UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

Study of consensus protocols and improvement of the Federated Byzantine Agreement (FBA) algorithm

A Master's Thesis Submitted to the Faculty of the Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona Universitat Politècnica de Catalunya by

ASPASIA ZOI

In partial fulfilment of the requirements for the degree of MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisor: LEÓN ABARCA, OLGA and ESPARZA MARTIN, OSCAR

Barcelona, October 2019

Title of the thesis: Study of consensus protocols and improvement of the Federated Byzantine Agreement (FBA) algorithm

Author: ASPASIA ZOI

Advisor: LEÓN ABARCA, OLGA and ESPARZA MARTIN, OSCAR

Abstract

At a present time, it has been proven that blockchain technology has influenced to a great extent the way of human interaction in a digital world. The operation of the blockchain systems allows the peers to implement digital transactions in a Peer to Peer (P2P) network in a direct way without the need of third parties. Each blockchain determines different rules for the record of the transactions in the ledger. The transactions are inserted in blocks and each one, in turn, is appended to the chain (ledger) based on different consensus algorithms. Once blocks have been inserted in the chain, the consensus has been reached and the blocks with corresponding transactions are considered immutable.

This thesis analyses the main features of the blockchain and how the consensus can be achieved through the different kinds of consensus algorithms. In addition, a detailed reference for Stellar and Federated Byzantine Agreement (FBA) consensus protocols is made in order to explain these algorithms, their limitations as well as their improvement. The development of a reputation mechanism is necessary to the improvement of above algorithms.

Revision history and approval record

Revision	Date	Purpose
0	10/3/2019	Creation
1	6/9/2019	Revision
2	16/10/2019	Correction
3		

Written by:

Date	15/9/2019
Name	Aspasia Zoi
Position	Project Author

Reviewed and approved by:

Date	
Name	
Position	Project Supervisor

Reviewed and approved by:

Date	
Name	
Position	Project Supervisor

Contents

List of Figures

List of Tables

1 Overview

The rapid development of information technology and data digitization has contributed to the creation of a new digital cryptocurrency. The idea of creating digital cryptocurrency based on innovative Blockchain technology was that of Bitcoin and it was proposed by Satoshi Nakamoto in 2008 [17]. By the passage of the time, blockchain is also used by other fields besides cryptocurrency like financial, healthcare, supply chain etc.

Blockchain is mainly a distributed ledger that is completely open to everyone in public networks. Each participant can communicate directly to any other in the network as there is no central authority in the P2P network. When a transaction happens, the sender of the transaction signs this transaction using its private key and sends it to the network. Then, some other peers according to the rules determined by the network can validate this transaction verifying the integrity of this by checking the signature of the sender using its public key. Once the transaction has been validated is inserted to the block and after the specific number of transactions in the block or after a specific time according to the different types of blockchain, the block is appended to the chain. In addition, the ledger is stored in a number of peers in the network according to the different kinds of blockchain regulation and the rest of the peers can request it in order to have a copy of this ledger. When the transactions have been recorded in the blockchain, it is impossible to alter that transactions and can be considered immutable. The blocks in the chain are considered immutable as each block except from the transactions contains the hash of the block and the hash of the previous block. In addition, this hash is unique for each block and in case of changing at least one transaction or for example just one letter, the whole hash will be changed.

Blockchains build trust across a business network through the combination of a distributed ledger, smart contracts, and consensus. The term smart contracts was firstly used by Nick Szabo in 1997 [28], a long time before Bitcoin was created. They refer to computer programs which are intended to reach to an agreement through credible transactions without the presence of third-parties. Each party must trust the other party that will fulfil the requirements of the contract. Smart contracts have the same kind of agreement, but removes the need for trust between the various parties. This is due to the fact that a smart contract is a piece of code that is stored on a computer without someone to violate and is able to perform or to impose a predetermined agreement using a network Blockchain when and if concrete achievements are met. Moreover, the contract is automatically activated when it complies certain conditions which are agreed by the parties involved.

According to the consensus protocol, each P2P network based on Blockchain can work properly only when all the participants or the majority of them in the network agree on a specific block in order to be inserted in the chain. Through that procedure, the participants can validate and verify the transactions in order to attach them to the ledger. The process of data validation in a P2P network requires the use of complex algorithms such as Proof-of-Work [17], Proof-of-State [20] [19] and Practical Byzantine Fault Tolerance (PBFT) [22].

The next chapter will determine the operation of different kinds of blockchain based on different consensus protocols.

1.1 Thesis organization

The main purpose of this thesis is the improvement of the Federated Byzantine Agreement (FBA) consensus algorithm [15]. Although FBA is based on a distributed ledger, it has been proven that FBA behaves like a centralized network. The aim of the thesis is to lead the network to be more decentralized though the implementation of my proposed algorithm. Before implementing this algorithm, a fully examination of some consensus algorithms was necessary in a cooperation with my colleague Gavriel Christofi in order to compare and improve the limitations of the FBA algorithm.

This thesis consists of five chapters where the second and the third chapter is a state of the art in blockchain and were implemented in a cooperation with Gavriel Christofi. An overview of each chapter will be explained in the following:

- **Chapter 1:** This chapter makes a brief reference of what is blockchain, when it was appeared and how it works.
- **Chapter 2:** This chapter, in a cooperation with Gavriel Christofi, analyses the different types of blockchains and the main techniques that a blockchain uses like hash, digital signatures and Merkle root.
- **Chapter 3:** This chapter is, also, in a cooperation with Gavriel Christofi, where a further explanation of what is consensus is made. In addition, it presents different kinds of consensus algorithms and how they operate in their blockchains.
- **Chapter 4:** This chapter illustrates our proposal in order to eliminate the centralization appeared in Stellar and FBA consensus algorithms. So, a reputation mechanism is developed in order to make these algorithms more decentralized.
- **Chapter 5:** This chapter refers to the final conclusion of this thesis and future work to be done.

2 Blockchain

2.1 Types of networks

When Satoshi Nakamoto [17] referred to the blockchain and the decentralised systems, a new era of different types of networks was born. Until 2008, the most typical type of network was the centralised network in which all the decisions were taken and the procedures in the system were controlled by a central authority. In order to implement a financial transaction between client A and B, a client A had to contact a bank to make the desired transaction by waiting some days for the transaction process paying some fees.

Once the blockchain was invented, users have more active role on the network as they can make transactions directly without the need of third parties or they can notice which actions are taking place in the network. This can be achieved through distributed and decentralized networks. These three different types of network have many usages but in this thesis only the use of the blockchain is going to be analyzed. However, a trustful dynamic network requires some specific rules that users should follow. The transaction should be immutable. This can be achieved through consensus algorithms, which are not necessary in centralized networks.

2.1.1 Centralized

In this type of network, all the peers are connected to a central authority. For example, all the clients communicate through a single server. The central authority is responsible for determining its own rules that the rest of the network should conform to. Users have to send a request to the central authority in order to implement a transaction and after that they have to wait for the response to proceed.

The main advantages of the centralized network is that the decision making is taken in a fast manner as only the central authority is responsible for this action. In addition, the participation of users in the network does not require expensive and specialized equipment as the only duty of users is to execute transactions. Furthermore, in the centralized network, the information of the users such as identity and address is kept secure as any operation is implemented through the central server and not directly between the users.

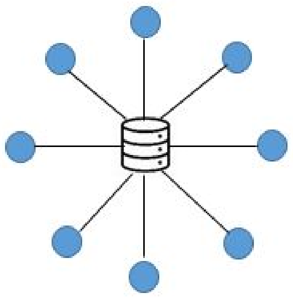


Figure 2.1: Centralized network

However, the centralized network suffers from a single point of failure. In case of any failure of the central server, the network goes down and then users cannot implement any request. In addition, another drawback of this type of network is lack of scalability: the case of high traffic could lead to the bottleneck where transactions would be implemented in a slow manner. Based on the high traffic issue, the centralized network is vulnerable to Denial of Service (DoS) and Distributed Denial of Service attack (DDoS) attacks where the attackers send a massive amount of data to the central authority in order to disrupt it to implement the proper procedures.

2.1.2 Decentralized

This network is the evolution of centralized network as eliminates some disadvantages, mainly the single point of failure. The network is controlled by multiple central nodes where users are connected to them. In addition, all the information is stored in all the central nodes unlike the centralized network where the data is stored only in a central authority. Furthermore, the peers can communicate directly each other without the need of third parties presence.

In case of failure of a central node, the network will keep operating as there are other central nodes to which users can connect in order to implement their transactions. In decentralized networks, the information is stored in multiple users and so it is difficult for any attacker to change this information in all the users. Furthermore the identities and the addresses of the users are difficult to be detected from a malicious node. It is impossible to track a specific user as the data is conveyed through different central nodes. Another advantage of this type of network is the absence of bottleneck as multiple nodes are responsible for different implementations.

On the other hand, users need to buy expensive equipment in order to act as central nodes for the validation of the transactions. Furthermore, central nodes can be selfish and can take decisions that act on their own interests and not on the whole network. Finally, the bigger the network, the more difficult could be to reach an agreement and, as a result, the speed for all the procedures in the network slow down.

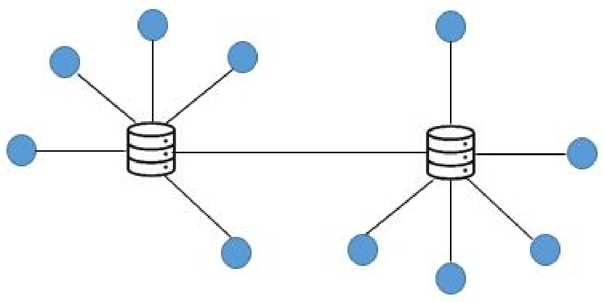


Figure 2.2: Decentralized network

2.1.3 Distributed

The distributed network consists of nodes that are all connected to each other and the communication is accomplished without the presence of a central node. All the participants in the network can have an active role in the network. For example, one percentage of the nodes in the network are responsible to store all the data that happens in the network and the rest of the peers can get a copy of the recorded data by requesting. Another percentage is responsible for controlling if everyone conforms to the network rules. If a part of the network collapses, the network will continue to operate normally, since communication can be done through the rest of the nodes.

The main advantage of a distributed network is the low latency due to the absence of central authorities and the geographically spread of the network. On the other hand, the main disadvantage of this network is the fact that is difficult to reach consensus because it is highly dependent on the size of the network. More peers in network, more complicated is.

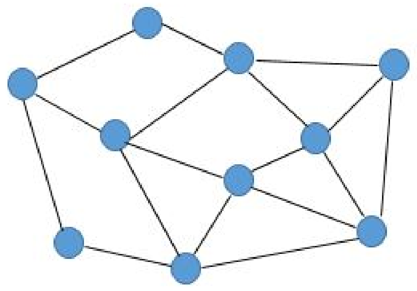


Figure 2.3: Distributed network

2.2 Blockchain

The term blockchain refers to a chain of blocks which was firstly described in 1991 by researchers, Stuart Haber and W. Scott Stornetta in order to timestamp a Digital document without being tampered [13]. In addition, in 1992, those researchers in collaboration with Dave Bayer improved the efficiency of this procedure by introducing the Merkle root, where the integrity of the data is checked in a efficient way [26]. In 2008, Satoshi Nakamoto published the first paper introducing the term chain of blocks describing an electronic cash system between peer to peer [17]. Through this paper, a new era of cryptocurrency was born and especially that of BITCOIN [17]. With the passage of the years, the term “chain of blocks” changed to what we call today blockchain.

Blockchain is a decentralized and distributed ledger across the network that can maintain and record transactions between peers unchangeably and securely. Each participant in the network can communicate directly each other without the need of the presence of the central authority. The P2P network determine some certain rules where all the peers who belong to this network should conform to. When the transactions are implemented between peers, every participant has to check the validity of these transactions according to the specific rules and then, these transactions are stored into a block. The blocks are linked to each other making a chain. Each block contains the hash of the previous block maintaining the transactions unharmed and making in this way the block immutable. So, the chain becomes more secure as the insertion of a new block verifies all the previous blocks. For example, if an attacker wishes to tamper the contents of the block, not only should she modify the data in that block but also the data of all the subsequent blocks [29]. The modification of the block is difficult to be implemented and it will be explained in more detail in the next chapter.

According to the operation of the blockchain, this technology makes use of cryptography and hash functions in order to protect and keep the identity of each user unique in the network. Each peer in the network has a pair of keys, the public and the private key. When a peer wants to make a transaction should sign his own transaction with his private key. Then the rest of the participants have to verify this transaction by checking the signature of the sender using its public key [9]. This procedure eliminates the need of the central authority to control the whole network as peers can implement transactions through their keys. Furthermore, another reason of no need of centralized authority is that every participant in the network has a replica of the updated ledger which includes all the blocks of the chain with the corresponding transactions.

All the peers in the P2P network have the right to join or to leave the network at any time they wish. The issue of the public blockchain is that the participants do not know each other, and as a result peers cannot know whom they can trust in the network. This problem can be prevented with the consensus algorithms which require all the peers or the majority of them to agree to the network rules as to reach the finality of the blocks that are stored in the ledger.

Except for the distributed ledgers and the consensus, blockchain can assure, also, trust and security through the smart contracts. It is an application that was introduced in the second generation of blockchain and especially in Ethereum. Smart contract is executed on top of the blockchain, and it is a set of computer programs that include codes and agreements in the form of business logic, between two or more parties without the need of third parties. When specific criteria are met, then the smart contracts are executed automatically in order to obtain the output and this result is stored and replicated in all participants in the network to provide immutability.

Despite of all these advantages, there are some limitations in the blockchain, which does not make it as trustable as seems. Some of the most typical limitations of blockchain are:

- Scalability: due to the consensus mechanisms, each peer in the network has to validate the transaction as a result the bigger the network is, slower the performance of processing transaction will be.
- Anonymity and data privacy: although in the case of public ledgers, where the peers are anonymous, when the transaction is occurred it is recorded in the blocks where each participant of the network can know about this transaction. Through these records, anyone has the possibility to identify other users as to learn private information of these users.
- Security: there is a possibility of a "51% attack", where if more than a half number of the peers lies, these lies will become the truth and then can persuade the rest participants of the blockchain. As a result, it would be able to have the control of the blockchain in order to change or remove the transactions.

2.3 Types of Blockchain

The blockchain can be implemented in three different categories, the public or permissionless, the private or permissioned and the consortium or Federated blockchain. Each type of blockchain has different characteristics and when the producers or businesses want to create a new blockchain, they can choose the specific type according to their needs in attempts to implement their own services. Some of these characteristics are the number of users that can take part in the network as their rights of participation to the verification of the transactions or the insertion of the blocks in chain etc.

2.3.1 Public Blockchain

In the permissionless blockchain everyone who wishes to join the network can take easily part in that and has, also, the right to read and write to the ledger. When a user joins the network, he is updated by all the information existing in the network and he has the right to make and validate transactions as to insert the block in the ledger [6]. Furthermore, due to the anonymity of peers in the public network, the users do not trust each other and for this reason, the network uses an incentive mechanism making the users to participate in the network. When users verify and validate transactions and blocks, the network gives them rewards, in order to encourage more and more participants to join that.

The public networks usually are more secure than the other types of blockchain as there is no presence of central authority and peers can keep their anonymity. In addition, in case that some nodes fail, this does not affect the normal operation of the network as each node in the network is fully aware of the transaction in the ledger. Furthermore, the security can be achieved through the consensus algorithms such as proof of work (PoW) and proof of stake (PoS) [29]. Through these algorithms, users have to prove that they have spent an amount of energy (PoW) or hold an amount of money (PoS) in order to be active members in the consensus process. Bitcoin and Ethereum are examples of this type of the blockchain.

One of the main limitations of public blockchain is that it needs a huge amount of resources like computational power and an expensive equipment in case of proof of work in order to ensure consensus in the network updating all the nodes. According to the previous case usually the transactions fees in the permissionless network tend to be high. In addition, each node has to possess a huge space of storage in order to record the transactions. For example, in Bitcoin, the size of the blockchain file is at about 217.8 GB so far [3]. The verification of transactions as the validation of the blocks is too slow on the grounds that each one has to be validated by the whole network. Finally, public networks can suffer from what is known 51% attack where a group of miners manipulate the majority of computing power on the network and thus they are able to choose what transactions can be valid regarding to their own perspective.

2.3.2 Private Blockchain

Unlike the public blockchain where everyone can join the network, in private network, the network administrators give the permission to a specific number of users to participate in the network. This kind of network consists of a small group of participants who are known each other. The network usually does not require any consensus algorithm due to the fact that all nodes are trusted. In case of malicious behavior, the peers become aware of this situation and they can solve the problem following the certain rules where network has established. In this permissioned network, every peer has different participation rights than the permissionless network where all the participants have the same rights. In this case, a small number of participants is responsible for the validation of the transactions and the insertion of the blocks into the chain. In addition, the ledger is not necessary to be replicated to the entire network. For example, a business in order to make an agreement, can communicate only with peers who are associated with this agreement and once the deal is carried out, then the whole network can be informed with this agreement if the business wishes.

Private networks present better performance than the public networks as the transactions are stored and validated only by a specific number of peers and not by the whole network. Furthermore, the cost of the transaction fees is cheaper as there is no need for huge computational power to reach finality. As described previously, when a peer behaves badly, the network can detect immediately this abnormal behavior, it fixes it and uses consensus protocols and it reaches finality much quicker unlike public network where achieving consensus could take time.

