Chapter 4
Blockchain Consensus Algorithms: Study and Challenges

Avita Katal, Vitesh Sethi, and Saksham Lamba

4.1 Introduction to Blockchain Technology

Before blockchain technology, when customers used to make transactions, they had to trust a third party that ensured the transaction is valid, so that it can be put into action. But sometimes the middle party cannot be trusted blindly as it could be exploited. The root cause for this problem is formal centralization, in which there is a single centralized authority on which everything depends. This problem was avoided by the use of blockchain which is based on decentralization. Blockchain was developed by Haber and Stornetta [1] and uses multiple independent organizations to validate the transactions. In more formal words, blockchain is defined as:

The distributed and decentralized database of public and private ledger of all transactions or digital assets that is unalterable and visible to each and every one who is associated with that particular network with the use of cryptographic hashing.

This technology gained popularity after the introduction of Bitcoin by Satoshi Nakamoto and group [2] in the year 2008. The main reason to introduce Bitcoin was to overcome the problems of traditional transactional methods, i.e. trust in third parties. The ledger in Bitcoin contains the records of all the transactions. It is established containing many blocks with transactions inside them. That is how the name Blockchain was introduced. The previous block could be referred by the block using a hash value. The first block added to the chain is known as Genesis block that contains the first transaction. The blocks are arranged according to a particular chronology. The transactions are verified by the nodes which are a part of the network. Once the transaction is considered to be valid and digitally

A. Katal (✉) · V. Sethi · S. Lamba
Department of Virtualization, University of Petroleum and Energy Studies, Dehradun, India

T. Choudhury et al. (eds.), *Blockchain Applications in IoT Ecosystem*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-65691-1_4

signed by the sender, the block containing the transaction details is added to the chain and thereafter cannot be changed. This method of verification of transactions can be confusing since each node would broadcast their block information. To find a solution to this problem, a consensus is made among all the nodes on what all blocks must be mined and which all nodes would have the authorization to make changes to their proposed blocks. Blockchain has different deployment strategies: Public Blockchain and Private Blockchain [3]. A Public blockchain or Permissionless or Unpermissioned blockchain is the one in which any node can enter and leave the chain at any point of time. Private blockchain also known as permissioned blockchain is different from public blockchain as it allows only trusted or authorized nodes to participate, thus ensuring privacy of the chain data. They can further be categorized as Consortium blockchain and Fully Private blockchain. In the Consortium blockchain, few nodes validate a transaction [3]. In the next section of the chapter, consensus algorithms will be discussed in detail.

4.2 Introduction to Consensus Algorithms

Consensus algorithms are considered to be an important component of blockchain technology which helps in achieving a tamper free environment. It accepts only one form of truth and is accepted by all the miners present in the network. This technology uses consensus mechanisms among nodes to verify the information, thus preventing the need of intermediaries [4]. All the nodes present in this network should come up with a consensus about the present state of the blockchain. This improves the security of the system as it becomes difficult for the attacker to create a tampered block and introduce it in the decentralized network [5]. Thus, choosing the correct consensus algorithm is very important to implement a blockchain solution. In addition to this, consensus algorithms help to solve two major problems in blockchain, Double Spending in which the same currency is used in two transactions at the same point of time and Byzantine Generals Problem which occurs in the distributed systems.

Double spending occurs when the cryptocurrency is stolen from the blockchain network. The attacker creates and sends a copy of the currency transaction so that it appears to be legitimate. The most common method used by the attacker is to send multiple packets to the network, and reverse the transactions so that it appears that the transactions never happened. Double Spending is solved by validating the transactions by many nodes in the distributed network. Since the data can be communicated among various nodes, some of the nodes may be attacked leading to alteration in communication which is also known as Byzantine Generals Problem. To distinguish the data that has been changed and to get efficient results with the help of other miners, consensus algorithms may be used.

In certain architectures, consensus can be easily achieved under the following scenarios when:

- There is no fault in the system and each node can receive and transfer the messages correctly.
- The system is synchronized.

Three properties that ensure the correctness of the distributed consensus protocol are [6]:

- *Safety/Consistency*: It makes sure that nodes involved in the process of consensus will never converge to an incorrect state.
- *Liveliness/Availability*: The acceptance of each and every correct value occurs eventually or in other words, all non-faulty nodes participate in the consensus process and produce an output.
- *Fault Tolerance*: The blockchain network performance goes unchanged even amidst node failures.

The traditional distributed consensus algorithms are based on certain approaches. They are:

- A message passing system that requires a cloud environment where each and every entity should know about other entities that are a part of the network.
- The shared memory approach in which a common memory space is created so that shared variables can be read and written by everyone present in the network. This approach cannot be implemented for internet grade computing as a readable and writable memory space needs to be created, so that it can be accessed by every user which is a part of the network. Similarly, message passing is not suitable for an open environment like the Bitcoin system where anyone can be a part of the network at any point of time.

4.3 Consensus Algorithms Characteristics

This section describes the different characteristics of the consensus algorithms. The consensus algorithm properties can be categorized into: Structural, Block and reward, Security and Performance [7].

