening in the network and in the hands of blockchain actors that are small and designated in number. Its working can be explained as follows: new blocks of transactions, that are to be included in the blockchain in the future, are generated by the validating machines which are one or more in number. In this mechanism, it is possible to accept a block without doing any verification process, or by the process that is based on block generators' undivided vote, or based on majority, but overall, it varies according to the chosen blockchain's configuration [50]. PoA uses the identities' value for its advantage, and this fact implies that validators of blocks are putting their reputation on a stake instead of coins, as was the case in PoS. Therefore, for the blockchains that use PoA as their consensus mechanism, the validating nodes, which are the nodes that are selected arbitrarily as trusted entities, are responsible for securing the blockchain. Models based on this mechanism are highly scalable, the reason being that they do not depend on a large number of block validators. Moderators, the participants selected prior, are responsible for verifying transactions and blocks. PoA model enables the organization to maintain privacy while being benefited by blockchain technology. Even though the conditions may fluctuate from framework to framework, the PoA mechanism is typically dependent upon the following:

(i) Valid and trustworthy identities: validators need to affirm their genuine identities. It is required for them to be enrolled in the open public accountant database with a similar identity they have on the stage. In basic terms, various programs (*"smart contracts"*) are stored on the blockchain, and for each validator and "piece of identity," the validator must start a verification through these projects, which produce a code that the validator must submit as confirmation of their identity. This code can only be obtained by the owner of what is being confirmed. Regardless of whether one takes a permit and some way or another turns into a validator claiming to be another person, the way that marking is open will uncover a malicious actor. A solitary actor with a hacked authority cannot overpower the network and return all exchanges. Consequently, requiring validators to have an open legal official permit makes someone difficult to cover, while a concerned party can without much of a stretch cross-confirm identity in freely accessible open databases.

(ii) Difficulty in becoming a validator: not just money but the reputation is also at stake when someone decides to become a validator. This extremely selective process of appointing the validators is what lowers down the risk of making a wrong decision concerning the validators.

(iii) A standard for validator approval: the method for selecting the validators must be equal to all candidates. In return, for revealing their true identities and proving their legitimate existence with the help of government-issued documents, validators are rewarded with power. The number of *"authorities"* who are responsible for validating the transactions and blocks on the blockchain should be minimum (around 25 entities). Following this, efficiency and security can be achieved by the group.

With PoA, individuals gain the proper recognition to become validators, so there is an incentive to retain the position that they need to gain. By attaching a reputation to identity, validators are incentivized to uphold the transaction process, as they do not wish to have an association with the negative reputation. PoA algorithms believe a group of authorities, trusted nodes that are N in number. A unique id is used as an identification of each authority, and it is assumed that the majority of authorities are honest, to be specific N/2 + 1. The role of the authorities is to run a consensus, with the aim of ordering the transactions that are issued by clients. The Proof of Authority consensus algorithm depends on a mining rotation schema [51]. In a PoA blockchain, the full power of making decisions regarding new blocks is given in the hands of validation nodes. This suggests that they need to prevent the transactions, which in the future can generate conflicts of interest and can even compromise the safety of the network. The steadiness of the system is ensured by permanent control and monitoring of the validity of the operations, as conflicts of interests may be generated if the nodes are controlled by actors, who do not have confidence in one another. Thus, each block validator is encouraged to fulfill its role as an "honest" node, as each validator's actions are closely monitored by the opposite validators.

PoA combines a particular level of decentralization efficiently during a new protocol that needs less computational power. Its ability to succeed in consensus while maintaining some kind of decentralization of the network makes it a preferred consensus algorithm in the blockchain. It might be a feasible solution for companies looking to implement in-house blockchain solutions to extend productivity.