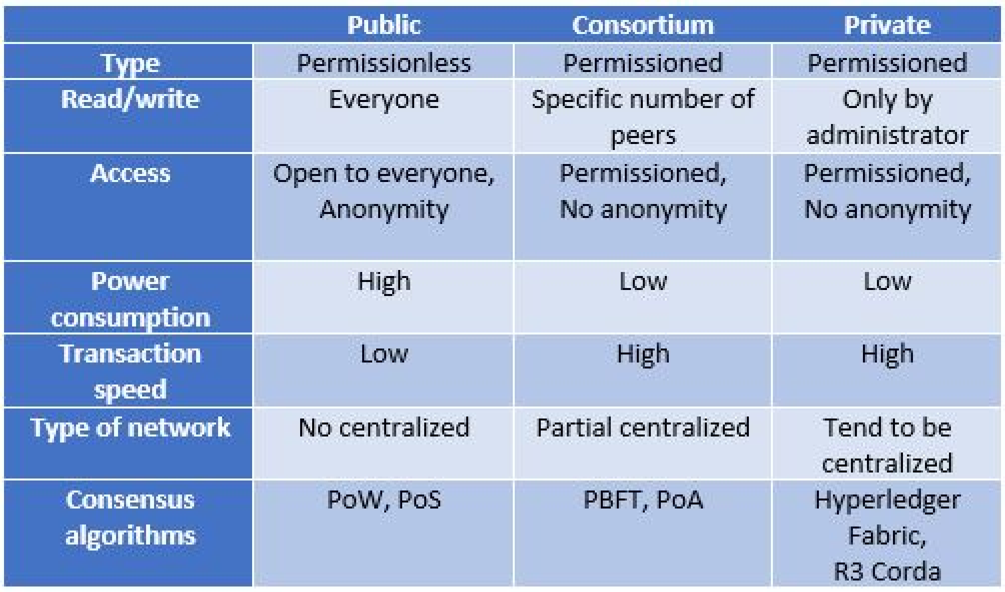
On the other hand permissioned blockchain have security issues. Due to the reason that the network consists of a limited number of peers, it gets easier for the attacker to take control of the entire network. In addition, this type of network tends to be centralized as only some peers are responsible for the normal operation of the network. Finally, the participants of the network are controlled from the business or the organization which are the owners of the network. This means that as the information is not published in the whole network, they can control or modify the transactions in their own perspectives. Hyperledger Fabric and R3 Corda are some platforms that uses private networks.

2.3.3 Consortium Blockchain

The consortium blockchain is permissioned and semi decentralized due to the small number of the participants and the hierarchy of the peers. Instead of public network where everyone can have access and participation for the consensus to the network, this type of blockchain relies on the predetermination of selected peers for the validation of the transaction and the insertion of the block in the ledger. For instance, if a network consist of 30 members, only the 2/3 of the peers (20) have to sign in one block in order to be valid. Furthermore, in this blockchain each participant knows every peer in the network having different rights from each other like in permissioned network.

Consortium blockchains, more or less, present the same advantages and disadvantages as the permissioned blockchains. It is not based on PoW like in case of public, but it is adapted from Practical Byzantine Fault Tolerance (PBFT), Proof of Activity (PoA) and other consensus algorithms as the peers trust each other. In addition, MultiChain and Hyperledger platforms are highly based on this type of networks.

The table 2.1 summarizes the main characteristics of these three different types of blockchain.



	Public	Consortium	Private
Type	Permissionless	Permissioned	Permissioned
Read/write	Everyone	Specific number of peers	Only by administrator
Access	Open to everyone, Anonymity	Permissioned, No anonymity	Permissioned, No anonymity
Power consumption	High	Low	Low
Transaction speed	Low	High	High
Type of network	No centralized	Partial centralized	Tend to be centralized
Consensus algorithms	PoW, PoS	PBFT, PoA	Hyperledger Fabric, R3 Corda

Table 2.1: Summary of the three types of blockchain

2.4 Architecture of Blockchain

The main purpose of blockchain is to eliminate the central authority. In order to achieve this, blockchain had to invent new features to provide security in the network. The following subsection will analyse the main characteristics of the blockchain.

2.4.1 Hash

One of these features is the use of the Hash function which keep the identity of each user and the contents of each block in the chain unique. Specifically, the hash function is similar to identification of the person through its fingerprint. For example, not only does a transaction include the asset will be sent but also the address of the sender and the receiver. Through the hash function, the network can prevent the peers from imitating other participants in the network as it can confirm that a transaction originated from the real sender.

The hash function is a mathematical algorithm which converts an input of a variable size into an output of a fixed size. For example, Bitcoin uses the Secure Hash Algorithm SHA-256 where the input can be 2^{64} bits and the output of this algorithm is a fixed size of 32 Byte [29]. This function is characterized as not irreversible and the result of this algorithm represents a unique digital fingerprint which cannot be invertible. Furthermore, a slight difference in the input data leads to a new hash creation and as it is depicted in the figure 2.4, the deletion of only one exclamation mark changes completely the obtained output.

In blockchain, hash plays a significant role to keep the block secure and immutable in the chain as each blockheader of the block includes the hash of the previous block. As a result, it's impossible for an attacker to modify the contents of a block because not only must he change the single block but also all the subsequent blocks in the chain.

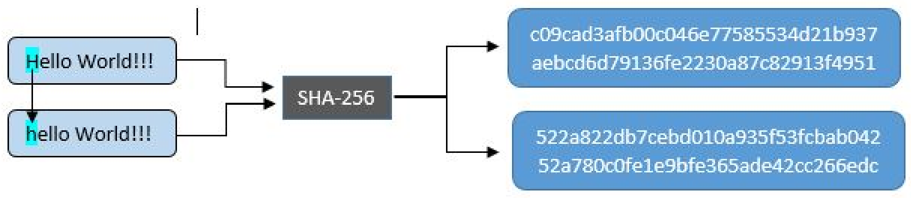

Hello World!!!
hello World!!!
SHA-256
c09cad3afb00c046e77585534d21b937 aebcd6d79136fe2230a87c82913f4951
522a822db7cebd010a935f53fcbab042 52a780c0fe1e9bfe365ade42cc266edc

Figure 2.4: Example of hash function

2.4.2 Public and Private Keys

Every peer in the blockchain network owns a pair of asymmetric keys, the public and the private key. These keys play a significant role not only on the creation of the transactions but also the validity of these transactions. The private key is secret and is only known by each user. This key is used to digitally sign the transactions in order to prove the authenticity. The public key is visible by everyone in the network as it represents the address of each user by hashing this key. In addition, the public key is used to verify if the specific transaction was executed by the real sender. Furthermore, hashing the public key gives an extra privacy to the user.

The picture 2.5 depicts the whole procedure needed for the implementation of the transaction. The digital signature consists of two phases[1]: the signing phase where is generated by using the private key and the second one is the verification phase where anyone can verify the digital signature.

Alice wants to execute a transaction with Bob. In order to make this transaction, she has to follow some steps.

- Alice, firstly, has to create a hash of all the data that correspond to this transaction.
- In addition, she creates a digital signature by encrypting the hash which was obtained by the data of this transaction with her private key.
- Finally, she sends the digital signature as the data of the transaction to Bob. Then, Bob can verify if Alice is the real sender of this transaction.

On the other hand, once Bob has already received the data of the transaction and the digital signature of Alice, he has to implement two procedures to crosscheck the validity of Alice's transaction.

- Bob has to decrypt Alice's digital signature by using her public key in order to get the hash.
- Secondly, Bob has to use the same hash algorithm as Alice in order to calculate the hash of the data.

If the results of the hash coincide then it is clear that the verification was implemented successfully without tampering the data and Alice is the real sender of the transaction.

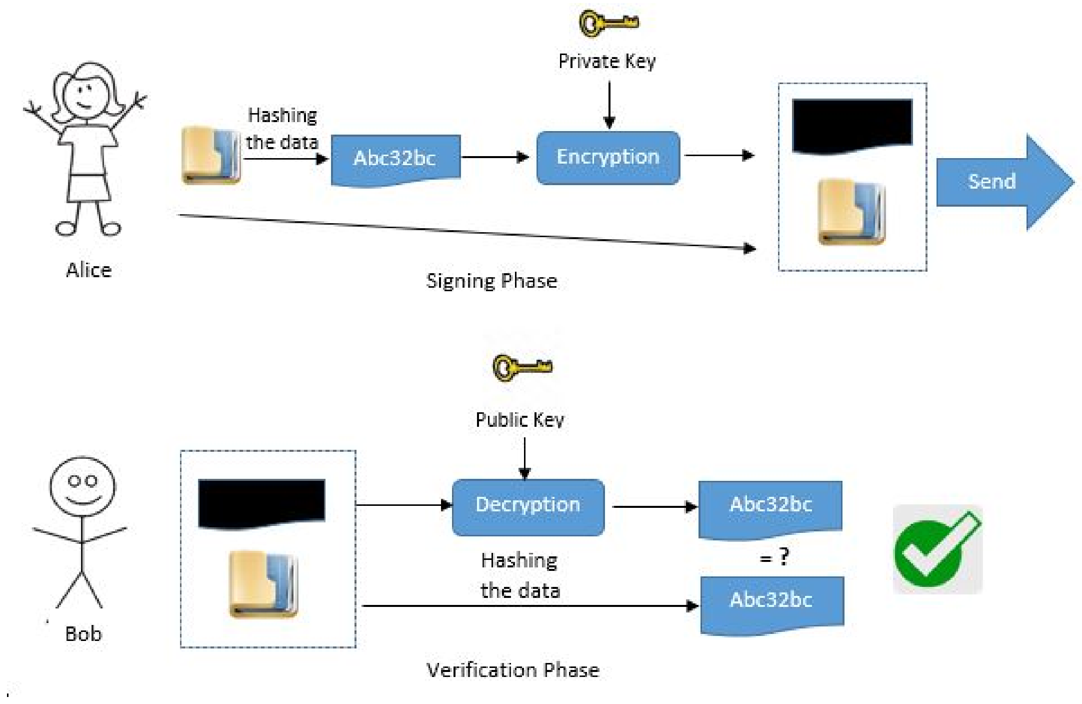


Figure 2.5: Implementation of transaction

2.4.3 Merkle root

In all types of blockchain and especially, in that of public network, the number of users that can take part in the network could be very high from thousands to millions of users as a result, the bigger the network is, more transactions are executed in the network. In order to be implemented, these transactions are stored firstly in the blocks before appending to the chain. Hundreds to thousands of transactions can be included in every block. Once a block has been inserted to the chain, then each user is informed about this situation and stores a full replica of the ledger. However, that operation cannot be implemented normally when the users have a limited capacity in their devices such as mobile phones and laptops, as there is no enough space to store the updated replica of the ledger. For this reason, the inventor of Bitcoin, Satoshi Nakamoto, was the first that made use of Merkle tree root in blockchain in attempts to resolve the issue of the storage space. This technique was invented by Ralph Merkle in 1979.

Merkle root is included in the blockheader of each block which belongs to the chain and represents a single hash value of the whole transactions corresponding to a specific block. Especially, Merkle root works as a binary tree which consists of hashes. An example of Merkle root is depicted in figure 2.6 supposing that a certain block consists of four transactions Tx1, Tx2, Tx3 and Tx4. These transactions are located at the bottom of the tree where a hash of each transaction is calculated separately hTx1, hTx2, hTx3 and hTx4 through a cryptographic hash function. Then, the hashes of these transactions known also as children are grouped in pair of two creating two new hashes hTx1Tx2 and hTx3Tx4 which are called parents. Especially, the concatenation of hTx1 and hTx2 is hashed and produces the hTx1Tx2. These hashes represent the parents and the leaves. This procedure is continued until reaching the final single hash known as Merkle root. As referred previously, Bitcoin makes use of SHA-256 cryptographic hash function and in the case of Merkle root, this function is used twice in order to obtain the hash result [1]. For instance, hTx1 = hash(hash(Tx1). Furthermore, the number of transactions included in each block should be even as the following example with the four transactions. However, in case of existing an odd number of transactions in the block i.e. 5 transactions, then the fifth transaction has to be paired with itself to create an even number of transactions and subsequently this pair is hashed creating a new parent hTx5Tx5.

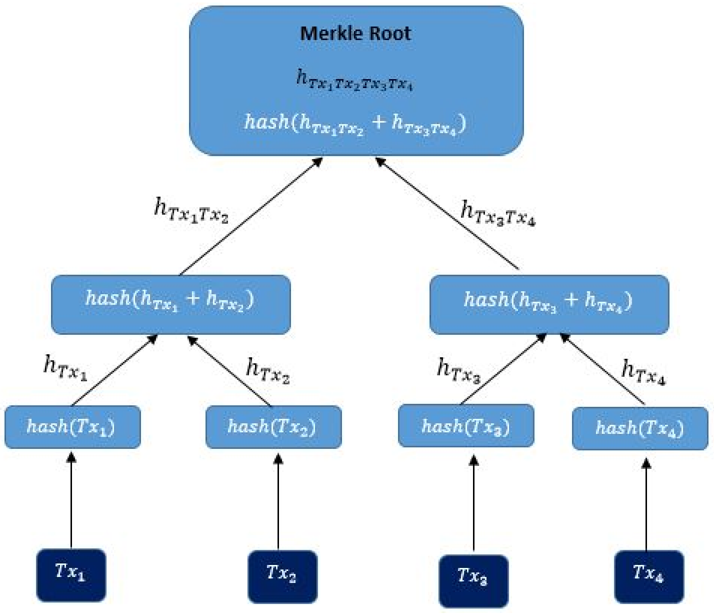


Figure 2.6: Structure of Merkle tree

On the other hand, not only was Merkle root used to overcome the storage issue, but also to crosscheck if a specific transaction or a number of transactions belongs to a certain block. This can be achieved by checking only some intermediate hashes of the Merkle tree. Given these hashes known as Merkle path, the verifier should calculate a single hash value and then he has to compare it with root hash. If both hashes are the same, then the transaction belongs to this Merkle root or it has not been tampered with. Let's assume that Bob receives the root hash hTx1Tx2Tx3Tx4 from a trusted source and Alice wants to prove to Bob that the Tx1 transaction exists in this specific root hash and the data is not tampered. So, Alice has to send the Tx1 transaction, the hTx2 and the hTx3Tx4 to Bob. He, in turn, should calculate the h'Tx1, the h'Tx1Tx2 and h'Tx1Tx2Tx3Tx4. Finally, he has to compare if both root hashes are the same hTx1Tx2Tx3Tx4= h'Tx1Tx2Tx3Tx4. In case of Bitcoin, where more than a thousand of transactions can be included into 1 Mb block, users have to produce only log2N hashes (from 10 to 12 hashes) to verify a single transaction. N is the number of the whole transactions that exist in the block. As a result, the computational effort can be reduced through the Merkle tree.

The Merkle root is a useful tool, especially for the users that they do not have available large space storage. In blockchain, the users can be divided in two categories: full nodes and light clients [27].

- Full nodes are responsible for downloading a replica of the ledger which consists of the genesis block until the last appended block in the chain as to verify the authenticity of the transactions which are recorded in the blockchain. Users are served as a full node need a huge amount of storage in order to store the updated replica. Nowadays, the size of the blockchain in Bitcoin has reached almost 250.6 GB [4] and this size is going to be increased as each block is generated every 10 minutes.
- Light clients, in contrast of full nodes, download only the blockheader of each block

which corresponds to 80 Bytes per block cite3. As a result, users with a limited capacity can have an active role in the network as light clients. They make use of the Simplified Payment Verification (SPV) [1][27]in order to check the authenticity of the transaction. When a user wants to verify a transaction, it makes use of Bloom filter (Bitcoin) in order to find only the associated transaction by asking for a full node. Once the full node has detected the desired transaction that corresponds to the Bloom filter, then the node informs the light client by sending to him the blockheader of the corresponding block as the Merkle path.

2.4.4 Structure of blocks

As mentioned previously, blockchain is a chain which consists of many blocks. Each block depends on the previous block as each one includes the hash of the previous block. Furthermore, such a feature makes a blockchain more secure because it is very difficult to modify the block without changing the subsequent blocks. As a result, the blocks inserted to the chain are immutable and neither can be deleted nor tampered with. The blocks differ from blockchain to blockchain and constitute the records of the transactions as the blockheader which contains some specific data.

When the transactions occur, every validator has a right to pick a number of transactions, verify them and insert them into a block. The number of transactions that can be stored in a block depends highly on the size of the block and the transaction. For example, the number of transaction in Bitcoin is limited as the size of the block is only 1 MB. In addition, validators can usually choose which transaction wishes based on the fees of each transaction. In Bitcoin, the transaction with the highest fees submitted by the producers of each transaction are selected quicker than those transactions with low fees.

Besides the transaction list which is included in the block, the blockheader is, also, part in that[1][30]. Blockheader consists of six fields:

- Version: it indicates the version number of upgrades and changes in the protocol used by a validator.
- Timestamp: it indicates the time where a block is generated.
- Hash of the previous block: this hash points to the previous block. Especially, all the data that are included in the block are inserted in a cryptographic hash function in order to obtain a single hash value. This hashed result is stored in the next block in the blockchain.
- Merkle root: as refereed to earlier, it is a single value of hash which represents the summary of all the transaction included in the block. Firstly, once the transactions have been hashed, they are grouped in a pair of two producing a new hash. This operation is continued until only one single hash is fount.
- Nonce: it is used in Proof of Work (PoW) algorithm by the miners and it is the only field in the block that can be changed. Modifying the nonce, miners have to solve the difficult mathematical puzzle in attempts to prove that the hash of the block header is lower or equal to the difficult target.
- Difficult target: it is a target threshold which is determined by the network every two weeks in order to follow the rules of Bitcoin where a new block is generated every

10 minutes. For instance, if the time that the blocks are inserted in the blockchain is less than 10 minutes per block, then the difficulty target should increase.

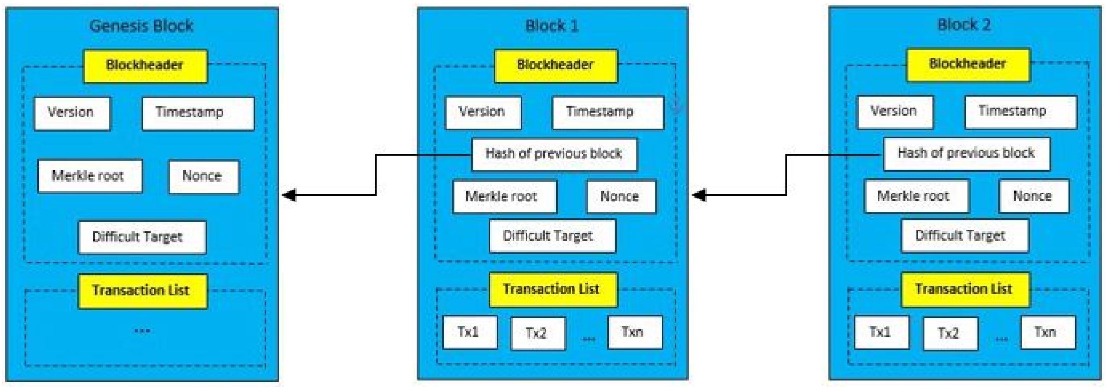


Figure 2.7: Structure of blocks

According to the sequence of the blocks, the block which is generated firstly in the chain is known as genesis block and is, also, referred to as block zero. Each block in the blockchain can be identified through a block height number which is always a positive integer number and increases by one every time that a block is inserted to the blockchain. For example, the second block which is inserted by the genesis block is referred to as block 1 and so on. A blockchain is depicted in the above figure 2.7 which consists of the genesis block and another two blocks. Furthermore, it illustrates how the block are linked to each other as the contents of them.