4.3.1 Structural Properties

These properties tell how various nodes/miners in the decentralized network are arranged to take part in a consensus algorithm. The structural properties have the following categories:

(a) Node type: It helps in achieving consensus by engaging different types of nodes in the consensus algorithms. The node types depend upon the consensus algorithm like some of the algorithms may have clients, miners, minters, validators, electors and stakeholders, etc. Node types will be discussed in detail in the coming sections.

(b) Structure type: It defines different ways of structuring nodes within the consensus algorithm by utilizing the committee mechanism. The committee can be subdivided in two types: single and multiple committees.

- Single committee: It defines a particular set of nodes among the nodes that take part in the consensus mechanism by the production of blocks and the extension of the blockchain network.
- Multiple committee: The time required to achieve consensus in a single committee increases as the number of nodes in a decentralized network increases. This reduces the efficiency. To maintain network performance, the concept of multiple committees is introduced in which each committee has different validators.

(c) Underlying mechanism: It is a mechanism of selecting a particular node that is deployed by a consensus algorithm. This mechanism utilizes lottery, coin age or a voting mechanism. A lottery uses a probabilistic mechanism which is based on cryptography or the other mechanisms that are randomized [7]. Voting can take place in single rounds or multiple rounds. The property used by coin age depends on the time for which owner owns a coin.

4.3.2 Block and Reward properties

These properties are used to differentiate the cryptocurrencies on the basis of quantitative metrics. Some of the properties are genesis date, block reward, total supply, formula and block creation time [7]. Even though these properties do not represent different consensus algorithms in a direct manner, they do have an impact on how consensus is achieved in cryptocurrency based blockchain networks. Some of the properties are mentioned below:

(a) Genesis Date: shows the timestamp of the first block created for a specific cryptocurrency.
(b) Block Reward: refers to the incentive that a miner achieves when he creates a new block.
(c) Total Supply: shows the total amount of cryptocurrency that is supplied.
(d) Block time: it refers to the average time taken for block creation of cryptocurrency.

4.3.3 Performance Properties

The performance properties are used to calculate the efficiency of a consensus algorithm. Some of the properties are described below:

(a) Fault tolerance: represents the number of defective nodes that a consensus algorithm can handle.
(b) Throughput: refers to the rate at which the transactions are processed.
(c) Scalability: means increasing the size and functionality of the system without affecting the throughput of the original system.
(d) Latency: defined as the total duration taken by the consensus to reach out and process the proposed transaction.
(e) Energy consumption: addresses whether the mechanism or the using system has high energy consumption.

4.3.4 Security Properties

The consensus protocols must follow the following security properties:

(a) Authentication: Ensures that the nodes participating in a consensus protocol are verified and authenticated.
(b) Non-repudiation: Checks if a consensus protocol satisfies non-repudiation i.e. it cannot deny the validation of transaction.
(c) Censorship resistance: Implies whether the corresponding algorithm is able to withstand any censorship resistance.
(d) Attack vectors: A combination of attack vectors that are relevant to consensus algorithms are presented.
(e) Adversary tolerance: Represents the maximum Byzantine nodes that can be accepted by the protocol.
(f) Sybil protection: In this type of security breach, the thief duplicates its identity to achieve the advantages that are against rules and protocols. In a blockchain network, the Sybil attack can be implicated when an adversary creates many nodes according to the need in the underlying peer to peer network so as to influence the consensus algorithm and gain advantage from it.
(g) Denial of Service (DoS) resistance: Ensures whether the consensus algorithm has mechanisms that are against the DoS attacks.

4.4 Consensus Algorithms

In this section, various algorithms that are used to achieve consensus in blockchain will be discussed in detail. These algorithms are required to provide equality

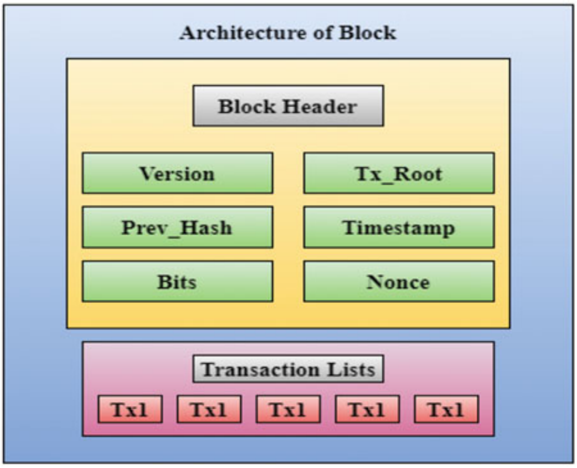


Fig. 4.1 Architecture/various fields of a block in a blockchain

and fairness to the whole system. Consensus algorithms in blockchain can be categorized as Proof based consensus algorithms and Voting based consensus algorithms. Before getting into the details of consensus algorithms, the block structure in the blockchain needs to be discussed and is shown in Fig. 4.1.

- *Prev Hash*: It can be defined as the connection of the block to its previous one and can also be considered as the reference to the parents.
- *Timestamp*: It refers to the time duration at which the block was obtained.
- *Tx Root*: Tx Root is also called the Merkle root. This field consists of all the hash values of the verified transactions contained inside a block. All the transactions that are present in the block are hashed by SHA 256 algorithm into a hash value. After hashing is done, these transactions are combined with each other, pair by pair and again put into some other hash function. The above process continues till only a particular value is obtained which is called as the Merkle root.
- *Version*: It refers to the protocol version which is used by each node in order to put the block into the chain.
- *Nonce*: Nonce also called as ‘number only used once’ is a pseudo random number that acts as a counter during a process of mining. It also describes the efforts made by node to append a block.
- *Bits*: This field generally shows the complexity of Proof of Work.