8.1. Applications. Two of the most significant PoA executions, which are utilized by Ethereum customers for permissioned arranged organizations, are discussed as follows:

(i) Aura: authority round, known as Aura, is an implementation of Ethereum, deploying the Proof of Authority consensus. Aura has parity implementation [52]. This rust-based implementation can deal with 50% of the faulty nodes with chain identifications possible up to a limited depth and hooked into the number of validators, after which finality is guaranteed. Here, the address of validators that are participating in every height is required. The process of gathering transactions connecting a header to the supply block is known as block sealing. During the sealing process, the blocks must be sealed on top of the most up to date known block inside the canonical chain. The block's header incorporates the step and the block hash's primary signature. Blocks are frequently confirmed by watching that, for a given step, the signature is held by the right primary. The conclusiveness of the chain is often reached within at most $2x\,(\#_{o}f_{v}$alidator) steps [51].

(ii) Clique: clique is another algorithm of Proof of Authority. Its implementation can be seen in Geth, i.e., the GoLang-based Ethereum client. The process begins with small periods that are recognized by a chain of blocks that are committed. When a replacement period starts, a transition block is disseminated. It mentions the "ids" of the validators and is utilized as a visual of the present blockchain where the new authorities would want to merge or synchronize. In contrast to Aura, clique employs a formula, that is, a mix of the number of authorities and block number, for calculating the present step and related leader [51]. Not only the present leader but also other authorities are allowed to add blocks in each level. The method of sealing a block is pretty straight forward. Here, there is no such thing as mining, and sealers do not spend time trying to brute-force the hashes, so very little computation is required. Besides the absence of mining, making a block in clique is analogous to the Ethash. Miner with the help of his private key signs the block after calculating the hash of the block when he has collected the transactions and updated the network. To limit the amount of processed transactions, clique allows creating one block per defined period of your time [53].

8.2. Performance. PoA consensus has several advantages and downfalls surrounding it which have been discussed in this section.

8.2.1. Advantages.

(i) Unlike the Proof of Work mechanism, commonly mentioned as "mining," there is no technical competition between validators here. This consensus mechanism requires almost no computing power and thus almost no electricity for its operation. The PoA algorithm does not have to solve puzzles, to make sure of the permanent connection between nodes. Since the PoA requires only a limited number of actors, the network can afford to update the blockchain timely by reducing the time between each block (block time) and process more transactions (block size) for processing fees on the brink of zero (transaction fees).

(ii) The time interval at which the generation of new blocks takes place can be predicted. For the other consensus like PoW and PoS, this point varies.

(iii) Blocks are generated according to the allotted interval by the nodes in the network that are authorized to do so. This increases the speed at which transactions are validated. Blocks are generated during a predictable sequence supporting the amount of validators; therefore, the blockchain registers a better transaction rate than PoW or PoS.

(iv) Forbearance to the faulty and malicious nodes, as long as 51% of nodes are not compromised. It implements a mechanism to restrain the nodes and means of invalidating block generation rights.

(v) The networks with PoA consensus are very ascendable, especially as compared to PoW blockchains, and are well suited to be a platform for the development and maintenance of DApps [53].

8.2.2. Disadvantages

(i) The perception of PoA is that it forgoes decentralization. It can be said that this form of consensus is simply an attempt for a centralized system to be more efficient, while this makes PoA a beautiful solution for giant corporations with logistical needs, and it brings some hesitation, especially within cryptocurrency scope. PoA networks definitely have high productivity in terms of throughput, but aspects of immutability inherit questions when it is very easy to attain censorship and blacklisting. PoA thus sacrifices the decentralization so as to realize high throughput and scalability [54].

(ii) The specifications of validators are accessible to anyone. The contention against this is frequently the group of setup players equipped for holding this position try to turn into a validator (as an openly known member). In any case, knowing the validator's identity might cause outsider control or interference, for instance, a competitor with the intention of disrupting the network and manipulating the validator to act dishonestly.

(iii) The threat of harming the reputation does not really shield a person from taking part in pernicious activities. The components of the gains which will be accumulated with a reputation wrecking occasion are regularly more important than the reputation in the network. This might cause third party involvement and thus potential chances in fraudulent or malicious activities.