3 Consensus

3.1 What is Consensus?

In a centralized network, the central authority is responsible for the validation of the transactions providing security and trust to the entire network. However, in a distributed and a decentralized network, the peers interact only each other as there is not any central authority. In this case, for better maintenance of the network, all the peers have to follow a set of rules determined by the network and come to an agreement over a single block which will be inserted in the chain. This agreement in the blockchain is achieved through consensus algorithms. Nowadays, there are a lot of consensus algorithms where each one is used by a different type of blockchain. For example, in permissionless networks, the most common consensus protocols are Proof of Work (PoW) and Proof of Stake (PoS) while the permissioned networks make use of practical Byzantine Fault Tolerance (PBFT) etc.

The main purpose of consensus algorithms is to solve the problem that is created when any node in the network try to reach a consensus either in the presence of malicious behaviour of the peers or any faulty process exists in the network. These issues are known as forking and Byzantine General Problem. Forking can happen when two peers propose simultaneously a new block or when a peer wants to behave maliciously creating the same block twice. In both cases, consensus mechanism can solve this problem by choosing the longest chain of the blockchain. Furthermore, the byzantine general problem occurs when there is not unanimity in the network between the peers agreeing for a single piece of data on the grounds that some of them can misbehave sending erroneous messages or being faulty due to the crash of the system etc. As a result agreement can not be reached.

As referred previously, the distributed network suffers from some issues such as an abnormal behaviour of the peer or the network. However, two main categories of consensus mechanisms have been developed in order to overcome these problems [18]:

- Proof-Based consensus algorithm: it is used in permissionless networks where every peer in the network can be a validator to verify the block and insert it in the chain giving the sufficient proof to the network that a validator deserves to participate in. For instance, Bitcoin uses the PoW consensus algorithm and a validator (miner) has to consume a huge amount of energy in order to solve the mathematical puzzle and append the block to the chain.
- Vote-Based consensus algorithm: this type of algorithm is used in permissioned networks based on multiple rounds of votes in order to achieve consensus. For example, Hyperledger Sawtooth makes use of PBFT where a lot of messages are exchanged between the participants in the network in attempts to reach the consensus.

3.2 Byzantine General Problem

The ”Problem of the Byzantine generals” was originally described by Marshall Pease, Robert Shostak and Leslie Lamport in 1982 [14] and was based on the scenario that parts of the Byzantine army camped outside the walls of an enemy city are facing problems with the creation of a joint action plan due to the possibility of transmitting unreliable information. In detail, each part is administered by its own general and it has been decided that after a certain period of monitoring of the enemies, all the generals should agree on a joint action plan. So in order to achieve the necessary communication and the coordination of all generals, it is safeguards to transfer the necessary information and news from the one part of the army to another. At this point it should be noted that in the given historical period the transmission of information was made exclusively by messengers of each troop. However, this method of transmitting information poses risks of falsifying the information, because some informants who have betrayed the troop and allied with opponents, they have the ability to transmit false information in order to prevent (law-abiding) generals from coming to an agreement.

So, on the basis of the above, it is clear that the generals must use an algorithm which guarantees that:

- The action plan to be implemented is a unanimous decision by all the generals involved.
- The possible existence of a limited number of informants who have betrayed their troops will not lead to the manipulation of the data and therefore to the adoption of a harmful action plan.

Figure 3.1 illustrates two examples of Byzantine general problem in a presence of honest and dishonest soldiers. The example appeared in the left icon depicts the successful attack in the city as all the army following the general's command will attack at the same time. On the other hand, the second example represents the unsuccessful attack when there are some traitors trying to manipulate the order of the commander. In more detail, the general (Commander) gives an order of attack to the army but a part of the army are traitor and falsifies the order in retreat.

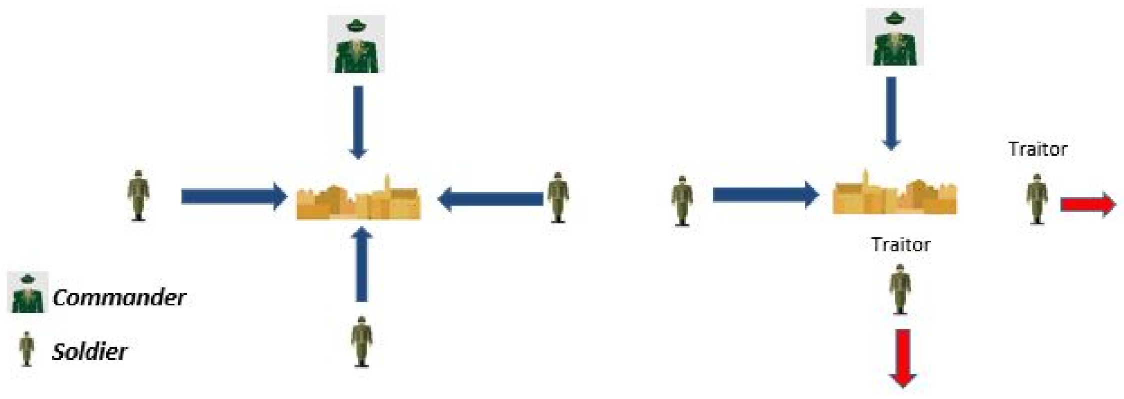


Figure 3.1: Coordinate and uncoordinated attack

It is therefore obvious that without the existence of synchronicity and integrity in the information exchanged, it is impossible to implement a unanimous and reliable action plan, because the risk of carrying forged orders from traitors informants always exists.

This method can correlate with the distributed system where generals represent the honest nodes and the generals who betray the troops represent the malicious nodes in the network. Lamport, Shaostak and Pease proved that if at least the 2/3 of the network acts honestly, then there will be a unanimous decision in the entire network. Especially, they determined a condition which indicates that if the number of malicious nodes is equal or exceeds the 1/3 of all the nodes, then the system will not work properly.

$$n \geq 3f+1$$

Where n is the total number of nodes in the network and f the number of malicious nodes.

The above condition can work properly if the whole network consists of at least four nodes. In the case of three nodes where one of those nodes behaves maliciously, agreement cannot be reached between of them as each node receives two different messages. As depicted in Figure 3.2, there is one commander and two generals. Commander and general 2 act honestly but general 1 is a traitor and wants to change the decision of the commander. Therefore, general 2 cannot make a decision as received both attack from commander and retreat from general 1. In normal case, general 2 should trust the commander as he posses higher position in hierarchy than general. However, what happens if commander behaves maliciously? This leads to the same scenario as previously, since commander sends attack to general 1 and retreat to general 2. So, General 1 does not know what decision can take.

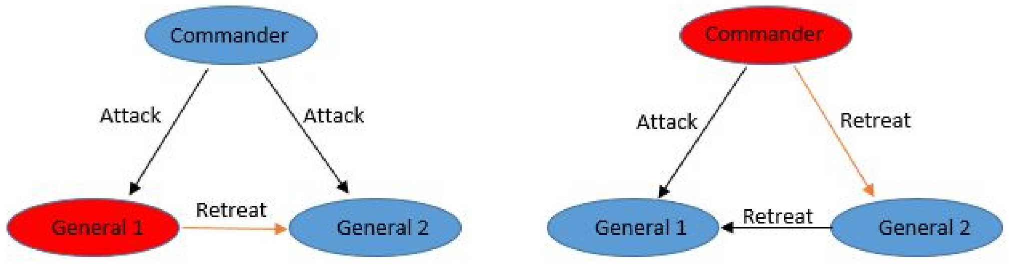


Figure 3.2: System with three nodes

In case of four nodes where one of them is a malicious node, generals can reach agreement as the 2/3 of the nodes in the system behaves honestly. Each general informs the other ones in the system about the command they have received. According to Figure 3.3, General 2 receives two attacks from the commander and the general 3 and one retreat from the general 1 acting as a traitor. As a result, general 2 takes into account the majority of the messages that he received and then he will attack. In addition, in case that commander sends erroneous messages, generals could be possible to reach in agreement, as two of the three generals will get the same messages. For example, all generals receive two attacks and one retreat.

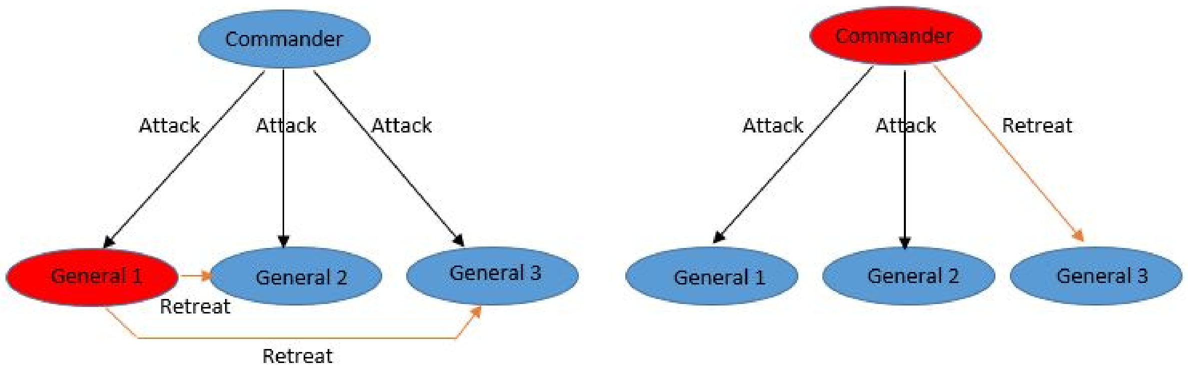


Figure 3.3: System with four nodes

3.3 Consensus Algorithms

3.3.1 Proof-Based consensus algorithm

3.3.1.1 Proof of Work (PoW)

The Proof-of-Work (PoW) is an algorithm in attempts to achieve consensus to ensure the validity of the transactions as the creation of blocks. Although, the concept of this algorithm was first developed in 1993 by Cynthia Dwork and Moni Noar trying to solve the problem of spam emails, the term ”Proof of Work” was first referred and formalized in a 1999 paper by Markus Jakobsson and Ari Juels. Later, this algorithm became widely known through the founder of Bitcoin, Satoshi Nakamoto, who used this mechanism to reach consensus successfully between many nodes on the network and he used it as a way to secure the Bitcoin Blockchain.

To understand the operation of the proof-of-work algorithm should be taken account that each time a transaction is executed in the Bitcoin network, it is recorded and is stored in a temporary block. Thus, once the block has been confirmed with all the validated transactions included in that block, then this block is inserted to the chain sending a replica of this record to the rest of the participants in the network. The validity of those transactions as the insertion of the next block in the chain are implemented successfully through the use of Proof-of-Work consensus protocol. The above process is called mining and is done by the miners who are trying to solve a difficult mathematical cryptographic puzzle in attempts to obtain the appropriate result determined by the difficult target. Once miner obtains that, it informs all the participants in the network about the result and then the participants verify this validity in an easy way reaching the consensus.

Especially, every time that a new block is generated, all the verified participants should include all the valid transaction into the block as the HASH of the previous block. In addition, in the new block, the block header consists of some other fields like the Merkle root, difficult target and the Nonce. All the fields included in the blockheader as the procedure of the PoW are depicted in the figure 3.4.

In Bitcoin, using an encryption algorithm called as SHA-256 hash, miners have to solve the difficult mathematical puzzle by modifying the nonce in order to prove that the hash of the block header is lower or equal to the difficult target. Always, the output of a

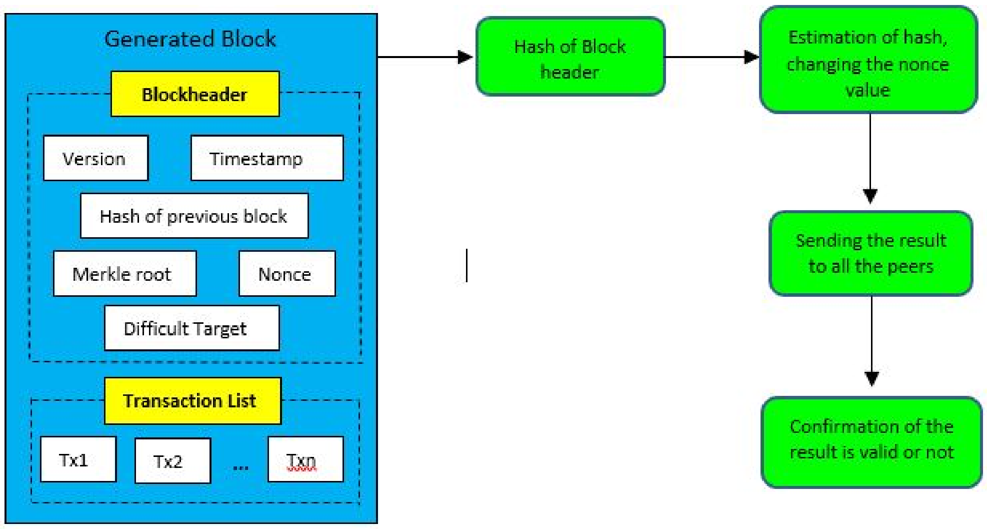


Figure 3.4: Procedure of PoW

SHA-256 hash will be 256 bits long regardless of the size of the input.

$$\text{hash}\ \big(\text{blockheader}\ +\text{nonce}\big) \leq \text{Difficult target}$$

For example, if the difficult target for the network is 0000 and the base string is "Hello, world!", then miners should find the desired result changing the value of nonce until the hash (Hello World! nonce) gives at least 0000. The rest bits of the hash does not affect the selection of the desired result.

Hash ("Hello, world!0") ⇒1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64

Hash ("Hello, world!1") ⇒e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8

Hash ("Hello, world!2") ⇒ae37343a357a8297591625e7134cbea22f5928be8ca2a32aa475cf05fd4266b7

...

Hash ("Hello, world!4249") ⇒c004190b822f1669cac8dc37e761cb73652e7832fb814565702245cf26ebb9e6

Hash ("Hello, world!4250") ⇒0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9

As shown in the example, the computer tried 4251 times to calculate the hash that gave the first 0000 bits. Furthermore, this obtained result is sent to all participants of the network and they, in turn, have to verify this result hashing the phrase with the corresponding nonce. If the validation is done successfully, then the block is verified and the miner can receive the rewards as a small amount of transaction fees. The miners, usually, choose the transactions with the highest fees in order to get included in the block. But the number of these transactions is limited because is fully depended on the block size which is 1 MB. The transactions fees are determined by the peers that execute the own transactions. More the fees that peers pay, faster the time to valid these transactions.

Especially, a reward should be given to the miner after the insertion of a given number of blocks. For example, in Bitcoin, a reward is given after at least 6 validated blocks appended to the chain making the block immutable. Nowadays, the rewards of miners are 12.5 bitcoins but during the 2016, the block reward was at 25 bitcoins. The amount of reward is halved every 210000 blocks and this happens in interval of four years. This procedure of halving rewards will keep going until the total amount of bitcoins reach the 21 million in 2140. Then, the generation of bitcoins will be stopped as a result, miners will not get the rewards but they will keep receiving the transaction fees.

Due to the different CPUs that exist worldwide, some miners can solve the cryptographic puzzle in less than 10 minutes or they want more than the expected time to have a result. For this reason, the network determines a difficult target every two weeks in order to follow the rules of Bitcoin where a new block is generated every 10 minutes. If the time that blocks are inserted in the blockchain is less than 10 minutes per block then the difficulty target should increase and vice versa. The value of the difficult target can be obtained from the following equation:

$$New\_Difficulty = Old\_Difficulty * \frac{Actual\ time\ of\ last\ 2016\ Blocks}{20160\ minutes}$$

The period of two weeks consists of 20160 minutes. During these two weeks, if every new block is generated every 10 minutes, then it will consist of 2016 blocks. In case that the blocks are generated every 9 minutes, then 2240 blocks will be generated in two weeks, 224 blocks more than the normal situation. As a result, the network should calculate only the actual time of the first 2016 blocks needed to calculate the new difficult target. Therefore, the time that corresponds to 2016 blocks are 18144 minutes and having the old difficult target and the actual time of last 2016 blocks, the network can calculate the new difficult target.

There is a case that two or more miners can find the same or different result where the hash of the block header is lower or equal to the difficult target by solving the cryptographic mathematical puzzle simultaneously. Each miner tries to solve its own cryptographic puzzle as each block includes different transactions.

In that case, the miners should broadcast the result to the rest of the participants of the network, which leads to the creation of the forking. As depicted in the following figure 3.5, fork can consist of more than two chains. In a normal situation, miners will choose in which chain they are going to mine because every time that a miner tries to mine a block consumes a huge amount of energy. They have the right to mine in more than one chain but this is needless due to the reason that the miner's reward will be smaller than the cost of energy he is going to consume. Furthermore, the fork will be stopped when one chain overtakes the other chains. So, the longest chain is chosen giving the reward to the miners belonging to this chain and the other ones are discarded without miners taking any reward.