4.4.1 Proof Based Consensus Algorithm

Many types of Proof based consensus algorithms have been implemented, that are based on Proof of Work (PoW), Proof of Stake (PoS) or a combination of both and various other types that are independently made from the two important ones that are listed [8]. Proof based algorithms concept works around the fact that among numerous nodes, part of the network, only the node with suitable proof will have the permission to append the node to the chain.

4.4.1.1 Proof of Work (PoW)

If each and every node in a blockchain network tries to present their blocks holding validated transactions, it will result in a lot of confusion. To get a solution to this issue, Proof of Work algorithm is introduced.

PoW was introduced in 1992 by Dwork and Naor in order to stop junk/spam mails in which the user had to do some work so that he or she can send and receive a valid email [9]. Proof of Work allowed only trusted users to calculate the result of the puzzle, thus preventing the attacker from sending junk mails. The receiver received the mail only when the result was correct. As discussed in section 3, Node types are one of the properties of consensus algorithms. In PoW, two types of nodes exist: Requestors and Verifiers.

One of the most important features of PoW is asymmetry [6], which ensures that even if the task is relatively complex it should be feasible for the service requestor.

In Bitcoin, PoW is used to extend the hashcash based PoW system so as to come up with the methodology in protecting the blockchain through distributed consensus mechanism. Hash cash system is based on the puzzle friendliness property which is a part of cryptographic hash function. In blockchain, before the puzzle is solved, the verifying nodes must put their validated transactions including information like hash of previous block (Prev Hash) and Timestamp into a block. Each puzzle is solved by guessing a secret value known as nonce field which should be present in the block.

All of this information present in the block header is combined and then added in a SHA 256 hash function. The secret value is accepted only if the result of the hash function is less than a given threshold which represents the complexity. Otherwise, node keeps guessing the other secret value until the correct answer is obtained. To ensure that the average speed for the addition of the block in the chain is 1 block per 10 min, the difficulty of the puzzle is adjusted and managed after appending every 2016 blocks [8]. The threshold value is based on the complexity of the puzzle. The more the difficulty level, the lesser the threshold value. The work for predicting the correct value is called Proof of Work (PoW). The node which joins the network using the PoW is known a miner and the work for getting a correct nonce is called as mining.

The proposed block is broadcasted to the other nodes by a particular node when the secret value has been found. This process is followed to notify the other nodes. After getting the signal the nodes who still have not got the correct mysterious value of their puzzles will stop guessing and would start checking whether all the transactions in the broadcasted block are valid or not. The proposed block will be added to the current chain if all the verifications are validated. Figure 4.2 explains the various steps in PoW.

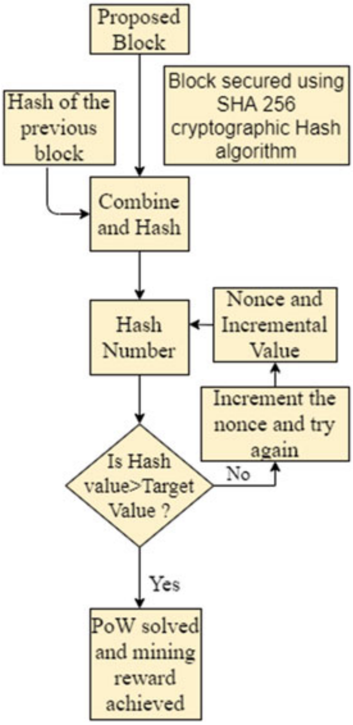

Proposed Block
Block secured using SHA 256 cryptographic Hash algorithm
Hash of the previous block
Combine and Hash
Hash Number
Nonce and Incremental Value
Increment the nonce and try again
No
Is Hash value>Target Value ?
Yes
PoW solved and mining reward achieved

Fig. 4.2 Flow chart representing proof of work algorithm

Sometimes, a case may occur when one miner gets the suitable answer for the puzzle before it is informed by some other miner. In this case, the block would still be presented by the miners with the discovered nonce. The other miners when receiving the first incoming block would neglect other blocks coming thereafter. This will lead to the Forking Problem in which there are various chains of blocks in the validated network. Satoshi proposed that the nodes who have got the correct answers will continue appending a fresh block on their respective forks, till one fork will be greater than the other. Therefore, the longest fork has to be followed by all the nodes at this time. After performing all the tasks, when the verified block is put into the chain, the node which appends this block gets rewarded in the form of bitcoins. The solution to forking problem is shown in Fig. 4.3.

4.4.1.2 Proof of Stake (PoS)

PoS provides equal opportunity to all the miners. The PoS consensus algorithm decides which node would get the permission to append the consecutive block on the basis of the stake it owns. This method can be very advantageous as the miner with more stake would be more trustful. PoS also provides the required security to

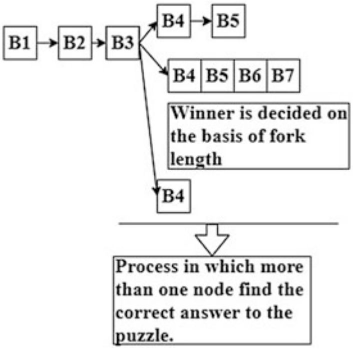

B1
B2
B3
B4
B5
B4
B5
B6
B7
Winner is decided on the basis of fork length
B4
Process in which more than one node find the correct answer to the puzzle.