8.3. Security

8.3.1. 51% Attack. The PoA consensus is intended to be significantly more insusceptible to assaults than the PoW consensus. This is on the grounds that the system cannot be undermined by a client who figures out how to get 51% of the computational force. In PoA consensus, the 51% attack requires an assailant to deal with 51% of system nodes [50]. This is regularly not quite the same as the 51% attack for the Proof of Work consensus types where an assailant must acquire 51% of system computational force. Acquiring control of the hubs during a blockchain network which has consented is far tougher than getting computational force. Although in a PoW consensus-type network, computation power (performance) can be increased by an attacker for the controlled network segment, controlled percentage is increased. For PoA consensus, there is no point in this because the blockchain network decisions are not affected by the computational power of the node. Dealing with 51% of authorities that are not associated legitimately is a lot harder than that. Nodes are previously verified, and on the off chance that one among them is inaccessible, the network can prohibit it from the approval procedure. An attacker sends an outsized number of exchanges and blocks to a focused on network node with an end goal to upset its activity and make it inaccessible.

8.3.2. Denial-of-Service Attack. The PoA mechanism makes it possible to defend against this attack:

(i) Since network nodes are verified, block creation rights are regularly allowed uniquely to nodes that will withstand DoS attacks.

(ii) In the event that a node is inaccessible for a specific period, it is frequently prohibited from the rundown of approving nodes.

9. A Comparative Study

In this section, we present the advantages, drawbacks, security, and applications of all the above discussed consensus algorithms. This is depicted in Table 1.

Table 1: Comparative study of the protocols.

S no.	Protocols	Advantages	Drawbacks	Security	Application
1	PoW	Reaches consensus quickly, rules out the possibility of spamming and is most tested over time	Lot of power consumed and wasted in the mining process, hardware dependency can lead to mining centralization	PoW is open to 51% attack along with selfish mining and eclipse attack	Bitcoin, Litecoin, Ethereum, bitcoin cash, Zcash
2	PoS	It is advantageous over other protocols because of reduced energy consumption, easy staking, and being environmentally friendly	One basic disadvantage is that large stakeholders dominate the network	Despite being secure against 51% attack, it is vulnerable to nothing-at-stake, grinding, long-range attacks	Peercoin, Ethereum 2.0, Nxt, ShadowCash, Qora, BlackCoin
3	DPoS	It offers better reward distribution and secured real-time voting; apart from these advantages, it is more scalable than other protocols	Two main demerit points of this protocol are cartel formation and being vulnerable to 51% attack	Vulnerable to bribing and DDoS attack vectors; apart from these attacks, the two other major security concerns are block producers can collude which can launch other major attacks like double spends and there is a possibility of exploitation of low voter turnout	BitShares, Steem, Steemit, EOSIO
4	PoET	Overcomes the need for resource-intensive mining and makes the process of mining completely randomized	Involvement of a third party (SGX) and use of TEE making the use of the protocol limited	Open to Sybil attack	Hyperledger Sawtooth
5	PBFT	Transaction finality, higher energy efficiency, and low reward variance are some of the advantages offered by PBFT	PBFT mechanism works fine with only small consensus group sizes and if the primary node does not work, the implementation is to be done all over again thus reducing the efficiency	Open to Sybil attacks	Hyperledger fabric, Zilliqa
6	PoA	PoA provided increases the speed transaction validation and works as a platform for the development and maintenance of DApps	Decentralization is not considered; the specifications of validators are accessible to anyone and the threat of harming the reputation does not really shield a person from taking part in pernicious activities	Secure against denial-of-service attacks and 51% attack to some extent	Aura, clique

10. Conclusion

The distributed ledger, a disruptive technology, powered by consensus protocols, with its adaptability and application has revolutionized the business processes. In this paper, we surveyed and analyzed a few consensus protocols. No consensus protocol is being perfect, and there are always certain trade-offs related to performance, security, and scalability efficiency. Each of these protocols provides domain-specific solutions and serves different purposes in spite of having their strengths and weaknesses. Above all, they all serve as a common solution for one of the main problems of distributed ledger, i.e., double-spending. Presently, the trend is shifting towards a hybrid approach; that is, implementation will be based on two or more consensus protocols. The consensus protocol, the backbone of a blockchain, comes in varied implementations to serve different use cases. Since the inception of the first consensus protocol, i.e., Proof of Work, researchers are working to develop a scalable, efficient, and secure consensus protocol that could produce excellent results and could help in the growth of the economy and infrastructure.