The main goal of PoW is to prevent cyber-attacks such as Sybil attack and Distributed Denial of Service attack (DDoS). Sybil attack occurs where a single entity creates multiple fake identities in order to control the majority of the network. If the attackers control more than the half of the peers in the network they can execute a double spend attack implementing which transaction they want and preventing transactions from other users to be validated. In addition, they can modify validated transactions that have already inserted in the block. In PoW, an effective attack is costly and time consuming on the grounds that every device wastes a huge amount of energy in attempts to solve the mathematical puzzle and that makes the PoW algorithm more secure. For this reason, Sybil attack can not be implemented. The purpose of the DDoS attack is to delay the transaction valida-

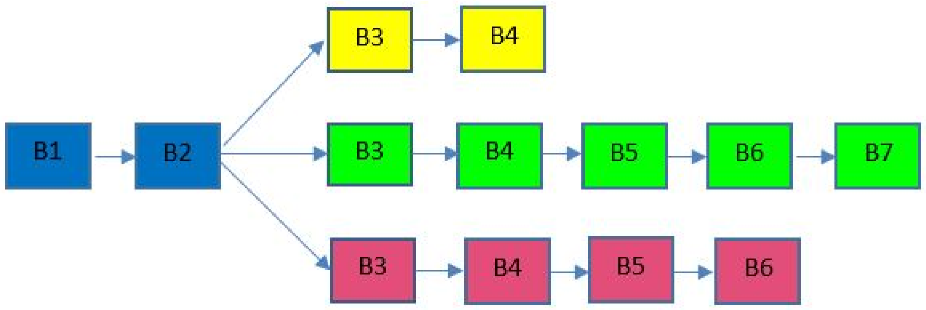


Figure 3.5: Case of fork

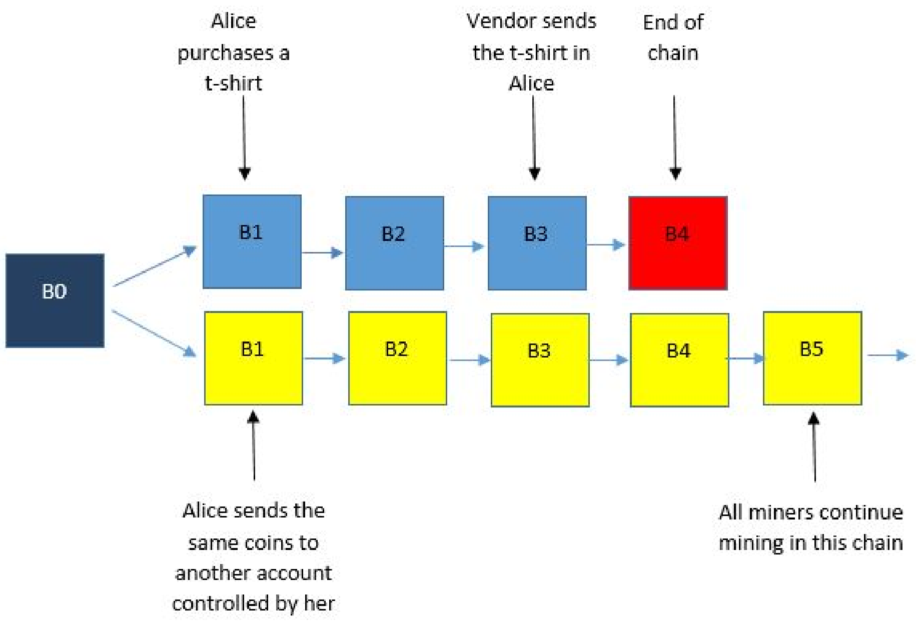
tion of the peers by sending a flooding of fake transactions in the network. The Bitcoin can validate up to 7 transactions per second, the size of each block is only 1MB and each block is inserted to the chain at about 10 minutes. As a result, only a specific number of transactions will be included in each block, some transactions will never be validated, and they will be discarded. Bitcoin, solved this problem by introducing transaction fees. Therefore, the miners include to their blocks the transaction with the highest fees.

The PoW suffers from what is known as 51% attack, where a group of miners manipulates the majority of computing power on the network and thus they are able to choose what transactions can be valid regarding to their own perspective. This can be easily achieved through the mining pools in which a lot of peers cooperate in order to find the solution of the cryptographic puzzle in a faster manner. They are given more chances to solve the cryptographic puzzle rather than other peers who are not included in the mining pool as a result they can control the network. In addition this cooperation leads to the centralization.

However, the presence of 51% attack leads usually to another issue that exists in the blockchain, called as double spent attack. Such an attack happens when a malicious peer wants to spend the same amount of coins on the blockchain twice. Every time that a transaction is executed, a vendor must wait n confirmations of blocks and then provide the product to the sender of the transaction. The n number of confirmations depends on the types of the payments, namely how fast or slow a transaction is implemented [12]. The slow payments are those that provide the most secure way of payments on the grounds that a vendor has to wait at least 6 confirmation in order to send the product and the time which is needed to the confirmation operation corresponds approximately to one hour.

On the other hand, the vendors do not need to wait 6 confirmations to make an exchange in fast payments. In this case, they usually do not wait for confirmations or just waiting for one or two confirmations. So, there is a possibility of a double spent attack where a malicious attacker creates a fork by putting its real transaction (a purchase of a product) in the block of the first branch and its fake transaction (the same amount of coins is sent to another account controlled by the attacker) in the block of the second one. The second branch, in essence, is produced by malicious peers and is not published to the entire network. This branch will be published when a malicious attacker receive the product and when this chain overtakes the real chain. As a result, all the transaction included in the real chain is discarded and the fake chain wins. According to the following figure 3.6, it illustrates the real chain in the light blue boxes where Alice wants to buy a t-shirt and the fake chain is represented by the yellow blocks in the second chain.

In addition, another drawback is that it has been proved costly and energy intensive



Alice purchases a t-shirt
Vendor sends the t-shirt in Alice
End of chain
B0
B1
B2
B3
B4
B1
B2
B3
B4
B5
Alice sends the same coins to another account controlled by her
All miners continue mining in this chain

Figure 3.6: Double spend attack

as miners require a huge amount of computation power and expensive equipment in order to execute the mathematical algorithms. Furthermore, PoW has some performance limitations. The first one is the low throughput as the average number of transactions that can be committed are 7 transactions per second, unlike Mastercard that executes 10000 transactions per second [2]. Furthermore, PoW suffers from the high latency since a block can be considered as valid after at least 6 verified blocks (1 hour) appended to the chain. For this reason, cryptocurrencies like Nxt makes use of other consensus algorithms like a PoS in attempts to improve these limitations. There is also another cryptocurrency called Ethereum which makes use of PoW but it attempts to switch to PoS.

3.3.1.2 Proof of Stake (PoS)

Even though the purpose of this Proof-of-Stake algorithm is the same as the Proof-of-Work algorithm, the process for validating the transactions is completely different. The main objective of this technique aims to avoiding long computationally expensive sequences of cryptographic operations leading to high energy consumption. The first idea of the PoS was proposed in the online forum bitcointalk.org in 2011, but the first cryptocurrency that made use of this method was the Peercoin in 2012 and Nxt followed in 2013.

Unlike the PoW algorithm where miners try to solve the mining operation, in the PoS, there are validators or forgers to proceed to that operation. The purpose of the forgers is to validate the transactions as to create new blocks. The selection of the forgers is done through the amount of coins (stake) that each peer holds in the network. The participant that owns a bigger amount of stake has a bigger probability to be a forger. For example, if Alice is holding 100 tokens, Bob 80 tokens, Daniel 60 tokens and Luis 20 tokens, then Alice has 5 times more chances to be the forger or validator than Luis. Despite PoW, where miners get the rewards when they solve the mathematical puzzle, in PoS, forgers receive only the transaction fees from the corresponding block.

But this situation leads to centralization, because the participant who owns the largest amount of stake in the network, he will always be chosen as a forger for the creation of the block. As a result, he would be the only one who would earn the transaction fees controlling the whole network. In order to prevent the above problem different cryptocurrencies make use of different methods. For example, the selection of the forger in Peercoin is based not only on stakes but also the coin age where is the amount of time that the forgers hold this stake. In addition, Nxt cryptocurrency chooses randomly forgers based on their stake. Through the randomization and the coin age, PoS prevents the issue of centralization, something which happens in PoW through the mining pools.

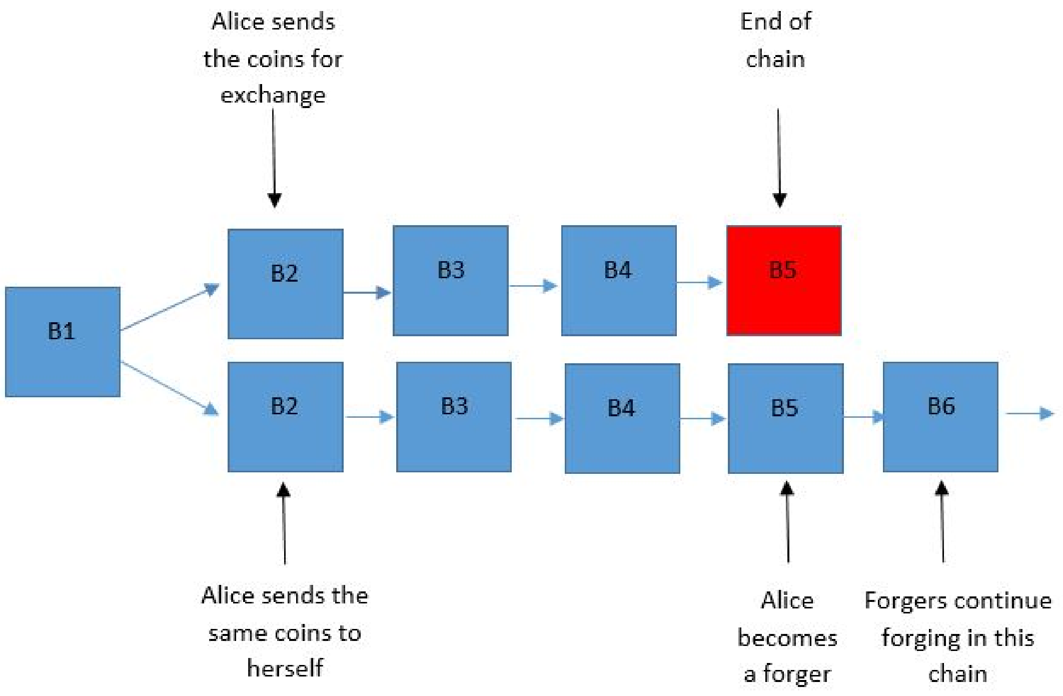
The coin age that Peercoin [20] makes use, refers to a numeric value which indicates the number of coins that belongs to the user multiplied by the duration in days that a user hold the coin. For example, if Alice received 20 coins from Bob and held it for 30 days, then the coin age of Alice is 600 coins-days. In order to proceed to the verification of the transactions or to the creation of a new block, forgers should be waiting for a minimum of 30 days without spending any coin. Usually, the user which holds huger amount of coins and bigger duration than others, he will have the probability of being chosen to forge the next block. Every time that a forger validates a new bock, then a coin age of this forger is being zero and returns at its initial state waiting at least 30 days to be forged again. In addition, the maximum period where the participant can be forger is 90 days. Otherwise, the participant who held old and huge amount of stakes would be able to control the Blockchain.

In Nxt[19] cryptocurrency, the block is generated on average every 60 seconds. In order a peer to be selected as a forger, he should be waiting for a minimum of 1440 confirmed blocks without changing their amount of stake. Through this technique, Nxt prevents the users to using the same stakes in different accounts in order to valid the transactions and generate new blocks. After that, all users can be selected as a forger in a random way. When a validator is chosen then he can validate up to 255 transactions including them in the block. The block size in this cryptocurrency is 42 KB and a forger can choose which transactions wants, usually according to their fees. The Nxt can validate up to 100 transactions per second. Despite Bitcoin, in Nxt the transactions have deadline and this is 24 hours. This means that if a transaction remained in the transaction pool for more than 24 hours and one second then this transaction is rejected. In addition the block can be finalized after the insertions in chain of ten confirmed blocks at about 14 minutes.

One of the main advantages of PoS is that participants do not need to invest a huge amount of money in order to get the reward. There is no need for solving mathematical puzzles and this results less energy consumption as no expensive equipment. So, they can spend their money to buy more coins (stakes) in order to increase the possibility to be a forger and to receive the transactions fees. In addition, PoS provides faster processing of transactions compared to PoW. As in PoW, 51% attack is also possible in PoS but this is very difficult and expensive to implement due to the fact that a hacker should buy and control more than the half of the stake in the network where is a very huge amount of coins.

On the other hand, PoS suffers from what is known as “nothing at stake”. The issue can occur anytime the forks appear. Fork is happened when the chain of the blockchain is divided in two or more chains. This can be taken place for two reasons, the first one is due to malicious action and the second one is when two forgers propose a new block simultaneously. The last one is more difficult to happen in PoS as each forger is chosen by the system in a random way and it is responsible for inserting the block in a specific timeslot. Furthermore, this issue appears more often in the PoW algorithm where all the validators try to solve the mathematical puzzle at the same time. The ideal scenario in case of forking is that the forgers should choose only one of the multiple chains, like in PoW. However, as referred above, it costs them nothing to take part in all the chains. So, the optimal strategy is that all the forgers can support multiple chains without the fear of having something to lose and always they can get the transaction fees from the longest chain of the wining fork.

Usually forgers use forks in order to implement double spent attack as they have ''nothing at stake" to lose. An example of how forgers can proceed to double spend attack is illustrated in the figure 3.7.



Alice sends the coins for exchange
End of chain
B1
B2
B3
B4
B5
B2
B3
B4
B5
B6
Alice sends the same coins to herself
Alice becomes a forger
Forgers continue forging in this chain

Figure 3.7: Example of double spending attack

Supposing that Alice is an attacker and she wants to execute double spent attack by changing her coins to bitcoins. Firstly, once fork has be created, Alice sends the coins to be exchanged in the above chain and the same coins are sent by Alice to herself in order to implement double spent attack. When the transaction becomes valid, then Alice can buy the bitcoins. In order to get the Bitcoin without paying any cost, when Alice is given the turn to validate the next block again, she will validate the block only in the second chain. While all forgers continue validating blocks in both chains, the second will be the longest chain as Alice stopped validating the block in the first chain. As a result, the first chain be dropped and Alice has achieved to add a new amount of coins at her stake by stealing the Bitcoin off an exchange as the transaction fees of the blocks that she insert in the chain. In addition, another cryptocurrency like the Ethereum tried to solve these problems by locking their stake as a deposit in order to be forgers. In this way, forgers are given the incentive to act always honestly, otherwise in case of fraudulent behaviour they are going to lose their deposit.

Another limitation of PoS is that the poorest participants in the network may never have the chance to be forgers in order to validate the transactions getting the corresponding transaction fees. This is unfair because the richer participants have always the greater possibility to be forgers than the others and PoS makes stronger only the richer participants in the network. Leased Proof of Stake (LPoS) solves that domination problem as will be described in the next chapters.

3.3.1.3 Delegated Proof of Stake (DPoS)

Dan Larimer created DPoS in 2013 in order to overcome the limitations of PoW and PoS providing better scalability, flexibility and security in the network. Initially, this consensus algorithm was used in BitShares platform [5] and with the passing of time, DPoS was used from different projects like by Steem, EOS, Ark and Lisk etc. making some changes in the protocol. The main purpose of DPoS is to give the right to all participants that hold stakes (stakeholders) to be an active member through the voting procedure in order to solve the issues in the blockchain in a democratic and fair way.

Unlike the PoS where only the richest participants have the opportunity to take part in the validation of the transaction and the creation of the blocks in order to get the fees, in DPoS, all stakeholders have a significant participation role in the network. Stakeholders are responsible for electing the witnesses and delegates in order to execute the appropriate operations. Each participant has the right to vote in or out any number of witnesses or delegates he wishes in real time. The voting procedure can be valid only when more that 50% of stakeholders participate in election of the witnesses or delegates. In addition, each stakeholder cannot vote one specific participant more than once.

The figure 3.8 depicts the voting procedure by the stakeholders in order to elect the witnesses and the delegates. The yellow circles depict the peers that want to be voted from the other users in the network (witnesses) and the blue ones represent the stakeholders. The voting procedure happens in real time and the stakeholders can vote as many peers as they want to be a witness or delegate according to their stake they have. In BitShares, the list of the votes is updated every day and the number of witnesses amounts to 101. As a result, in the end of each day the top 101 candidates with the highest votes become the witnesses.

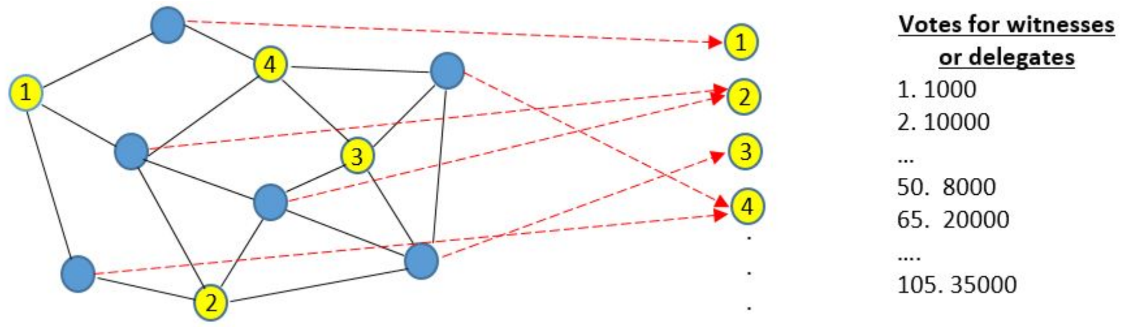


Figure 3.8: Example of voting procedure in DPoS

Furthermore, the votes of each stakeholder depends on the number of stakes that hold and this operation is known as Stake-weighted voting. For example, Alice has 100 coins and Bob 60 coins. If Alice wishes to vote two witnesses, Tom and Charlie, then both of them will receive the 50% of her voting weight. Bob, in turn, will vote only one witness, Bill, and so this witness will receive the 100% of his voting weight. As a result, Bill has a greater voting power than the others and then he will become the witness. Sometimes, this property benefits the weaker stakeholders as they can get the votes from richer stakeholders and can be witnesses or delegates. In addition, another feature of DPoS is that users have the right to choose others trustful users in order to vote on their behalf.

Witnesses are responsible for validating the transactions and creating the blocks. As referred, they are elected by the Stakeholders of the network who have the right to vote out the witnesses at any time they want leading witnesses to behave honestly. In case of behaving badly, witnesses can lose the transaction fees of the block creation as their reputation. In addition, the number of witnesses is limited depending on the different projects that exist so far. For instance, EOS has a specific number of 21, BitShares with Lisk 101 and ARK 51 witnesses.

In BitShares, the list of active witnesses is updated every one day after the end of vote counting. The validation of the blocks in DPoS is done in rounds. Each block is produced by the witnesses every two seconds and the validation can be executed only once for each witness until the round is completed. The previous procedure is depicted in the figure 3.9. Assuming that the system has four witnesses the 1, 2, 3 and 4 which are responsible for inserting the block in the chain. In each round, each witness has a specific time slot that can validate the transaction and the insertion of the block in the chain. For example in the first round, witness 4 will be the first one which will insert the block, then follows the witness 1 and etc. In case that witness 3 misbehaves in its specific time slot, then the block is skipped and the next block is given the turn to be validated by the next witness which is 2. Due to that behaviour, witness 3 can lose its reputation and the transaction fees of this block. Once the round has finished, then witnesses are mixed randomly and the procedure starts again with witnesses validating new blocks as appears in the second round of the figure.

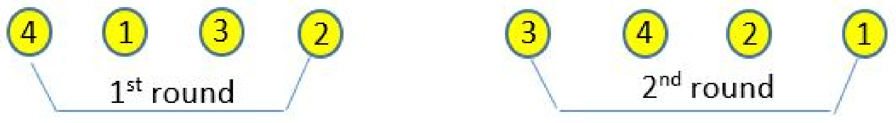


Figure 3.9: Example of validation procedure in DPoS

The majority of the projects that make use of the DPoS consensus protocol, except of the witnesses, presents, also, the delegates. Although delegates are elected in a similar manner to witnesses, their purpose is to propose changes for better maintenance of the network. Some of the proposed parameters that have to be changed are the block size, block interval and the amount of transaction fees that witnesses will receive. Once the changes have been implemented, the stakeholders, in turn, have a specific period of time (i.e BitShares two weeks) to decide if they proceed to the modification of parameters or not. Usually, changes in the network are not implemented easily because bigger the network is, more difficult for 51% of stakeholders to vote for changing the network. In few words, neither the witnesses nor the delegates are able to control the network, but the Stakeholders are they who have the greatest power in the network.