Fig. 4.3 Solution to the forking problem

the network because any attacker cannot perform a double spending attack until or unless he or she owns at least 51% [8] of the total stakes in the network.

The implementation of PoS can be seen in Nextcoin [10] which ensures that the miner having more stake will get the opportunity to mine a recent block.

Bentov et al.[11] gave a mechanism similar to Nextcoin which stated that the chance of a miner to append a block lies on the amount of stake he owns. The more the stake the node owns, the more would be the chance of him appending a block. Bentov also proposed the procedure called the Satoshi procedure.

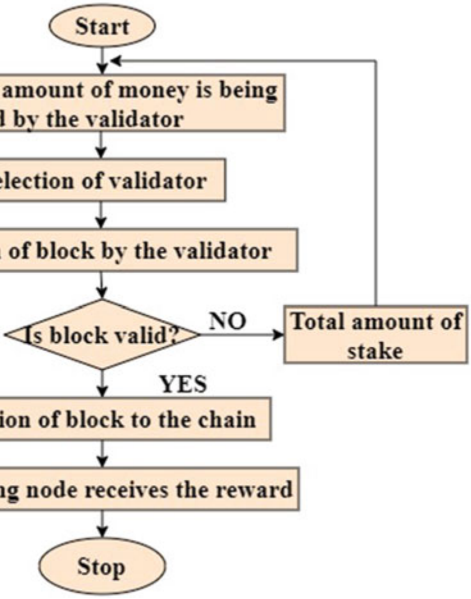
Kiayias et al.[12] implemented the same technique as Bentov, to execute PoS consensus by following the procedure of Satoshi. They proposed that the leader selection should be done randomly by some calculation that should be secure enough. This would prevent the protocol from predicting the calculation easily. Leader selection is the process of selecting a particular miner to issue the next block. The steps for PoS are shown in Fig. 4.4.

4.4.1.3 Implicit Consensus

In the implicit consensus model, individual blockchain is processed by each node. Unbound throughput is one of the major advantages of this consensus model. Here special types of blocks known as check point blocks are considered instead of the transaction blocks. Since consensus cannot be implemented in each and every transaction, the implicit consensus algorithm leads to the scalability of the scheme in addition to linear message complexity. Some of the main features of this consensus model are the following [13]:

- With the help of self-interest formula this algorithm replaces the termination property of Byzantine Fault Tolerance schemes.
- Agreement and correctness for each transaction will hold on till the transaction is validated by the validation scheme locally.

Fig. 4.4 Flow chart representing proof of stake algorithm


Start
Some amount of money is being staked by the validator
Selection of validator
Creation of block by the validator
Is block valid?
NO
Total amount of stake
YES
Addition of block to the chain
Validating node receives the reward
Stop

4.4.1.4 Secure Sharding Algorithm (ELASTICO) [14]

ELASTICO consensus algorithm is a great asset for permissionless blockchain. It is a scalable agreement algorithm in which the transaction rate varies proportionally to the processing present in the mining process i.e. more transactions can be processed in the blocks with more computational power. The idea upon which this algorithm works is to segregate the network into smaller parts which are known as committees. Each committee is responsible for the processing of a disjoint arrangement of exchanges and the entire technique is parallelized simultaneously.

4.4.1.5 Hybrid Algorithms (PoW/PoS)

The hybrid consensus algorithms use the combined features of PoW and PoS so as to overcome the weakness of each of them. PoW suffers from intensive energy consumption as it requires plenty of computational power. Also, PoW is vulnerable to a ‘51% attack’, meaning that if the node earns 51% of the entire computational energy in blockchain network, that miner/node can make changes in the blockchain network. It can also introduce double spending which can be a serious problem. Similarly, in PoS only the rich stakeholders are allowed to control the consensus in the blockchain. And only those stake owners are allowed to have a control of consensus in blockchain.

The first variant that implemented a hybrid consensus algorithm was PPcoin proposed by King and Nadal [15]. A term called ‘coin age’ was defined for each node in the network which is computed by multiplying the stake by the time, till the time the node has earned that stake.

Vasin [16] did not associate coin age with his Blackcoin. The concept behind his idea was that with coin age there is an increased chance of attackers to collect sufficient amounts. Also, some nodes would keep a hold on their own money till they receive sufficient amounts of coin age, when they are not online in the validation system. To overcome this problem, Vasin [16] proposed that instead of using coin age, raw stake would provide the miners to append a new block. Thus, ensuring that more nodes are online in order to gain rewards.

If the nodes owned greater than 51% of the computational power, there would be a high security risk. Duong et al. [17] thought of mitigating the double spending attack using the Hybrid consensus algorithm. Duong et al. [17] addressed that each of the Proof of Stake blocks is connected to another Proof of Work block and each Proof of Work block is connected to a preceding Proof of Stake block. Thus, it becomes tough to make a double spending attack. This attack can only happen when the attacker owns 51% of the mining power but also he should own greater than 50% of all the stakeholders.