Data Availability

There are no data available for this work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” 2019.

[2] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: architecture, consensus, and future trends,” 2017.

[3] H. Natarajan, S. Krause, and H. Gradstein, *Distributed Ledger Technology and Blockchain*, World Bank, London, UK, 2017.

[4] I. Bashir, *Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained*, Packt Publishing Ltd., London, UK, 2018.

[5] W. Wang, D. T. Hoang, P. Hu et al., “A survey on consensus mechanisms and mining strategy management in blockchain networks,” *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[6] Q. Antonius, A. A. Jillepalli, M. A. Haney, and F. T. Sheldon, “Blockchain: properties and misconceptions,” *Asia Pacific Journal of Innovation and Entrepreneurship*, vol. 34, 2017.

[7] R. A. Andreev, P. A. Andreeva, L. N. Krotov, and E. L. Krotova, “Review of blockchain technology: types of blockchain and their application,” *Intellekt Sist Proizv*, vol. 16, no. 1, pp. 11–14, 2018.

[8] M. Andreas, *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, O’Reilly Media, Inc., London, UK, 2014.

[9] Y. Xiao, N. Zhang, Li Jin, W. Lou, and Y. T. Hou, “Distributed consensus protocols and algorithms,” *Blockchain for Distributed Systems Security*, vol. 25, 2019.

[10] R. Zhang, R. Xue, and L. Liu, “Security and privacy on blockchain,” *ACM Computing Surveys*, vol. 52, no. 3, pp. 1–34, 2019.

[11] A. Narayanan, B. Joseph, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press, London, UK, 2016.

[12] J. Reed, “Litecoin: an introduction to litecoin cryptocurrency and litecoin mining,” 2017.

[13] F. Haffke, “Technical analysis of established blockchain systems,” Master’s thesis, Technical University of Munich, SW Engineering for Business Informatics, New York, NY, USA, 2017.

[14] H. Cho, “Asic-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols,” *IEEE Access*, vol. 6, pp. 66210–66222, 2018.

[15] S. Sayeed and H. Marco-Gisbert, “Assessing blockchain consensus and security mechanisms against the 51% attack,” *Applied Sciences*, vol. 9, no. 9, p. 1788, 2019.

[16] R. Fitri Sari, “Evaluation of proof of work (pow) blockchains security network on selfish mining,” 2018.

[17] A. Gervais, G. O Karame, K. Wüst, V. Glykantzis, R. Hubert, and S. Capkun, “On the security and performance of proof of work blockchains,” 2016.

[18] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, “Eclipse attacks on bitcoin’s peer-to-peer network,” 2015.

[19] A. Kiayias, A. Russell, B. David, and O. Roman, “Ouroboros: a provably secure proof-of-stake blockchain protocol,” 2017.

[20] E. Ferreira Jesus, R. L. Vanessa, C. VN de Albuquerque, and A. A. D. A. Rocha, “A survey of how to use blockchain to secure internet of things and the stalker attack,” *Security and Communication Networks*, vol. 34, 2018.

[21] F. Irresberger, *Coin Concentration of Proof-of-Stake Blockchains*, Leeds University Business School Working Paper, London, UK, 2018.

[22] S. King and N. Scott, “Ppcoin: peer-to-peer crypto-currency with proof-of-stake,” 2012.

[23] B. Stiller, “Bazo: a proof-of-stake (pos) based blockchain,” Technical Report, World Bank, Zürich, Switzerland, 2019.

[24] H. Chen, M. Pendleton, L. Njilla, and S. Xu, “A survey on ethereum systems security,” *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–43, 2020.

[25] Abdul Wahab and W. Mehmood, “Survey of consensus protocols,” 2018.