Compared to PoW and PoS, DPoS provides better performance in terms of confirmation of transactions. For example, PoW and PoS validate 7 transactions per second and 8 transactions per second respectively. However DPoS, theoretically, can achieve 100.000 transactions per second but in practice only 3300 transactions have been proven so far. In addition, all the participants have an active role on the network due to the real time voting, as a result, the network becomes more secure rather than the other consensus protocols.

Once the network has detected any malicious behaviour, is able to inform the stakeholders about this misbehaviour during 1 minute and then stakeholders, in turn, can vote out this specific witness. Another significant advantage is that even weaker stakeholders can be elected as witnesses getting the rewards.

Moreover, DPoS prevents the 'nothing at stake' problem which is the main limitations of PoS. First of all, as it was referred previously, when a witness behaves in a malicious way, network realizes this situation fast giving the chance to the stakeholders voting out the malicious witness in order to replace him. As a result, witnesses are given the incentive to behave honestly to keep not only the future incomes but also their reputation. Furthermore, the order of the blocks in each round is known by all the participants in the network and in each round the witnesses are arranged in a mixed up order in order to validate the blocks. This means that in each round the witnesses validate the blocks in different timeslots. In addition, in case of forking, the DPoS algorithm chooses the highest chain in order to continue the insertion of the block in the chain. In that way, it would be very difficult for an attacker to create forking longer than the main chain. The figure 3.10 represents the canonical form of how the blocks are ordered in the chain with the elected witnesses. The outlined blocks indicate the start point of each round.

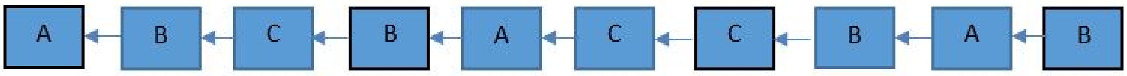


Figure 3.10: Form of canonical chain

In the figure 3.11, let us suppose that witness B wants to make a fork in order to behave maliciously. In that case, Bitshares always chooses the chain with the highest witness participation rate. This rate can be estimated by comparing the required number of produced blocks with the number of existed blocks. For example, if there are only 3 witnesses A, B and C, per round then, according to the figure 3.11 the participation rate of the top fork is 3 required number of produced blocks vs 2 existed blocks and on the bottom fork is 3 vs 1. So, the next witness C will continue validating the new block in the highest chain where A and B coexist, and not in the second chain where only B witness appears.

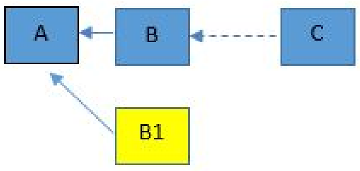


Figure 3.11: Fork with one peer behaving maliciously

Even in the worst case where a majority of the elected witnesses behaves maliciously creating multiple forks, the honest witness will define which chain will be the longest. Despite this situation, the DPoS algorithm has the privilege to identify the nodes which behave badly and to replace them with the honest ones. Especially, each witness is scheduled to valid one block for one time slot in each round and every time that witness produces forking creating more than one block, then there will be clear cryptographic evidence where the network can identify. For example, as depicted in the figure 3.12, the witness B created three blocks with the same timestamp and block height.

On the other hand, DPoS, also, presents some limitations. One of them is that a

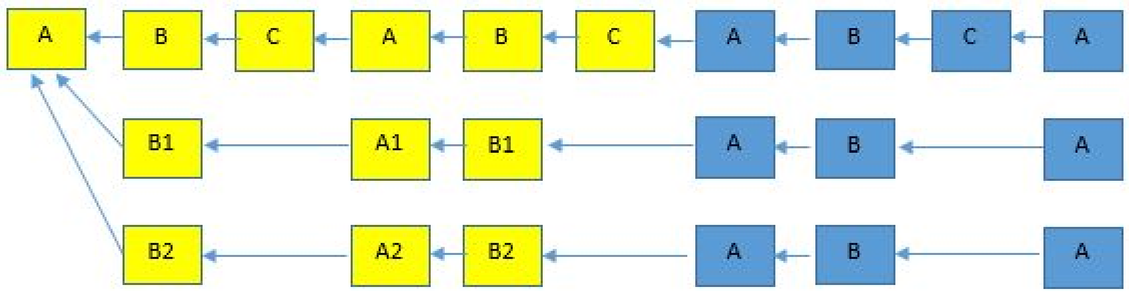


Figure 3.12: Fork with the majority of peers behave maliciously

stakeholder has the right not to vote in order to elect witnesses, as a result, the network becomes less secure and trustless. More peers vote, more secure and trustful the network is. Furthermore, the stakeholders who have the largest amount of stakes have the possibility to cooperate with other rich participants voting each other in order to control the network. As a result, this could lead to the centralization.

3.3.1.4 Leased Proof of Stake (LPoS)

The LPoS was invented to overcome some issues of the Proof of Stake (PoS) [21]. In PoS, the participants are chosen to be a forger based on their stake that hold. This means that, only the peers that hold a huge amount of stakes have the privilege to participate and not the weaker ones that have no opportunity to be chosen. The LPoS solves that limitation by giving the right to the weaker to be active in the network. In that case, the peers can lease their stakes to richer peers and when they validate the blocks, both of them take advantage of the reward.

The participants in LPoS can be divided in two categories: those who run a full node in order to be forger and that ones that leasing their stakes to the full node. The selection of the forger is executed like in the case of Proof of Stake. The higher the stake that a full node has, the more probability to be chosen in order to validate the transactions and create the next block. Once a full node has been chosen and the block has been inserted in the chain, then the reward of the transaction fee will be given to the full node. In addition, leaser will receive a percentage of the transaction fee. For instance, if a full node has 100000 coins and leaser leases 1000 coins to the full node, then the reward of leaser will be 1% of the transaction fees.

The main advantages of this algorithm is that all the participants in the network are active members making the network more secure. In addition, the leasing is safe as the stakes that are going to be leased are always locked in the owner's wallet. Furthermore, each leaser has the right to cancel the leasing operation every time he wishes.

Despite these benefits, centralization is still concerned as a few members can control the whole network.

3.3.2 Vote-Based consensus algorithm

3.3.2.1 Practical Byzantine Fault Tolerance (PBFT)

In 1999, Miquel Castro and Barbara Liskov introduced the algorithm "Practical Byzantine Fault Tolerance" (PBFT)[7], which was the first practical solution against the problem of the Byzantine generals. It should be clarified at this point that the problem of the Byzantine generals is presented when different nodes in an unreliable network must reach a final decision, checking the data exchanged. Especially, the PBFT algorithm was regarded as the first practical high performance consensus algorithm that is suitable for use in asynchronous networks like the Internet.

PBFT consensus algorithm is used in private networks, where the number of peers is limited in comparison with public networks, which consist of millions of users. Furthermore, in permissioned network, the participants are known each other and they are given the permission by the administrator of the network to participate to that. This algorithm is based on state-machine replication and voting mechanism by exchanging a lot of messages among the peers in order to reach the consensus in a proper way. In addition, the algorithm can work properly if there is an equal or smaller number of (R-1)/3 faulty nodes in the network, where R is the total number of nodes. Especially, the network consisted of 7 nodes can tolerate up to two faulty nodes.

In a PBFT system, the participants are divided into two categories, the primary (or leader) node and the secondary (or backup) nodes. The primary is responsible for the validation and order of the transactions inside the block as the insertion of the block in the chain. Every node in the network can act as a primary node. After a specific period of time, the new primary replaces the old one through the operation of view change mode. In addition, the view change can happen when the primary node behaves maliciously or stop working. For instance, in Hyperledger Sawtooth [23], network switches to a new primary after the appendage of 100 blocks to the chain. Moreover, the backup nodes exchange messages each other to verify the block and to check if the primary node behaves honestly.

The procedure of committing the block in the chain in PBFT algorithm is divided in three phases:

- Pre-prepare
- Prepare
- Commit

Firstly, as described to the previous consensus algorithms, each node in a PBFT system has a pair of keys, the private (secret) and the public key. Not only does this algorithm make use of the digital signature, but also the Message Authentication Codes (MACs) [8] in attempts to authenticate all of the messages. Digital signature is used to authenticate only the messages that correspond to view-change and a new-view and the MAC is used to the rest of the messages making the system faster.

The MAC algorithm makes use of symmetric encryption where two peers A and B can verify the communication based on a share secret key in both directions. Specifically, the two peers make use of a couple of session keys k_{ab} and K_{ba} where the first one is used for the calculation of the MAC for a message that sends from peer A to B and the second key for the reverse operation. The MAC of each message is computed by creating the MD5 of the concatenation of the share secret key with the corresponding message. Then the peer A sends both the MAC and the message to the peer B and peer B in turn, calculates the MAC of the message that received form peer A with the share secret key. In the figure 3.13 is depicted the previous procedure. If the two MACs are the same then the peer B authenticates the message and vice versa. In addition, in PBFT algorithm, only the last 10 out of 16 bytes of the MD5 are used in order to reduce the size of the MAC and make the procedure ever faster.

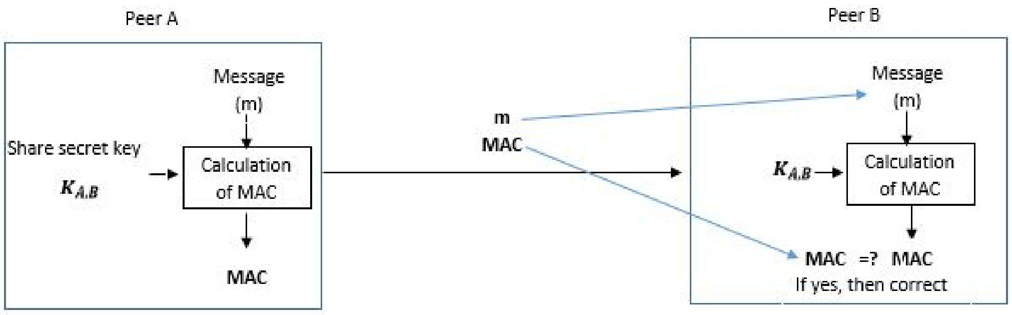


Figure 3.13: Procedure of message authentication code

However, through this technique a third peer cannot verify the authenticity of the message. This is one of the main reasons why digital signatures provide more security than the MAC. Despite this limitation, the developers of PBFT algorithm make use of the MAC by improving this algorithm. In case of multiple recipients, they use the authenticators instead of the digital signatures which is the vector of MAC and are symbolized as $\langle m\rangle\alpha_i$ where m is the message and a is the signature of the authenticator i. Messages like client request, pre-prepare, prepare commit etc. are signed by the authenticators but this doesn't happen in case of view change and reply messages. In the presence of one recipient, the digital signature is replaced by the MAC and it appears like $\langle m\rangle\mu_{AB}$ where peer A sends the message and the MAC to peer B. A reply message is an example where only one recipient appears.

According to the PBFT procedure, a node c sends a request to execute a transaction by sending a message m to the primary node. Every message m signed by the sender with its private key σc includes the timestamp t which is the time of message creation, the c which is the name of the node and o which is the implementation of the state machine operation. This message has the form of $\langle REQUEST, o, t, c\rangle\alpha_c$.

Afterwards, the three phases of PBFT algorithm are executed. Once the primary node has received the request message m, creates a sequence number n of this message and then it sends a pre-prepare message accompanied by the request message m to all the backup nodes. The sequence number determines in which block the message belongs to. The pre-prepare message forms as $\langle\langle PRE-PREPARE, v, n, d\rangle\alpha_p, m\rangle$ where v is the current view of the message creation, n is the sequence number of the request, d is the hash of the request message m and α_p is the signature of the authenticator primary node. Notice that the request message m is not directly included within the pre-prepare message, but appended at the end. The objective is keeping the size of this message small. All the backup nodes store the message to their logs if:

- The id of the request and the pre-prepare messages are valid and if the hash of m is the same as the d.

- The view v of the pre-prepare message is the same as the current view of the network
- There is no other pre-prepare message in the logs of the nodes with different hash of the request message and the same sequence number and view.
- The sequence number of the message should be between the low water mark h and the high water mark H. Through this technique, malicious nodes are prevented from limiting the space of sequence number by choosing a number near to H. The h is equal to the sequence number n of the last stable checkpoint and H=h+k where k is a huge number in order to create a big space between h and H.

If the above criteria are fulfilled, then the backup nodes store the pre-prepare and request message on their logs and they move to the next phase where all backup nodes send a prepare message to all the nodes of the system. The Prepare messages have the form of $\langle PREPARE, v, n, d, i\rangle\alpha_i$, where i corresponds to each node that sends the message. Once the nodes have received the prepare messages, they check the correctness of signature of the prepare messages as their view is equal to node's current view and the sequence number is between the h and H. If the previous conditions meet the requirements then, each node stores the message to its logs.

In order to proceed to the next phase, nodes have to ensure that they stored the request, the pre-prepare as the prepare messages in their logs. They have, also, to verify that the pre-prepare messages match to the prepare messages. If, specifically, the view, the sequence number and the digest of pre-prepare are equal to the contents corresponding to the prepare messages. Finally, if each node has, also, received 2f prepare messages from different backups, they move to the final phase.

In the commit phase, all the nodes, including the primary node send a $\langle COMMIT, v, n, D(m), i\rangle\alpha_i$, message to all the nodes in the network. After receiving the commit messages, they check the validity of these messages in order to store them in their logs like in the prepare phase. Afterwards, once nodes have received 2f+1 commit messages that match to the pre-prepare messages such as the view, the sequence number and the digest, then nodes send directly a reply message $\langle REPLY, v, t, c, i, r\rangle\mu_{i,c}$ to the node, where t is the timestamp of the corresponding request and r is the final result of that request. If node receives f+1 reply messages from different nodes with the same timestamp and result, then it accepts this result. Otherwise, it sends the request to all the nodes of the system and if the rest of the nodes has already executed the procedure of three phases then they send only the reply message. The figure 3.14 depicts the procedure of PBFT algorithm.

Besides the operation of the three phases, there is, also, the view change mode, which plays a significant role on PBFT algorithm. This mode occurs when there is a faulty behaviour of nodes or abnormal condition in the network and when the mandatory change of the primary happens after a specific interval of time (100 blocks in Hyperledger Sawtooth). When a primary node receives a request m from node c, then all the backup nodes start a timer to check that all the nodes will receive the corresponding message in a given time, which is determined by the system. On the other hand, unless they have received these messages before the timer expires, then they send a view-change message. Once the view-change procedure has started, all the nodes accept only the view-change, new-view and checkpoint message.

The view-change message has a form of $\langle VIEW - CHANGE, v + 1, n, C, P, i\rangle\sigma_i$, where v+1 indicates that the primary at view v is faulty and it has to be moved to the

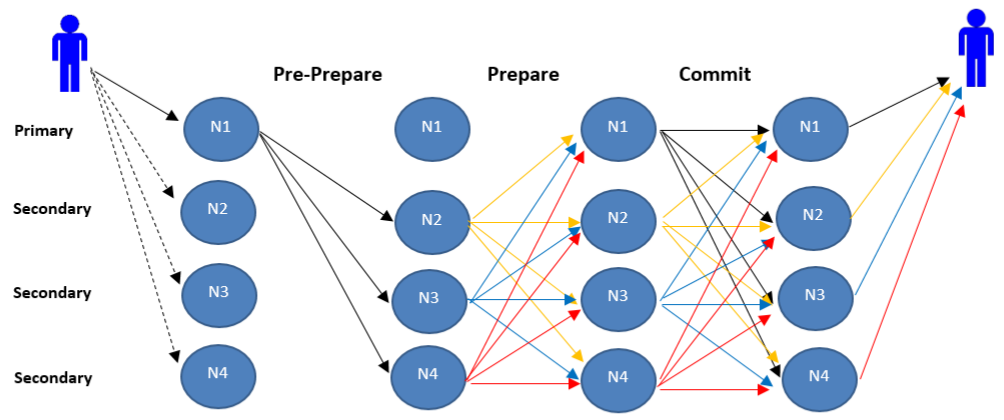


Figure 3.14: Procedure of PBFT algorithm

next view v+1 with the primary that accounts for that view. Furthermore, n is a sequence number of s, which is the latest stable checkpoint and C is a set of 2f +1 messages proving that the checkpoint s is stable. In addition, P is a set of the Pm messages where the sequence number of them is higher of the sequence number n of the stable checkpoint s. Pm includes all the valid pre-prepare messages and 2f prepare messages that match to the pre-prepare messages. When the new primary of v+1 view receives 2f valid view-change messages from different nodes send a new-view message to all nodes and starts the normal operation.

PBFT uses another technique to reduce a huge amount of messages that can be stored in the logs of each node. This method is called garbage collection. After an interval of time determined by the network, nodes create checkpoints by broadcasting a message $\langle CHECKPOINT, n, d, i\rangle\alpha_i$ to other nodes. The sequence number n indicates the latest message request m that executed in the system and d is the digest of the corresponding phase. When each node receives 2f +1 messages from different nodes with same both sequence number n and digest d, the checkpoint becomes stable. As a result, all the checkpoints and the messages that have lower or equal sequence number with n are deleted from the logs.

3.3.2.1.1 PBFT in Hyperleder Sawtooth

As mentioned previously, Hyperledger Sawtooth, except from the Proof of Elapsed Time (PoET) makes use of PBFT consensus algorithm [22]. In Sawtooth, a primary node is responsible for creating a block as to insert the block in the chain. each block is inserted to the chain aproximately every 1000 msec. Firstly, the primary node creates the block and sends a signed request to its validator. Then, the validator sends this request to all other validators of the network to check if the signer of this block is the real one and then they store this request to their PBFT logs by informing all the nodes with a Block new update message. After that, a primary node assigns a sequence number to the request and sends pre-prepare messages without the request message m to all the backup nodes like in the classical case of PBFT.

Then, the procedure is the same as the classic PBFT until the end of the commit phase. In the final step of the insertion of the block, instead of all nodes have to send a reply message to the request node, in Sawtooth each node informs its validator to commit the block in the chain by sending a Commit Block message. This is done only if nodes have received 2f+1 commit and 2f+1 prepared messages that match to the pre-prepare messages such as the view, the sequence number and the digest. Furthermore, validator sends to its node a Block Commit update message if the block is inserted to the chain successfully. Then the primary node starts the procedure for the creation of a new block.