The main problem associated with the work proposed by Duong et al. [17] was that initially the executing environment was not dynamic because the physical hardware that was invested and stake remained unchanged. To improve this proposed method, Chepurnoy et al. [18] proposed a method to adjust the difficulty on the basis of the environment the rate at which the block is created.

4.4.1.6 Proof of Stake Velocity (PoSV)

To enhance security in the blockchain network, another consensus called Proof of Stake Velocity algorithm was implemented in 2014 [19] as a substitute to PoS and PoW algorithm. The PoSV was first used in Reddcoin's inception and was based on the traditional PoS algorithm. The idea behind using PoSV was to validate the transactions of the Reddcoin, which was the cryptocurrency developed mainly for social interactions in the digital age. PoSV ensured ownership (stake) and activity (velocity), the two major functions of Reddcoin as a real currency. The formula to calculate the velocity of money in a particular time is: [19]

$$V_T = nT/M \tag{4.1}$$

where V_T is the velocity with which the money flows, nT stands for the aggregate notional of transactions and M is the total amount of stake which is in flow. One of the disadvantages of PoSV is that it is particularly designed for the digital social currency; Reddcoin cannot be implemented for other cryptocurrencies. PoSV is evaluated as a part of the Reddcoin system and not as a standalone.

4.4.1.7 Proof of Activity (PoA)

Bentov et al. [20] came up with the idea and implemented a combination of Proof of Work and Proof of Stake called Proof of Activity consensus algorithm. This algorithm stopped the double spending attack as well as examined the tragedies made by Proof of Work known as tragedies of common. The first tragedy is that only miners who solve the puzzle get the reward, whereas the ones who preserve the ledger, update it and validate the new block do not get any reward. Another tragedy is that the nodes can cooperate with other miners in order to increase the transaction fee to charge from the user. This would lead to the less usage of blockchain technology. So as to overcome these tragedies, the researchers came up with the idea to create a vacant block by Proof of Work where all the nodes would try to crack the nonce associated with the block with no transactions. The block is broadcasted to other nodes for authentication when the nodes find the correct nonce. On the basis of the received block, they would also check if they have won another lottery. This fortunate chance is similar to that of followed in Satoshi procedure (this method receives index as an input of a Satoshi (smallest value of cryptocurrency) between 0 and the entire Satoshis in distribution. It gets the block from the ledger data into which Satoshi is appended and keeps a check on the transactions which had carried this Satoshi to the consecutive addresses. This process continues till it gets the stakeholder who could presently spent this Satoshi) in [21].

4.4.1.8 Proof of Burn (PoB)

Ian Stewart [22] proposed another consensus algorithm called Proof of Burn. In PoB, the miners have to first burn their coins in order to take part in the mining activity. Burning of coins refers to the sending of the coins to an address without the private key so that coins are never usable. This means that burning of coins is similar to the idea of investing for the building of the mining rig. The value of the coins burned has a specific relation with the probability of being chosen to mine the following block. This process is more or less like proof of work in which the nodes, in order to uphold the hash power invest in modern equipment.

Cryptocurrency implements the idea of Proof of Burn in combination with Proof of Work and Proof of Stake is Slimcoin [23]. The concept is very much identical to the Proof of Stake algorithm with additional PoB mechanism which is sandwiched between PoW and PoS mechanism. For the generation of initial coin supply using the bitcoin mechanism, PoW is used. It plans to transfer to the hybrid mechanism which consists of PoW and PoS when enough amount of stake is supplied to the system similar to Peercoin where miner is selected by the PoB. For the participation in the PoS minting process, the miners have to burn their accumulated coins. The PoB mechanism is generally used in the selection of minter without affecting the security of the network.

4.4.2 Voting Based Consensus

For the implementation of the consensus mechanism based on voting, miners in the blockchain network must be recognized and should be adjusted, so that message passing between the nodes becomes easier. Before deciding to mine the proposed block they would have to first communicate with other nodes in the network.

The implementation of the voting based consensus algorithm is very much identical to the conventional methods in similarity to tolerate the faults in the network. The Voting based consensus is proposed to solve some of the problems that arise in the blockchain network:

- Crashing of nodes.
- Damage of the well established system by the nodes.

4.4.2.1 Proof of Trust (PoT) [24]

The Proof of Trust algorithm enhances the efficiency of crowdsourcing services. The Shamirs secret sharing algorithm and RAFT leader election algorithm are used for the election of these nodes. PoT is implemented through four phases. Phase 1 uses the Raft leader algorithm for leader selection. Phase 2 is used for the selection of validators of the transactions. The next phase focuses on the validation of transactions by those which were selected in the previous phase and the fourth phase links the verified transactions with the blockchain network. This algorithm implements fault tolerance in the network when [24]

$$p => 3q + 1 \tag{4.2}$$

where p represents the nodes which participate in the blockchain network and q stands for the number of Byzantine nodes.

4.4.2.2 Proof of Vote (PoV) [25]

The PoV algorithm is responsible for the validation of blocks by using the voting process. The Proof of Vote algorithm is superior from all algorithms in terms of power usage or power consumption. There are 4 roles that are described in a consortium network model: commissioner, butler candidate, butler, and ordinary user.