[26] V. Buterin, D. Ryan, H.-W. Wang, T. Tsao, and C. C. Liang, “Ethereum 2.0 phase 1–shard data chains,” 2018.

[27] Y. Gao and H. Nobuhara, “A proof of stake sharding protocol for scalable blockchains,” *Proceedings of the Asia-Pacific Advanced Network*, vol. 44, pp. 13–16, 2017.

[28] B. Pascal, “Survey of consensus protocols and scalability solutions,” 2018.

[29] K. Qin and A. Gervais, *An Overview of Blockchain Scalability, Interoperability and Sustainability*, Hochschule Luzern Imperial College London Liquidity Network, New York, NY, USA, 2018.

[30] M. Snider, K. Samani, and T. Jain, “Delegated proof of stake: features & tradeoffs,” 2018.

[31] F. Schuh and D. Larimer, “Bitshares 2.0: general overview,” 2017.

[32] C. Li and B. Palanisamy, “Incentivized blockchain-based social media platforms: a case study of steemit,” 2019.

[33] Y. Huang, H. Wang, L. Wu et al., “Characterizing eosio blockchain,” 2020.

[34] N. He, R. Zhang, L. Wu et al., “security analysis of eosio smart contracts,” 2020.

[35] A. Corso, “Performance analysis of proof-of-elapsed-time consensus in the sawtooth blockchain framework,” 2019.

[36] V. Costan, “Intel sgx explained,” 2016.

[37] B. Hill, S. Chopra, V. Paul, and N. Prusty, *Blockchain Developer’s Guide: Develop Smart Applications with Blockchain Technologies-Ethereum, JavaScript, Hyperledger Fabric, and Corda*, Packt Publishing Ltd., London, UK, 2018.

[38] K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton, and C. Montgomery, *Sawtooth: An Introduction*, The Linux Foundation, New York, NY, USA, 2018.

[39] J. R. Douceur, “The sybil attack,” 2002.

[40] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, “On security analysis of proof-of-elapsed-time (poet),” 2017.

[41] M. Castro and B. Liskov, “Practical byzantine fault tolerance,” *OSDI*, vol. 99, pp. 173–186, 1999.

[42] M. Castro and B. Liskov, “Practical byzantine fault tolerance and proactive recovery,” *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.

[43] M. Du, X. Ma, Z. Zhang, X. Wang, and Q. Chen, “A review on consensus algorithm of blockchain,” 2017.

[44] N. Gaur, L. Desrosiers, V. Ramakrishna, P. Novotny, S. A. Baset, and O. ’D. Anthony, *Hands-on Blockchain with Hyperledger: Building Decentralized Applications with Hyperledger Fabric and Composer*, Packt Publishing Ltd., London, UK, 2018.

[45] P. Sajana, M. Sindhu, and M. Sethumadhavan, “On blockchain applications: Hyperledger fabric and ethereum,” *International Journal of Pure and Applied Mathematics*, vol. 118, no. 18, pp. 2965–2970, 2018.

[46] Q. Wang, J. Huang, S. Wang, Y. Chen, P. Zhang, and Li He, “A comparative study of blockchain consensus algorithms,” *Journal of Physics: Conference Series*, vol. 1437, 2020.

[47] Y. Liu, K. Qian, J. Yu, K. Wang, and He Lei, “Effective scaling of blockchain beyond consensus innovations and moore’s law,” 2020.

[48] K. Olson, “The zilliqa project: a secure, scalable blockchain platform,” 2018.

[49] N. Chondros, K. Kokordelis, and M. Roussopoulos, "On the practicality of practical byzantine fault tolerance," 2012.

[50] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[51] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain," 2018.

[52] M. Kuperberg, "Towards an analysis of network partitioning prevention for distributed ledgers and blockchains," 2020.

[53] S. De Angelis, "Assessing security and performances of consensus algorithms for permissioned blockchains," 2018.

[54] X. Liu, G. Zhao, X. Wang et al., "Mdp-based quantitative analysis framework for proof of authority," 2019.