In addition, when a primary node sends a pre-prepare message, all backup nodes start an idle timer where a primary has to send the pre-prepare message to all nodes in less than 30 seconds. If the primary exceeds this time, then backup nodes send a view-change message. Moreover, in the prepare phase, all nodes start a commit timer where each node has 10 seconds to continue the procedure until informing the validator to commit the block in the chain. If the time expires without any result then a view-change message will be sent to the new primary.

Unlike the classic PBFT where there is a garbage collection and the checkpoint to reduce the contents of the logs, Sawtooth has the Log Pruning procedure. Each node in the network implements this procedure when the log size of each node exceed the 10000 messages. They delete all the messages where the sequence number are smaller than the last block sequence number that committed in the chain.

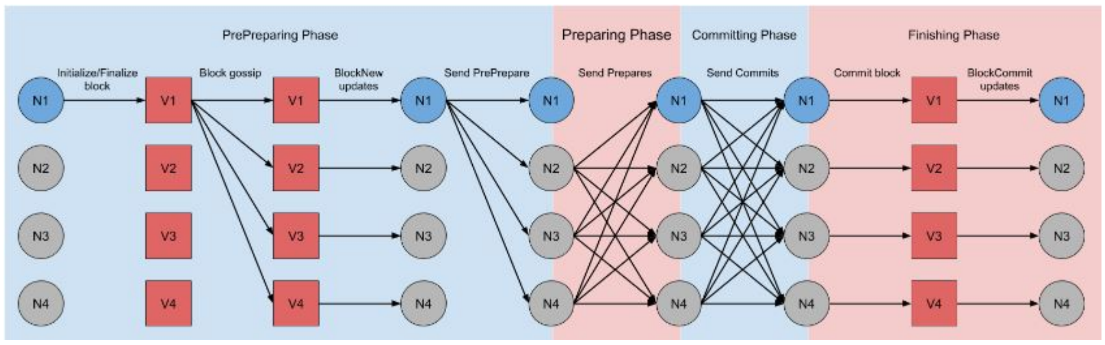


Figure 3.15: Procedure of PBFT algorithm in Hyperledger Sawtooth

The PBFT algorithm suffers from scalability when the number of participants in the network increases. All the participants in the network have an active role by exchanging a lot of messages each other in order to reach consensus. As a result, more peers in the network, more time is needed to get 2f +1 commit messages. Last but not least, the network suffers from the Sybil attack that occurs when a single entity can manipulate huge amount of nodes in order to control the majority of the network. This limitation can be avoided when the number of peers in the network increases.

3.3.2.2 Federated Byzantine Agreement (FBA)

In 2015, David Mazieres introduced the Federated Byzantine Agreement (FBA) system [15], which constitutes the base of Stellar Consensus Protocol (SCP). FBA was invented to overcome some limitations of Byzantine Agreement (BA). This type of algorithm, in essence, is used in public networks where every participant can join and leave the network at any time. As a result, the consensus in FBA cannot be achieved by the majority of participants voting like in Byzantine Agreement Protocol (i.e. PBFT).

Furthermore, an agreement on state updates between nodes is succeeded by voting for a specific slot (i.e. block) in each Federated voting round. Quorums should have to agree to the same slot in order to update the ledger [15]. FBA uses quorum and quorum slice to come to an agreement. A quorum is a group of nodes (quorum slices) which consists of at least one quorum slice of each node and this group votes for a unique slot in order to update the ledger. The quorum slices are subsets of quorums which are responsible to convince peers to reach an agreement. The nodes have the right to choose which nodes they trust in order to create their own quorum slices making the network more decentralized.

As referred previously, FBA does not depend on the votes of the majority of the peers as the network is public and the peers can join or leave the network at any time. Therefore, FBA uses the quorum slices and the quorums to overcome this limitation. More specific, each node can choose its own quorum slice that it trusts but this slice may consist of other nodes, which have different quorum slices. Quorum is formed by quorum slices where each one is interconnected to other. Furthermore, quorum can consist of at least one quorum slice. A node, which belongs to a specific quorum slice, can be shared by multiple quorum slices.

Figure 3.16 depicts an example of quorum and quorum slice. Nodes V2, V3 and V4 form a quorum as each one depends on each other. Specifically, each of these three nodes has its own quorum slice and their decision making is highly affected by the others. However, V1, V2 and V3 do not form a quorum but form a quorum slice. Although the V1's decision making depends on V2 and V3, it isn't applicable for V2 and V3 on the grounds that the decision making of V2 and V3 relies on the V4 and not on V1 node. In addition, V1, V2 and V3 are not a quorum as the quorum slice of V2 and V3 are not included in this group. A possible quorum with V1 included is V1, V2, V3, V4.

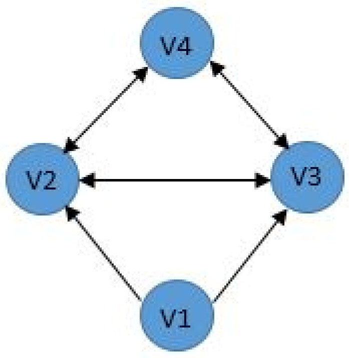


Figure 3.16: Example of quorum and quorum slice

Another example of a quorum is presented below in figure 3.17. In this case, the system is divided by three tiers. The top tier consists of V1 to V4 nodes and it can tolerate one byzantine failure node f, the middle tier includes V5 to V8 nodes and the last tier is composed of V9 and V10 nodes. In the first tier, the nodes depend on each other and can form a quorum slice but the rest of the tiers forms a slice by picking nodes from the above tier. More specific, a node belonging to the middle tier can form a slice by choosing only two nodes from the top tier. As the top tier tolerates one failure node then three out of four nodes act honestly. On the other hand, in case of choosing only one node from the above tier, it there would be the possibility to form a quorum slice with one malicious node by affecting the normal operation of the system. So, the system will work properly by forming quorum slice with two nodes as there will always be at least one honest node. For example, a quorum can consist of three quorum slices which interconnect each other. A possible combination could be one slice in the top tier consisted by V2, V3, V4 nodes, another slice in the middle tier formed by V7, V2, V1 and the last slice in the leaf tier consisted by V9, V7, V8.

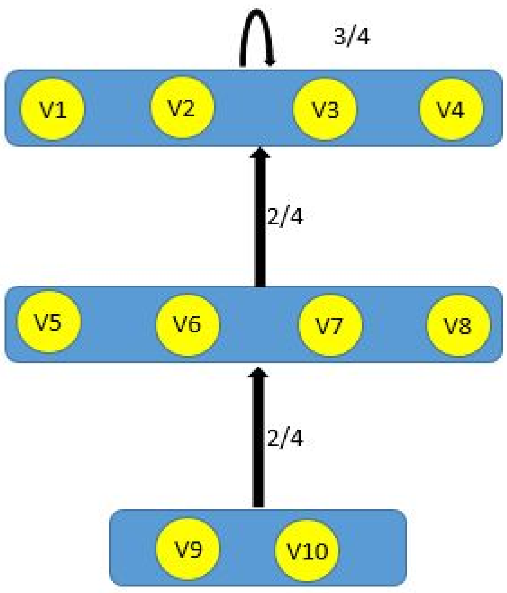


Figure 3.17: Example of tier quorum

In [15], David Mazieres gave an in depth example in order to explain how the FBA and the quorum tier works. This example aims at preventing the double spent attack and is depicted in the figure 3.18. The first top tier consists of one tier of four banks and another parallel tier of three non-profit organizations. The rest of the tiers represents the clients. For example, V4 wants to make a transaction with the V7 client by sending 10000 XLM (Stellar digital currency) to him. On the grounds that V7 client does not trust these banks, it wishes to reach consensus by trusting one of the non-profit organizations.

So, the quorum consists of three quorum slices. As referred previously, the first tier consisted by banks forms a quorum slice with three out of the four banks (V2, V3 and V4). In addition, in the parallel top tier, two out of the three non-profit organizations need to constitute a quorum slice (V11 and V12). A node in the middle tier (V7) forms a quorum slice by picking two banks from the top tier (V3 and V4) and one non-profit organization from the parallel top tier (V11). V7 client is going to receive this amount of money if and only all the corresponding nodes agree to that transaction (reach consensus). Through this method the double spend attack could be prevented.

In case that V2, V3 and V4 banks behave maliciously and want to send the same amount of money to V8, V8 should contact one of the non-profit organization to reach an agreement. Suppose that now V8 trusts V13 and forms a quorum slice with V3, V4 and V13 in order to proceed to the desired transaction. The node V13, in turn, forms a quorum slice with any one of the two non-profit nodes V11 or V12. As the other two nodes have already accepted the same transaction for the V7 client, then they reject the transaction related to V8 as a result V8 does not accept the money from the banks.

Besides the above presented scenario, in FBA system there are two other critical conditions in order for the system to reach consensus: safety and liveness. Liveness is achieved when the system reaches an agreement without the presence of any malicious node. The safety is determined when at least two quorums of the whole network agree at the same statement. This can be achieved only through the quorum intersection. Supposing that there are two independent quorums, the V1, V2, V3 and V4, V5, V6 as shown in the figure 3.19 without the presence of V7. If there is not a shared node (V7)

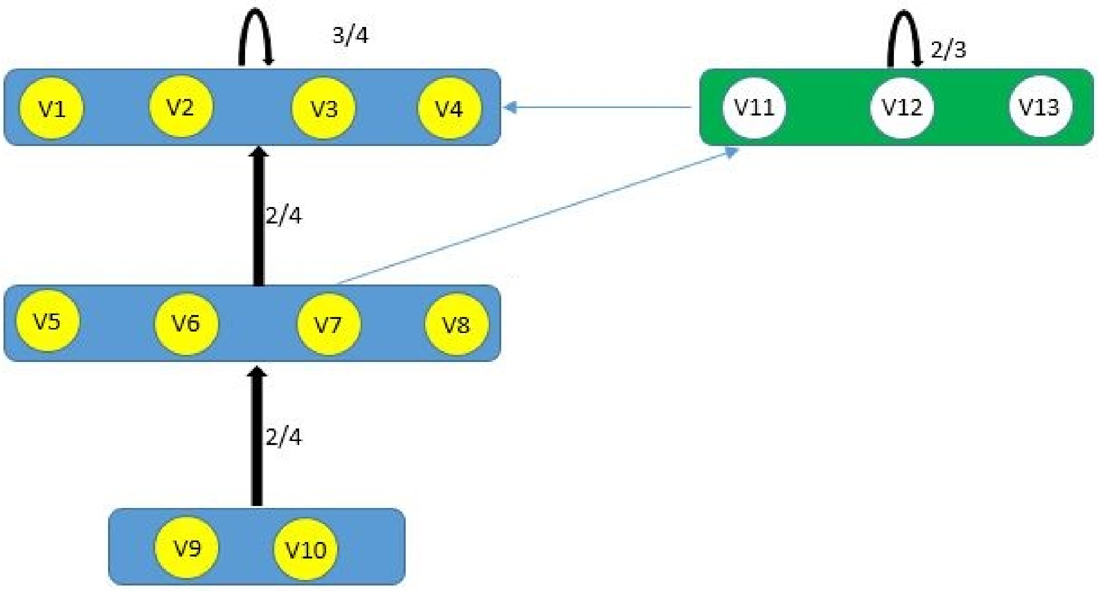


Figure 3.18: Example of parallel tier quorum

that intersects with these two quorums, then the system will get stuck as it ends up with two different agreements and it will never reach consensus. V7 acts as an intermediate node between two quorums.

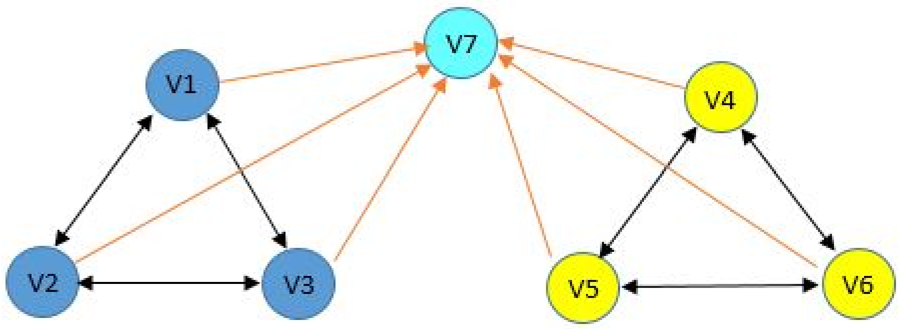


Figure 3.19: Quorum intersection

The FBA system divides the nodes into two categories, the well-behaved and the ill-behaved nodes. The ill-behaved nodes are known as the byzantine failure nodes which behave maliciously by sending erroneous messages to other nodes or stopping to send any kind of messages. Furthermore, ill-behaved nodes are those that do not work properly due to Internet connection or misbehaved operation. As a result, this kind of nodes does not help the system to reach consensus.

On the other hand, well-behaved nodes are those that choose the suitable quorum slices they trust exchanging correct message with other peers by voting only valid transactions. The presence of well-behaved nodes guarantees safety and liveness. Not only can the ill-behaved nodes fail, but also a well-behaved node can misbehave too. The well-behaved nodes that behave maliciously are characterized by blocked and divergent nodes. The blocked nodes are the nodes, which cannot reach an agreement due to the lack of liveness. The divergent ones are characterized by those nodes, which vote for a state where other nodes contradict that state. As a result, the sa fety cannot be achieved.

In the FBA system, the nodes can reach consensus for a specific slot (block) through the Federated Voting procedure. This procedure is composed of three states: the voting, the accepting and the confirming state. All nodes that belong to the quorum exchange messages with each other to check if the block is valid or not and then vote for or against that specific block. In addition, there is a possibility in the voting procedure to get stuck when nodes support contradictory states for that block. FBA can solve this by making use of view change or ballot method. Furthermore, the nodes accept the block and then the block is confirmed when there is a unanimous agreement for this specific block by all the members in the quorum.

In the voting phase, a node N votes for a specific block X only when it declares that it is valid assuring that it has never voted against for the block X and it will never vote against in the future as depicted in step 1 in the figure 3.20. When a node has voted sends a message to all other members of its quorum to inform them that it is a node N belonging to quorum slices S and it votes for block X. Furthermore, this vote is permanent and unchangeable.

As the nodes in the network exchange messages into these three phases, there exist two kinds of thresholds responsible for the transition of one state to another in the federated voting procedure, the quorum and the blocking threshold [11]. The blocking threshold ensures that at least one peer of each quorum slice where node N belongs accepts block X, regardless of the participation of N. The figure 3.21 represents analytically the blocking threshold which is actually the v-blocking. The quorum threshold is defined when all the peers of N's quorum accept a specific block. The usage of both thresholds are explained analytically in the following paragraph.

When the majority of the nodes voted for or against block X, then the acceptance phase, in turn, takes part and nodes accept what they have voted (see step 2 in figure 3.20). More specific, nodes accept block X if they have never accepted a contradicted state for block X. There is the possibility for the node N to accept block X even if it has voted for a different block (i.e it has voted for block Y, as depicted in the step 3 in figure 3.20). This can happen when at least one node of each N's quorum slice accepts block X (blocking threshold). Furthermore, when a node N accepts block X, it sends a message to the rest members of the quorum informing about its identity, the set of the quorum slice that N belongs and the acceptance of block X. The acceptance phase acts as a second vote that can confirm that all the peers of the quorum agree on a specific block (quorum threshold).

The last state is the confirmation phase. When the acceptance messages of a specific quorum reach the quorum threshold (step 4) then all the peers in the quorum move to the confirmation state and they confirm block X. Afterwards, the node N may send a message that has confirmed the block X in order to affect other peers' decision in the whole network to accept this block even if they have voted against it. When at least two intersected quorums confirm the same block, then the federated voting procedure is done and the system reaches consensus.

Despite of all of these advantages, some researchers consider that FBA suffers from some limitations [16]. The most significant disadvantage is that quorum slice leads the system to be more centralized than decentralized, which is the main purpose of FBA system. As the nodes have the right to choose what quorum slice they trust in order to participate in the network, they usually choose only some large companies and organizations such as IBM, satoshiPay etc. Furthermore, there is neither incentive nor a mechanism to lead

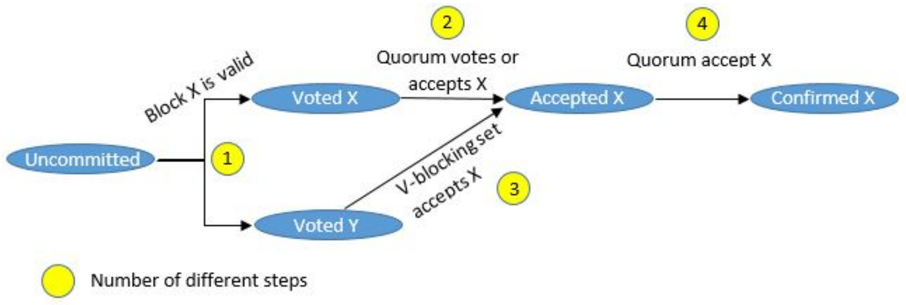


Figure 3.20: Federated voting procedure

other nodes to behave as validators in order to reduce this centralization.

Another disadvantage of the FBA is that the centralization leads the system to be vulnerable to cascading failure [16]. In blockchain, this happens when the peers of the network interact with each other. As a result, a possible failure of some peers can provoke the failure of the other peers in the network too. In case of FBA, due to centralization, researchers have proved that the system is going to crash if only two validators belonging to stellar foundation are deleted. This problem can be solved by introducing a large number of validators in the network (decentralization).

3.3.2.3 Stellar Consensus Protocol (SCP)

As referred previously, SCP is based on FBA and was introduced by David Mazieres in 2015. Nowadays, SCP is one of the most important decentralized consensus algorithms achieving transactions in just a few seconds. In addition, all the nodes have the right to trust which nodes they want in the network and provide safety through the digital signatures and hash functions like SHA-256. Stellar is used for any kind of payments as well as for the implementation of smart contracts over the world, and has a native token known as lumens (XLM). When Stellar was created, lumens amounted to 100 billion but with the inflation rate increasing by 1% per year [24] the number of lumens has risen up to 105.2 billion [25].

In the stellar network, each user has to create a seed in order to be an active member. Through this seed, a user can generate its own public and private key pair. Stellar uses public-key cryptography to ensure that transactions are secure. With the secret seed, users prove that their own a given account and Stellar provides them full access to them (including its pair of keys). When a user implements a transaction has to sign this transaction with its own private key. Then, it broadcasts to the whole network and the other peers can verify the validity and the identity of this transaction through the public key of the executor of the transaction.