- *Commissioner*: Various organizations around the globe form a league committee so that the consortium blockchain is being maintained. One of the members is called the commissioner who is selected by the alliance law. The machine working is responsible to represent the commissioner in a consortium decentralized network. The commissioner evaluates, recommends and votes for the butler. They also validate and process blocks and transactions. The block is considered to be verified when it gets at least 51% of the votes.

- *Butler*: The production of blocks is done by the butler. Butler is specially designed for segregating voting and execution right. The butler collects the data related to a transaction from a network and puts all the information into a block. After putting all the data, butlers need to put their signature on the block.
- *Butler Candidate*: Since there are only a few number of butlers, the election of the butler should be done by the butler candidates, and the candidates will be voted by all the commissioners. If any candidate is not selected for the butler, it can remain active and can wait for the upcoming elections.
- *Ordinary user*: Ordinary users need not to authorize their identity and they have the permission to enter and exit the network anytime they want. They do not have the right to take part in the procedure of creation of blocks and can only participate in block distribution and message passing. The whole consensus mechanism can be seen by them while utilizing the services of the network.

4.4.2.3 Ripple Algorithm [26]

The Ripple consensus algorithm makes use of subnetworks that are considered to be trusted within the decentralized network. So as to ensure the correctness of the network system, the protocol is deployed after a few seconds. The ledger is considered to be closed after the consensus is achieved. The last closed ledger that is executed by each node in the network should be the same. This protocol runs in different rounds. At the beginning, a candidate list is being announced publicly by each of the nodes in a network which contains all the verified transactions. After this process, voting is done by every node so as to check the correctness of all the transactions. It then combines the candidate list prepared by all the nodes. In order to consider a transaction as verified the number of votes received is compared with the threshold value and the decision is made on the basis of it. The last round requires at least 80% of the nodes participating in the network must reach consensus on the transaction. When the transaction meets all of the above mentioned criteria, it is considered to be validated and is then applied to a ledger. Thus, creating a fresh closed ledger.

4.4.2.4 Practical Byzantine Fault Tolerance (PBFT)

Castro and Liskov [27] implemented the Practical Byzantine Fault Tolerance consensus algorithm. PBFT consists of two categories of nodes: A leader node and a few verifying nodes. For the addition of a block in the chain, the miners need to execute some rounds of mining. After this process, the execution of three phases of PBFT occurs. The first phase is called the Prepare phase, in which the proposed block is being presented to the other nodes by the leader. This block is stored locally. To validate the authenticity of the block that is being received, the nodes re-evaluate through presenting it in the prepare phase and commit phase. When the node gets the same block which is stored locally after the prepare phase or more than two third of the entire nodes present in the blockchain system, the commit phase is executed. The similar process is followed even after the commit phase that is the major need of the node for the processing of the transaction in a particular block and then mining it to the current chain.

4.4.2.5 Delegated Byzantine Fault Tolerance (DBFT)

The DBFT consensus algorithm was initially implemented in NEO blockchain and was proposed by Da HongFei and Erik Zhang [28]. DBFT consensus can be achieved in a public network in a very fast manner.

Since DBFT can also work with few miners and the system can also handle up to [28]

$$(p-1)/3 \tag{4.3}$$

faulty nodes where p stands for a group of consensus nodes and not like PBFT where p represents the nodes that take part in the blockchain network.

Delegated Byzantine consensus algorithm works on the voting mechanism. All the nodes that are a part of a network and have NEO token are called ordinary nodes. These nodes have the power to implement transactions in the network and voting procedure for consensus nodes in real time. The consensus nodes contain the speaker and delegates. The responsibility of the speaker is to fetch the transactions from the memory, verify them and then put them into a new block whereas delegates are responsible for verifying the blocks by the voting mechanism.

4.5 Comparison of Different Consensus Algorithms

This section compares the various Consensus algorithms based on generic and performance parameters as shown in Table 4.1

4.6 Research Challenges and Future Scope

This section discusses the main issues in consensus algorithms in blockchain technology. Some of them are as:

(a) Security Problems:

Security issues in consensus algorithms lead to various security attacks. This can lead to unauthorized access of a blockchain network by the individuals who

Table 4.1 Comparison between different consensus algorithms

Algorithms	Blockchain type	Mining	Category	Scalability	Latency
ELASTICO (2016)	Permissionless	On the basis of computational power	Proof based	Scalable	Low
Implicit consensus (2017)	Permissioned	On the basis of proof based	Proof based	Not Scalable	High
Proof of trust (2018)	Permission based consortium	Based on probabilistic and voting mining	Vote based	Scalable	Low
DBFT Consensus Algorithm (2018)	Permissioned	Non-proof of based mining(Random selection of miner)	Vote based	Not Scalable	Very low
Ripple (2014)	Permissioned	On the basis of voting mining	Vote based	Scalable	Low
Proof of vote (2017)	Consortium	On the basis of voting mining	Vote based		Very low
Proof of work (2008)	Permissionless	On the basis of computational power	Proof Based	Not Scalable	Very high
Proof of Stake (2011)	Permissioned and permissionless	On the basis of nodes wealth and staking age	Proof based	Scalable	High
Proof of stake velocity (2014)		On the basis of stake and amount (velocity)	Proof Based Hybrid (PoW/PoS)	Scalable	Low
Proof of activity (2014)	Permissionless	On the basis of effectiveness Of work by the miner	Proof Based Hybrid (PoW/PoS)	Scalable	Low
Proof of burn (2014)	Permissioned and Permissionless	On the basis of coin burning(Probabilistic lottery)	Proof Based Hybrid (PoW/PoS)	Scalable	Average
Practical Byzantine Consensus algorithm (1999)	Permissioned	On the basis of round of mining	Vote Based	Not Scalable	Very low