According to the structure of SCP, nodes are divided into four categories: watchers, archivers, basic and full validators. A user has the right to choose which role can take in the network according to its interests. All of them can implement various transactions in the network such as execution of different kind of payments, or give the right to another user to manage its own lumens, etc. The first category refers to watcher nodes who have fewer responsibilities than the nodes in other categories. The main responsibility is to observe and monitor any activity that happens in the network. For example, they watch which validators are active in the network.

In the second category, archivers not only can behave as a watcher node, but also they store all the activities in the Binary Large OBject (BLOB) store. The BLOB is a cloud space storage where a massive amount of data are stored there, like images and audio texts. Microsoft Azure cloud and Amazon Web Services (AWS) are the two most known BLOB stores that users use.

On the other hand, the last two categories have a significant role on the network as the basic and full validators participate in the consensus procedure. Firstly, the basic validators have the same duties as watchers except that watchers do not participate in the consensus procedure. In addition, they have the right to stop any procedure they consider invalid.

However, the full validators are the only category which participates fully in the network. This kind of nodes is responsible for storing all the activities in the network in the archive every 64 blocks (at about 5 minutes) as to inform newcomers in the network about the data stored in the ledger. Furthermore, full validators can behave as a v blocking node where at least one of these nodes are presented in each quorum slice in order to persuade other nodes in the quorum slice to agree on the same value. Figure 3.21 depicts node 20 and all the quorum slices of this node. Node 20 consists of three quorum slices the nodes (2,3,4) with blue arrows, (5,6,7) with black arrows and (7,8,9)with orange arrows. So, the yellow nodes 2, 5 and 9 could be the v-blocking nodes of node 20 and they can persuade node 20 to accept i.e. a block x in case that node 20 votes for a different block i.e. the block y. In addition, as depicted in the right side of the figure 3.21 another possible combination of v-blocking nodes are the nodes 2 and 7 as node 7 is appeared in two quorum slices.

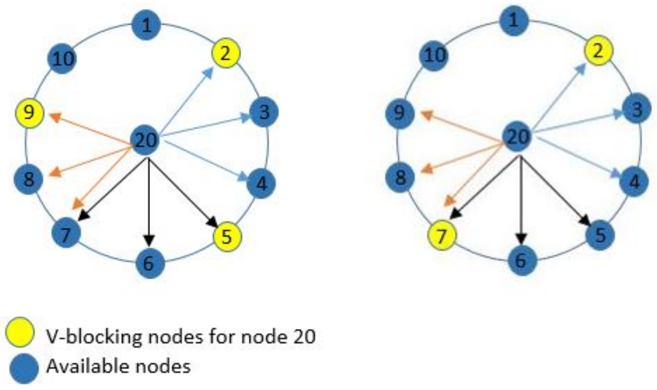


Figure 3.21: example of v-blocking

The number of full validators indicates how decentralized the system is. The bigger the number of full validators, more decentralized the system could be. Like the archivers, full validators need a blob store to record all the activities. Finally, the nodes from all the categories need a database to record the blockchain.

As referred previously, all the nodes in stellar can execute transactions on the condition that each node's account has a minimum balance of 1 XLM. According to the Stellar, this method is necessary in order to stop users overloading the network with useless multiple accounts. Each transaction can consist of multiple operations such as payments or some offers about exchanging different kinds of currencies and etc. This number of operations fluctuates from 1 up to 100 per transaction. In case that any operation included in the transaction does not succeed, then the transactions is discarded.

For example, supposing that Alice has an available amount of 50 lumens and she wants to exchange this amount to dollars and euros. Therefore, Alice executes two different transactions where each one contains one operation. The first transaction is the exchange of 30 lumens in dollars and the second 30 lumens in euros. Alice has only 50 lumens, then only the first transaction can be implemented successfully, and the second is discarded due to the fact that the balance left is not enough. On the other hand, if Alice groups these two operations in one transaction, then the transaction cannot be executed because the lumens in her account are not sufficient for executing these operations.

Furthermore, the successful implementation of one transaction does not only depend on the operations included in the transaction but also on the sequence number that a user assigns to. When a new user creates an account in the stellar network, then this account is given a sequence number, which corresponds to the current specific number of the ledger. For instance, Bob creates its account when the ledger has a height block number of 2000. As a result, Bob's account gets also a sequence number equal to 2000. When he wants to implement one or more transactions, he should assign to each transaction a sequence one number greater than the sequence number of the account. Moreover, if he wants to implement a transaction, then the sequence number of this transaction should be 2001 and once the transaction is used, then the sequence number of the account is updated to the number of the current transaction.

In the stellar network, the nodes have to pay some transaction fees if they want to execute some transaction in the network. These fees consist of two categories: the base fee and the base reserve. According to the first category, the base fee is equal to 100 stroops (0.00001XLM) and a user has to pay the amount of fees of the transaction, which is calculated by multiplying the base fee with the number of operations included in every transaction. Moreover, the second category refers to the base reserve that is equal to 0.5 XLM and is used to calculate the minimum balance of each account, which is 2 times the base reserve. If a basic user wants to implement a transaction and the remaining balance is lower than one XLM then this transaction is discarded.

All the users in the stellar network have the right to pay more transaction fees than the base fee if they want. By paying more fees they will get more possibilities to include their transaction in the current ledger. For example, when the block closes (approximately every 5 sec), if there are more transactions than the predetermined number of validators, then validators propose the transactions with the highest fees. Furthermore, if the number of transactions is lower than the predetermined number of validators, then all users will pay the fees according to the base fee regardless of some users want to pay higher fees. The transaction set includes all the transactions that a validator proposes in order to be included in the current block.

The Stellar collects all the transaction fees in the fee pool and it distributes them once a week through the inflation voting. The distribution of the fees refers to the peers who votes overcome the 0.05%. Each user that has more than 100 lumens in its account can be voted or vote in order to take part in the inflation procedure. Furthermore, the votes of each peer depends on the number of tokens that holds and this is known as weighted voting procedure. For example, if Alice has 200 XLM and wants to vote Bob, then Bob will be given 200 votes. The nodes will receive the percentage of the fees which corresponds to the percentage of the votes they collected.

As referred previously, the stellar consensus protocol is based highly on FBA and only the basic and full validators take part in the consensus procedure. Each validator has the right to choose whom they trust by choosing the more trusted quorum slice. Furthermore, validators are responsible for collecting and validating all the transactions that happen in the whole network. They can check the public key of the executor of the transaction as they can check if the source account has enough token to pay the transaction fees according to the minimum balance requirement. Then validator broadcasts these transactions to all other validators through the quorum slices and quorums.

When it is time to close the next ledger (approximately 5 seconds), the validators collect these transactions into a transaction set in order to append to the ledger. They can validate more than 3000 transactions per second. There is a possibility of existing a multiple number of transaction sets due to delays in the network or other reasons. As a result, the SCP solves this problem in order to agree to one transaction set through two new protocols, the nomination and ballot protocols[11][10][15].

The nomination phase specifically is a federated voting procedure where validators vote for a specific slot which consists of a transaction set as appeared in the first step of figure 3.22. Because the messages of the voting procedure traverse into quorum and quorum slices, the validator can vote as many transaction sets as it wants. For example, validator v1 nominates for a value A and this means that it will keep agreeing on value A and will never contradict to this statement. Moreover, in case that v1 notices that another validator v2 nominates for value B, then v1 can nominate also for value B providing that the values A and B will be never inconsistent.

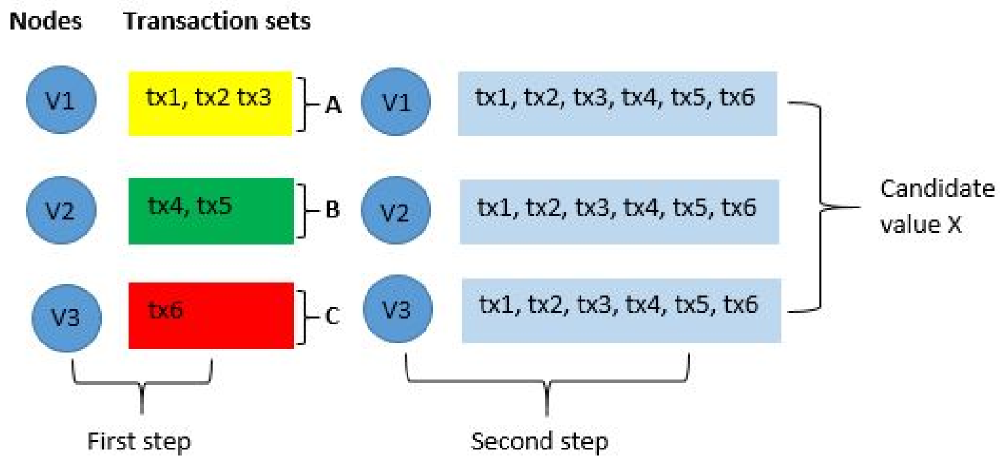


Figure 3.22: Example of nomination phase

As a result of possible multiple nominated values, V1 converges these multiple values in a single value as depicted in the second step of figure 3.22. Also, V2 and V3 follow the same procedure as V1. When all the nodes in the quorum agree to this single value then this value is confirmed and it is considered as a candidate value which is the X as shown in the figure 3.22. However, once the candidate value has obtained, this doesn't mean that the nomination procedure will stop as the validators will continue adding more transaction to their transaction sets. Every time that a quorum agrees on a specific transaction set, a new candidate value is generated. So, the ballot protocol, in turn, can solve the the problem of multiple candidate values and the nomination process finishes when a specific candidate value is externalized through the ballot protocol.

The operation of the ballot protocol begins when a validator has a candidate value. This procedure can be executed in parallel with the nomination phase. The ballot protocol is divided into three categories, prepare, commit and externalize. The ballot consists of two values (c,x), the counter c and a candidate value x. The counter starts from 1 and is used to order the ballot from smaller to larger ballot numbers by discarding all the previous smaller numbers. When the ballot gets stucked like a validator v3 in the figure 3.23 where the rest of them votes for the ballot (1,x), a new round of ballot procedure begins increasing the counter by one (2,x) like in the second step in the figure 3.23. The stuck can happen when there are some delayed messages in the system or there is not unanimous agreement due to the presence of malicious nodes.

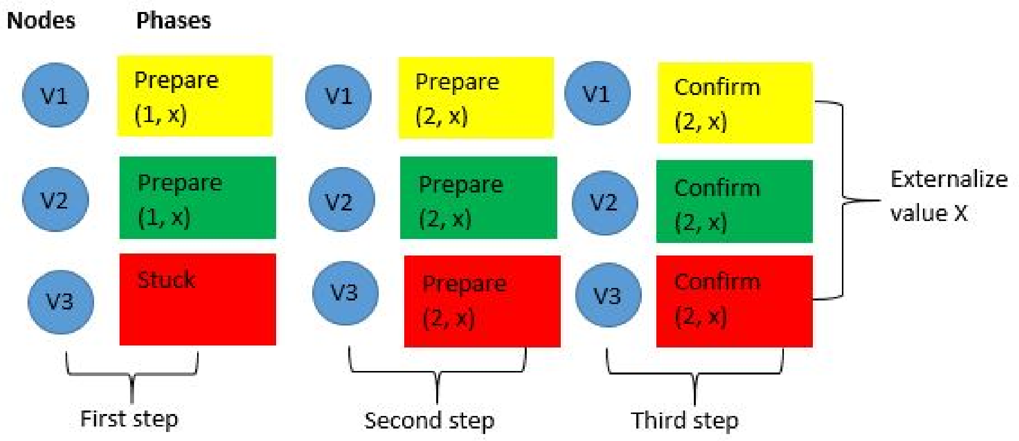


Figure 3.23: Example of ballot protocol

In the prepare phase, each validator assures that it votes for a specific ballot (c,x) and it will never vote in the future for any ballot smaller than (c,x) i.e. if the counter of the ballot is c=2 and ballot value is (2,x) as shown in the second step of figure 3.23, then validator won't vote for smaller number of counter c i.e. ballot value (1,x). Subsequently, validators have to accept the ballot in the prepare phase in order to move to the commit phase as depicted in the third step of figure 3.23. In this case, the validator confirms the pair (2,x) of the ballot and then moves to the last phase which is the externalize. Through the externalize phase, the rest of the validators can be convinced to agree to this value and then to confirm the same ballot by reaching consensus in a faster manner. Once the validators have externalized the same value, then the procedure of consensus algorithm starts from the scratch.

3.3.3 Analysis of consensus protocols

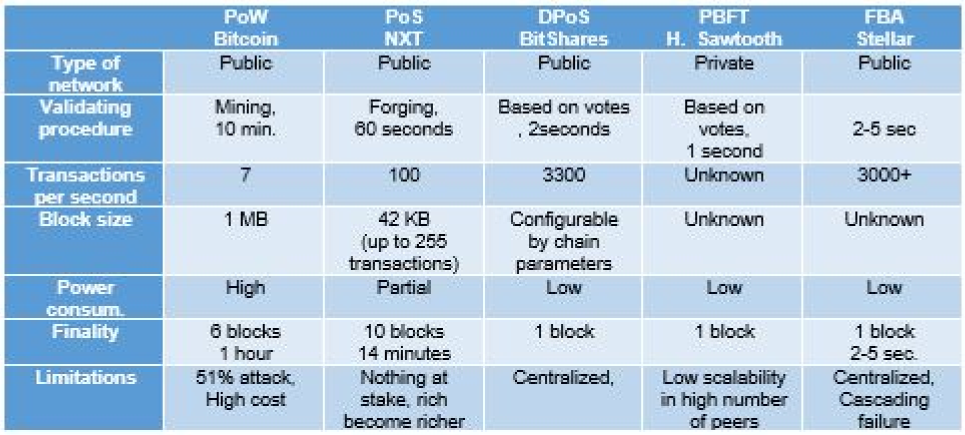
This section and the table 3.1 presents the main characteristics of all the consensus algorithms. Firstly, the first main algorithm known as Proof of Work (PoW) came out by the founder of Bitcoin, Satoshi Nakamoto who introduced this algorithm in the Bitcoin blockchain. The Proof of Work is a consensus algorithm in which the validators validate the transactions and insert them into a block. These validators are called miners and the process of this consensus procedure is known as mining. During the mining process, the miners compete against each other by validating the transactions, inserting them in the block and they try to solve a difficult mathematical puzzle in order to find a solution. When a miner solves the puzzle which is computed every almost 10 minutes, he publishes the block to the network. Once a miner has solved the puzzle and has appended the block in the chain gets as a reward the transaction fees and 12.5 bitcoins. But this amount of bitcoins decreases every four years and when the reward reaches to 0 approximately in 2140, miners will receive only the transaction fees. The generated block except from the transaction list, includes the block header which consists of the hash of the previous block, the nonce, the difficult target etc. The miner tries to solve the mathematical puzzle by modifying only the nonce of the corresponding block in order to prove that the hash of the block header is lower or equal to the difficult target. Although this kind of algorithm gives an incentive to the miners to participate in a honest way to the consensus process by consuming a huge amount of power, it suffers from some limitations. One of the main limitations is the 51% attack where a majority of peers controls the whole network and validates the transactions they want. Another limitation is that PoW suffers from the low throughput as the Bitcoin can validate up to 7 transactions per second. In addition, this algorithm is costly as miners need a huge amount of computational power and expensive equipment to compute the mathematical puzzles. Furthermore, the finality of the block is reached after at least 6 confirmed blocks and the confirmation of each block is executed every one hour. In order to overcome these limitations, other cryptocurrencies like Nxt use other consensus algorithms such as PoS.

The PoS is another algorithm which was used by Nxt cryptocurrency in order to overcome some of the above limitations. The PoS algorithm is completely different from PoW as it doesn't need any computational power to create the block. Unlike the PoW where the miners try to solve mathematical puzzle, the PoS consists of the forgers which validate and create the block. Once the block has been inserted in the chain, the forgers will receive the transaction fees of the corresponding transactions. The forgers are selected in a random way based on the state that they have in their possession. The forgers usually can collect up to 255 transactions with the higher fees in order to insert them in the block and receive these fees as a reward. In addition, in PoS, the way of inserting the block in the chain is much simpler and faster than PoW. The Nxt appends a new block in the chain almost every minute. The block in the PoS is considered immutable after the insertion of 10 blocks. Although the PoS has better characteristics than PoW, this presents also some limitations. The main drawback is the double spent attack where the attacker decides to spend the same amount of money twice. This can be achieved through the forking. Attacker implements two different transactions with the same coins, one in the first chain and the other in the second one. Furthermore, the new forgers as they have not something to loose continue to forge in both chains. When the attacker is given the turn to forge again, he will insert the block in the chain he wants. So, like in the Bitcoin the highest chain wins and both transactions and blocks in the other chain are discarded. Another limitation of PoS is that all participants don't have the chance to be forgers since the selection of them is based only on the stake they hold. More stakes they have, higher probability to be a forger.

Another algorithm which improves the limitations of PoS and PoW is the Delegated Proof of Stake (DPoS). This algorithm is more fair and democratic than the previous algorithm on the grounds that every participant in the network that has stakes can vote in or out for the election of the witnesses and delegates. Witnesses are responsible for validating the transaction and inserting the block in the chain while Delegates are in charge of proposing the changes in the system in order to work properly. According to the Bitshare, every day the top 101 stakeholders with the higher amount of votes become witnesses for the next day. The consensus process of DPoS is executed in rounds. Through this process, DPoS can eliminate the double spent attack as the witnesses are shuffled in each round and they can validate and insert the block in the chain in a specific timeslot and only once per round. So far, the Bitshare can validate up to 3300 transactions per second and generate a new bock every two seconds. The main drawback of this algorithm is the fact that the number of witnesses is too low compared to the members in a public network and this leads to the centralization. Specifically, the public network can consist of millions of peers and through the DPoS and Bitshare, only 101 peers control the whole network.

Although the previous algorithms are implemented in a public network, the PBFT algorithm works in a private network which consists of a specific number of participants. Each one has different rights which are determined by the administrator of the network. This consensus algorithm is based on the message exchanges and can tolerate up to f=(R-1)/3 faulty nodes in order to work properly. The R refers to the total number of peers in the network. The participants in this algorithm are divided by two categories, the primary node and the backup nodes. The primary node is in charge of validating the transactions, inserting them in the block and informing the rest of the peers by sending a Pre prepare message. The backup nodes, in turn, receive this message, check its validity and if it is valid, they send a prepare message in the whole network including itself and the primary node. The next step is the confirmation of each node in the network that has received at least 2f prepare messages from different backup nodes in order to move to final step. This step is known as Commit phase where all the nodes send a Commit message and after that they check that they received 2f+1 commit messages. If the received messages are correct then the block is appended to the chain. The Hyperledger Sawtooth creates a new block every one second and the primary node is changed after the insertion of 100 blocks in the chain. This algorithm suffer from scalability when the number of peers in the network increases. More peers in the network, more time to send and receive the messages.