act as miners but in real sense they are not miners but attackers. Various forms of attacks which can harm the blockchain system are: Distributed Denial of Service Attacks, Double Spending Attack, Denial of Service Attacks, etc.

(b) Performance Problem:

Performance of the blockchain network is dependent on the consensus algorithm implemented. Since performance problems can decrease the efficiency of the blockchain network by reducing scalability and increasing the latency in the network, choosing the right consensus algorithm becomes a major challenge. Some of the failures affecting performance are: Temporal failure, Omission failure, Transient failure and Software failure [29].

- Temporal failure: This failure occurs due to the latency in the network, though it generates correct results but takes more time to be processed.
- Omission failure: It arises due to transfer problems like buffer overflow and improper functioning of the transmitter.
- Transient failure: This failure is permanent in nature. It occurs in hardware due to the issues in batteries and a sudden spike in power whereas in case of software these issues can be in the form of bugs in the codes which cannot be detected even when the testing face is going on.
- Software failure: These occur because of the flaws in designing and modelling. Software failures can further trigger other failures such as omission.

(c) Consensus mechanisms:

Each consensus algorithm discussed in the above sections performs a particular task for the validation of the transactions. Thus, it becomes necessary to implement the right algorithm for the verification of any transaction. For example, when a stakeholder owns 51% of the total computational power, PoW should not be implemented as it is vulnerable to the double spending attack. Instead a hybrid and PoS consensus algorithm should be implemented.

(d) Energy Management:

The computational power which is being utilized by the consensus algorithms is very high due to their complex mechanisms. Thus, proper management of energy resources is required to prevent its unnecessary usage.

(e) Byzantine failure:

It is a fault that shows various symptoms in the network. It prevents the nodes from reaching a consensus because the system gets confused and cannot handle faults in the network. For example, a particular server operating in the network might appear to be functioning improperly to one of the nodes and operating properly to the other, this server cannot be called as failed as both the nodes would not come to an agreement because both the servers do not have the same information.

Over the time, the evolution of blockchain has developed decentralized applications beyond financial transactions in different areas. Nowadays the need for an open, energy efficient and scalable blockchain becomes important. This is because of the services provided by the blockchain network in a large scale collective ecosystem like social networking, smart healthcare, smart cities and social networking with the aim of cost reduction and green computing. Initially, blockchain network was developed using a public network and it was open allowing anyone to participate disabling access control to data. Security, scalability and consumption of energy were the major threats in the system [30–33]. Afterwards, blockchain architecture and consensus protocols evolved in order to tackle the above mentioned issues. However, one of the major issues which is still a topic of research in these architectures is the high utilization of energy so as to maintain the security of the system. More the number of participants, higher is the consumption of energy in the consensus protocols which also puts a negative impact on the environment.

The Proof and Voting based Consensus protocols were developed in order to overcome the problem of high energy utilization but this led to architectures which were not scalable. A possible solution should be developed that should be less computationally complex and more energy efficient.

Also, inflexible and non-adaptive behaviour of present consensus protocols and architectures act as an obstacle for a growing collaborative digital ecosystem because they target a particular field. These architectures should be modified to provide a suitable environment for the applications.

4.7 Conclusion

With the increasing popularity of blockchain technology, according to International Data Corporation (IDC) the transactions made in the blockchain system could reach a very high value of 12.4 billion dollars by 2022 [34]. Blockchain can be implemented in many sectors like business activities, IoT, Medical informatics, etc. It is expected that this technology would overcome the other technological domains. A suitable consensus algorithm must be used to implement the blockchain technology as it is one of the major components of the decentralized network and dictates the efficiency of the system. In this chapter, we have discussed the consensus algorithms, their categorization, their implementation and usefulness in the blockchain network. We have compared the various mentioned consensus algorithms on different parameters and how the implementation of each consensus algorithm differs from other. Apart from the advantages, we have listed the various research challenges being faced in this subdomain of blockchain.

References

1. S. Haber, W.S. Stornetta, How to time-stamp a digital document. J. Cryptogr. **3**(2), 99–111 (1991)
2. S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System* (2008). https://bitcoin.org/bitcoin.pdf
3. S.M.H. Bamakana, A. Motavali, A.B. Bondarti, A survey of blockchain consensus algorithms performance evaluation criteria. Expert. Syst. Appl. **154**, 113385 (2020)
4. H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, J.J. Kishigami, Blockchain contract: a complete consensus using blockchain, in *Proceedings of the IEEE 4th Global Conference on Consumer Electronics* (2015), pp. 577–578

5. Ashok Kumar Yadav, Karan Singh.: Comparative analysis of consensus algorithms of blockchain technology, ambient communications and computer systems, in *Advances in Intelligent Systems and Computing*, vol 1097 (Singapor, Springer, 2020), pp. 205–218
6. S.S. Panda1, B.K. Mohanta, U. Satapathy, D. Jena, D. Gountia, T.K. Patra, Study of blockchain based decentralized consensus algorithms, in *IEEE Region 10 Conference* (2019)
7. Md.S. Ferdous, M.J.M. Chowdhury, M.A. Hoque, *Blockchain Consensus Algorithms: A Survey* (2020). shortcomarxiv: 2001.07091v2 [cs.DC]
8. G.-T. Nguyen, K. Kim, A survey about consensus algorithms used in blockchain. J. Inf. Process. Syst. **14**(1), 101–128 (2018). https://doi.org/10.3745/JIPS.01.0024
9. C. Dwork, M. Naor, Pricing via processing or combatting junk mail, in *Proceeding of the Annual International Cryptology Conference* (Springer, Berlin, 1992), pp. 139–147
10. Nxt Whitepaper (2016). Last accessed 28 March, 2020. https://nxtwiki.org/wiki/Whitepaper:Nxt
11. I. Bentov, A. Gabizon, A. Mizrahi, Cryptocurrencies without proof of work, in *Financial Cryptography and Data Security* (Springer, Heidelberg, 2016), pp. 142–157
12. A. Kiayias, A. Russell, B. David, R. Oliynykov, *Ouroboros: A Provably Secure Proofof-Stake Blockchain Protocol* (2016). https://eprint.iacr.org/2016/889.pdf
13. Z. Ren1, K. Cong, J. Pouwelse, Z. Erkin, *Implicit Consensus: Blockchain with Unbounded Throughput* (2017). arXiv:1705.11046v3 [cs.DC]
14. L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, A secure sharding protocol for open blockchains, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (ACM, New York, 2016), pp. 17–30
15. S. King, S. Nadal, *PPcoin: Peer-to-Peer Crypto-Currency with Proof of Stake* (2012). https://decred.org/research/king2012.pdf
16. P. Vasin, *Blackcoin's Proof-of-Stake v2, The BLK Community* (2014). https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf
17. T. Duong, L. Fan, H.S. Zhou. *2-hop Blockchain: Combining via Proof-of-work and Proof-of-Stake Securely* (2016). https://eprint.iacr.org/2016/716.pdf
18. A. Chepurnoy, T. Duong, L. Fan, H.S. Zhou, *TwinsCoin: A Cryptocurrency via Proof of-work and Proof-of-stake* (2017). https://eprint.iacr.org/2017/232.pdf
19. L. Ren, *Proof of Stake Velocity: Building the Social Currency of the Digital Age* (2014). https://www.reddcoin.com/papers/PoSV.pdf
20. L. Bentov, C. Lee, A. Mizrahi, M. Rosenfeld, Proof of Activity: extending bitcoin's proof of work via proof of stake. ACM SIGMETRICS Perform. Eval. Rev. **42**(3), 34–37 (2014)
21. I. Bentov, A. Gabizon, A. Mizrahi.: Cryptocurrencies without proof of work, in *Financial Cryptography and Data Security* (Springer, Heidelberg, 2016), pp. 142–157
22. *Proof of Burn*. Last accessed 22 March, 2020. https://en.bitcoin.it/wiki/Proofofburn
23. *Slimcoin* Last accessed 22 March, 2020. http://slimco.in/
24. J. Zou, B. Ye, L. Qu, Y. Wang, M.A. Orgun, L. Li, Proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services. IEEE Trans. Serv. Comput. **12**(3), 429–445 (2018)
25. K. Li, H. Li, H. Hou, K. Li, Y. Chen, Proof of vote: a high performance consensus protocol based on vote mechanism and consortium blockchain, in *Proceedings of the 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (IEEE, New York, 2017), pp. 466–473
26. D. Schwartz, N. Youngs, A. Britto et al. The ripple protocol consensus algorithm, in *Ripple Labs Inc White Paper*, vol. 5 (2014)
27. M. Castro, B. Liskov, Practical Byzantine fault tolerance, in *Proceedings of the Third Symposium on Operating Systems Design and Implementation* (1999), pp. 173–186
28. A distributed network for Smart Economy, in *Neo, White Paper* (2019). https://docs.neo.org/docs/en-us/basic/whitepaper.html

29. N. Chaudhry, M.M.Y. Punjab, Consensus algorithms in blockchain: comparative analysis, challenges and opportunities, in *International Conference on Open Source Systems and Technologies (ICOSST)* (2018)
30. A. Khanna, R. Anand, IoT based smart parking system, in *Proceedings of the 2016 International Conference on Internet of Things and Applications (IOTA)* (2016)
31. S. Purri et al. Specialization of IoT applications in health care industries, in *Proceedings of the 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)* (2017)
32. A. Khanna, R. Tomar, IoT based interactive shopping ecosystem. *Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT)* (2016)
33. A. Khanna et al. Intelligent mobile edge computing: a deep learning based approach. Commun. Comput. Inf. Sci. **1244**, 107–116 (2020)
34. https://www.idc.com/getdoc.jsp?containerId=prUS44898819. Last accessed 30 May, 2020