Nowadays, one of the top blockchain is the Stellar which is implemented through the FBA algorithm. This algorithm works in a public network where all the participants can join or leave the network at any time. As a result, the consensus can not be reached like in PBFT with a majority of the messages (2f+1 correct messages), but through the quorum and quorum slices. A quorum is a group of nodes (quorum slices) which is consisted of at least one quorum slice of each node and this group votes for a unique slot in order to update the ledger. In addition, the quorum slices are subsets of quorums which are responsible to convince particular peers to reach an agreement. Each node can trust which node they want in attempts to be inserted in an existed quorum slice. The consensus round is implemented through the federated voting procedure which consists of three steps, vote, accept and confirm. Each node in the network should vote for a specific block i.e. block a and once it has voted the block, it declares that it will never contradict this block. The next step is the acceptance step where the nodes accept what they have voted if and only if they haven't accept a contradicted state for this specific block i.e. block a. In addition, a node (node 1) which votes for a different block i.e. block b, it can accept block a if at least one node of each quorum slices of node 1 have accepted block a (v-blocking procedure). If all the peers of the quorum accept for this specific block i.e. block a, then the last phase takes part which is the confirmation state. When all the peers of two quorums that intersect confirm this specific block (block a), then the block is appended to the chain and a new consensus procedure begins. When the block is inserted in the chain, it reaches finality. This procedure in Stellar network is executed at approximately every 5 seconds and validators can validate more than 3000 transaction per second. The stellar protocol suffers from some limitations. Although it is implemented in a distributed ledger, researchers have shown that Stellar becomes centralized on the grounds that the Stellar don't give an incentive to the peers to become validators. In addition, another limitation is the cascading failure which is the consequence of the centralization. So, in the case of failure of some validators, then a big part of the system is going to be crashed.



	PoW Bitcoin	PoS NXT	DPoS BitShares	PBFT H. Sawtooth	FBA Stellar
Type of network	Public	Public	Public	Private	Public
Validating procedure	Mining, 10 min.	Forging, 60 seconds	Based on votes , 2seconds	Based on votes, 1 second	2-5 sec
Transactions per second	7	100	3300	Unknown	3000+
Block size	1 MB	42 KB (up to 255 transactions)	Configurable by chain parameters	Unknown	Unknown
Power consum.	High	Partial	Low	Low	Low
Finality	6 blocks 1 hour	10 blocks 14 minutes	1 block	1 block	1 block 2-5 sec.
Limitations	51% attack, High cost	Nothing at stake, rich become richer	Centralized,	Low scalability in high number of peers	Centralized, Cascading failure

Table 3.1: Overview of consensus protocols

4 Improvement of FBA algorithm

4.1 Overview of Stellar and FBA consensus protocol

Nowadays, Stellar is one of the top ten decentralized blockchains that is implemented in the public network. As referred in the previous chapters, the Stellar blockchain can reach consensus for a specific block through the FBA consensus algorithm. This algorithm was developed to overcome the limitations of PBFT, which is used in private networks where the number of peers is limited and known to each other. FBA and Stellar achieve consensus through the quorum and quorum slices. Each node has the right to trust whichever node it wants in order to form a quorum slice. The purpose of quorum slices is to persuade the rest of nodes belonging to these quorum slices to agree on the same decision. Once unanimity has been reached in the quorum, which consists of all the possible quorum slices for each peer in it, and it intersects with another quorum that agrees to the same specific block, then the consensus is reached and the block is appended to the chain.

In 2019 three researchers Minjeong Kim, Yujin Kwon and Yongdae Kim proved that the FBA is not decentralized as it should be but it is significantly a centralized blockchain [16]. As referred previously, the peers in the Stellar are divided in four categories, the watchers, archivers, validators and full validators. In the consensus procedure, only the validators and full validators take part in order to validate the transactions and insert them in the block. Nowadays, the Stellar consists of at about 70 validators and full validators and in [16] they showed that most of them become validators because they are controlled by Stellar organisation or they have business with Stellar. In addition, only a small number of peers that has no profits by the system becomes validator. This happens for two reasons. The first one refers to the fact that nodes lack of incentives to be validators. The other reason refers to the freedom of the peers to choose which node they trust. As a result, a node trusts only the bigger organisation that exists in the system.

The purpose of this proposal is to improve the centralization of the Stellar system by implementing an algorithm where all the peers have the incentive to be validators as to develop a mechanism to measure the reputation of each validator in a democratic way in order to be trusted by the rest of the network. As the Stellar works in a public network consisting of millions of peers, the network should be composed of a huge number of validators in order for the system to be decentralized. As a result, Stellar should give an incentive to the peers in order to be validators.

4.2 Incentive to be validator

In the proposed algorithm, the way how a node can be a validator in Stellar will be analysed. First of all in Stellar, the peer which wishes to implement a transaction has to pay an amount of transaction fees. Every peer has the right to pay only the base fee which is calculated by multiplying the base fee with the number of operations included in each transaction. Furthermore, the peers have the right to pay higher fees to assure that their transactions will be included in the current block or directly in the next blocks. In current version of Stellar protocol, all these transaction fees are gathered in the system and they are distributed once a week through the inflation voting procedure. Every participant in the network can take part in this procedure and can receive the transaction fees as a reward according to the percentage of the votes they have. Our proposal is that these transaction fees should be distributed only to the validators that make an effort in the system by taking part in the consensus procedure. As a result, this is a strong incentive for every peer in attempts to be a validator. Moreover, another incentive for the peers to be validator is the right to implement transactions without the need of paying transaction fees as to prioritise these transactions in the current block.

4.3 Incentive to be full validator

Full validators have a significant role on the normal operation of the system as described so far. Furthermore, in our proposal, the full validator will also have an additional role compared to the role that had in Stellar. In our proposal, full validators will measure the reputation of a specific validator for one day. So, there is a need to give more incentive to the peers in order to be full validators. All the full validators that took part in the reputation measurement will get a small percentage of the whole transactions fees as a reward. For instance, a percentage of the whole transaction fees of each block will be collected in the full fee pool and every day these fees will be distributed among all full validators. According to the distribution of this small percentage of fees, this can be proportional to the number of full validators that took part in the reputation for one specific day divided by the number of the total validators. This percentage can be calculated through the equation 4.1.

$$Percentage\ of\ fees\ in\ \% = \frac{Full\ validators(fv)}{Validators(v)} * 100 \tag{4.1}$$

Where fv the number of full validators that took part in a daily reputation measurement, and v the total number of all validators and full validators that exist in this specific day.

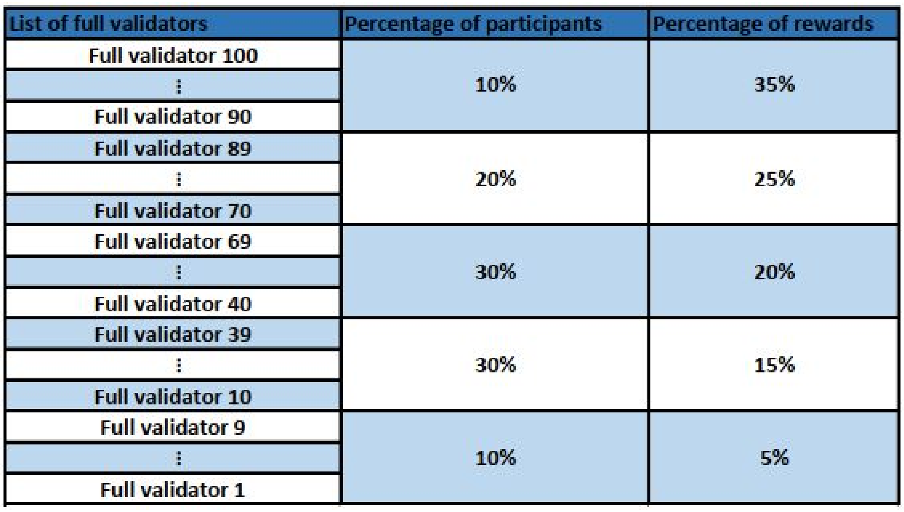
For example, nowadays, there are 70 validators where only 30 of them are full validators. In the case that only 20 full validators took part for the reputation measurement for a specific day, then the percentage of the transaction fees which will be reserved from each block will be 28.57 % according to the formula 4.2.

$$\frac{20(fv)}{70(v)} * 100 = 28.57\% \tag{4.2}$$

All the full validators that will take part in the reputation measurement should send the results to the whole network in the end of the day by signing them with their private key. In addition, the matlab list presents every day the full validators that will take part in the measurements for the reputation in a specific day.

Furthermore, the reward of full validators will be based on the number of execution that full validators have implemented. More specific, the full validators that took part in a daily reputation measurement will not get all the same rewards. There will be full validators which will measure the reputation for only one specific validator and others that do it for more validators. As a result, the rewards will be distributed according to the number of participation of full validation to the reputation measurement. The figure

4.1 depicts an example of the distributions of the fees rewards. Assume that in a specific day only 100 full validators participated. Furthermore, the number of each validator corresponds also to the number of participations in the reputation measurement with 100 the highest number of participations and the 1 the lowest. As a result the first 10 % of the full validators with the highest number of participations will get the 35 % of the fees as a reward and the last 10 % of the full validators will get only the 5% of the fees.



List of full validators	Percentage of participants	Percentage of rewards
Full validator 100	10%	35%
⋮		
Full validator 90		
Full validator 89	20%	25%
⋮		
Full validator 70		
Full validator 69	30%	20%
⋮		
Full validator 40		
Full validator 39	30%	15%
⋮		
Full validator 10		
Full validator 9	10%	5%
⋮		
Full validator 1		

Figure 4.1: Percentage of reward distribution

4.4 Development of reputation mechanism

4.4.1 General explanation of the mechanism

Despite the incentives, there is also a need for the development of a reputation mechanism giving the chance to the validators that do not belong to the big and known organizations to be trusted by other peers in order the system to be decentralized. The measurement of this reputation will be implemented by the full validators. According to the Stellar, being full validator except for the database where the ledger is stored requires, also, an additional internet facing blob store. It is about a cloud space storage where a massive amount of data is stored there such as images, audio, text and etc. As a result, in our proposal, the full validator will record all the results of their measurements.

According to the number of validators, three or more full validators will measure and calculate the reputation for a specific validator per day. More specific, each full validator will count the appropriate measurements of each validator considering four categories and the results of each category will be scaled between 0 and 1. This scale will be achieved by two different ways.

In the first case, the full validators will divide the exchanges messages of a specific full validator by the maximum number of exchanges messages that a validator can send in a specific day. This technique, also, can be used for the measurements of the participation of validators in the consensus rounds. More specific, a full validators will divide the number of consensus rounds that a validator took part in a specific day by the maximum number of consensus rounds that a validator can take part in one day. These examples correspond to the first and the fourth category.

For example, full validator X is responsible for measuring the reputation for validator 1. In the case of exchanges messages, the maximum number of exchanges messages for one day is at about 51840. Assume that validator 1 has sent 10000 exchanges messages, then the the result of the measurement will be equal to 10000 / 51840 = 0.1929. In the case of consensus rounds, the maximum number of consensus rounds that a validator can take part in one day is approximately 17280. Assume that validator 1 took part 3000 times, so the result is equal to 3000 / 17280 = 0.1736.

The case of the third and the fourth category will make use of the min-max normalization technique in order to there will be results scaling from 0 to 1. The expression 4.3 will be used for this technique.

$$Result = \frac{value - min}{max - min} \tag{4.3}$$

Where min is the minimum value for a specific category in the whole network and the max the maximum value in the whole network. For example, in the end of the day the full validator X which calculates the reputation for validator 1 should measure the tokens that validator 1 has as the total amount of the transactions that a validator 1 implements during the day based on the second and third category. So, in the case that validator 1 has 10000 tokens and the minimum amount of tokens that validators have is 2000 tokens and the maximum is 20000 tokens, then the result will be equal to 0.4444 as shown in the expression 4.4. The same procedure is done also for the total amount of the transactions that a validator 1 implements during the day.

$$Result = \frac{10000 - 2000}{20000 - 2000} \tag{4.4}$$

The measurements of each validator will be based on the four following categories:

- In the first category, the full validator will count the exchanges messages of the validator in the federated voting procedure for each consensus round. Correct messages will be considered only the messages from validators that correspond to the block which is inserted to the chain. This doesn't mean that the erroneous messages are malicious messages on the grounds that different quorums through the nomination protocol will try to insert different blocks in the chain. In the Stellar blockchain, a new block is inserted in the chain at about every 5 seconds. As a result, 60 seconds correspond to 12 consensus rounds (cr) and one hour to 720 cr. Moreover, a day consists of approximately 17280 cr. So, in each consensus round, a validator sends three messages (vote, accept and confirm). A validator which takes part in all the consensus rounds and sends only correct messages during the day will send 51840 messages (3*cr). In case of erroneous messages, where validators accept and commit invalid blocks, the counter will not measure the messages.
- The second category will be based on the tokens that a validator has. In the end of the day, the full validator will consider the total amount of tokens that a validator has in its possession.
- The third one will refer to the amount of transactions that a validator implemented. A full validator will calculate the sum of the amount that each validator spent and received during the day.

- In the last category, full validator will count the consensus round that a validator took part. The full validator will count the total number of the consensus rounds that a validator has executed.

The above categories have a different weight on the grounds that each field has different significance for the system needs. The sum of all the weights should be equal to 1 as depicted in the equation 4.5.

$$a + b + c + d = 0.3 + 0.15 + 0.2 + 0.35 = 1 \tag{4.5}$$

- The first category is weighted with 0.30 which is the second highest weight as the exchange of the messages in the consensus round are important. This case has the second significant position and not the first one as different validators can vote or accept different blocks.
- The second category has the lowest weight with 0.15 as the tokens have no so important role on the system.
- The weight of the third category corresponds to 0.20 where the validators through the implementation of transactions indicate to the network that they are active.
- In the fourth category, the number of the consensus rounds outweighs with 0.35 which is the highest weight. Furthermore, this proves to the network that the validators participate fully and they are not idle.

In the end of the day, the full validator will get the average of the results obtained by these categories for a specific validator based on the formula 4.6:

$$fv_j(resi) = \frac{(1stcat * 0.30) + (2ndcat * 0.15) + (3rdcat * 0.20) + (4thcat * 0.35)}{4} \tag{4.6}$$

Where fv_j is the full validator with number j, the 1stcat corresponds to the results of the first category, the 2ndcat to the second categpry and so on.

4.4.2 Measurement of the reputation

According to the reputation procedure, one specific validator which participates in the consensus round will be counted by three different or more full validators for achieving more accurate results. In the end of the day, each full validator will send a message including its own result to the whole network. This message will be signed with full validator's private key and it will include the results, the hash of the results and the name of the corresponding validator. As a result, everyone in the network can crosscheck the results using the public key of full validator.

Besides this, in the end of the day, the three or more full validators which are responsible for the measurements of a specific validator using the formula 4.6, they will calculate, also, the average of the their own results using the expression 4.7:

$$Rep_j(vi) = \frac{fv_x(resvi) + fv_y(resvi) + fv_z(resvi)}{3} \tag{4.7}$$

For instance, if full validators 1, 2 and 3 are responsible to measure the participation of the validator 10, then it will calculate the average number of these per day like in the formula 4.8.

$$Rep_{1,2,3}(v10) = \frac{fv_1(resv10) + fv_2(resv10) + fv_3(resv10)}{3} \tag{4.8}$$

Moreover, the reputation of a specific validator will be calculated through the equation 4.9.

This is the final stage where the three or more full validators responsible for the measurements of a specific validator will, also, calculate the reputation of a specific validator for the next day by combining the results of two days.

$$R_i = w * Rep(vi) + (1 - w) * R_{i-1} \tag{4.9}$$

Where i is the name of the validator, w=0.80 indicates the weight of the day, Rep(vi) is the average of the measurements of a validator with number i and R_{i-1} is the reputation of validator i for the previous day.

In addition, the R_{i-1} indicates actually the history of the whole reputation of a specific validator as the reputation depends on the previous day.

The average result of the $Rep_j(vi)$ should be the same by the three full validators. The reputation result R_i should, also, be the same by the three full validators. In case that the results differ each other, the system should examine the results and it will take the appropriate decisions. For example, in the case that one full validator crashes or goes offline will loose only the rewards of the transactions fees. But in the case of malicious behavior, the system will remove their right to be full validators. In both cases, the system will continue with the rest of the results by the honest full validators.

Finally the system will collect all the results in a list in ascending order and it will publish the list in the network. All the results that occur in the system (i.e. snapshots of the ledger, transactions and etc.) are stored in the archive through the archivers and the full validators. If any node wants to crosscheck the results, it can do it by searching in the archive through the public key of the full validators.

4.4.3 Mechanism of grouping full validators

As referred previously, the full validators will be grouped by three or more in order to measure the participation of each validator by using the expression 4.10.

$$c = \binom{fv}{k} = \binom{30}{3} = \frac{30!}{3!(30-3)!} = 4060 \tag{4.10}$$

where the fv is the number of full validators k is the number of the full validator's group

Nowadays, there are only 30 full validators and according to the above expression, there are 4060 different combinations to measure only 70 validators. So, only one combination of a group will be responsible for a specific validator. This group will be obtained by the equation 4.11.

$$cv_i \ = sn \ mod \ c \tag{4.11}$$

Where i is the name of the current validator

sn the sequence number of the validator

and c is the number of available combinations

As explained in the previous chapters, the node should create an account in order to participate in the Stellar network. Once each validator has created its account, it will be given a specific sequence number. The sequence number of a validator which takes part in a consensus procedure increases by one every consensus round.

The result of cv_i equation will correspond to the number of the list in the Matlab table. This list is distributed in an ascending order and it includes the possible groups of three or more full validators. This list is calculated by the command in the Matlab which is nchoosek (n=30, k=3). The whole procedure is explained analytically in the rest of this chapter.

For example, validator v5 has a sequence number sn= 5 and the result of cv_5 through the equation 4.12 is cv_5=5 . The first day there are only 30 full validators that correspond to 4060 different combinations. As it is depicted in the figure 4.2 which is obtained by the Matlab, the number 5 corresponds to 1, 2, 7 full validators.

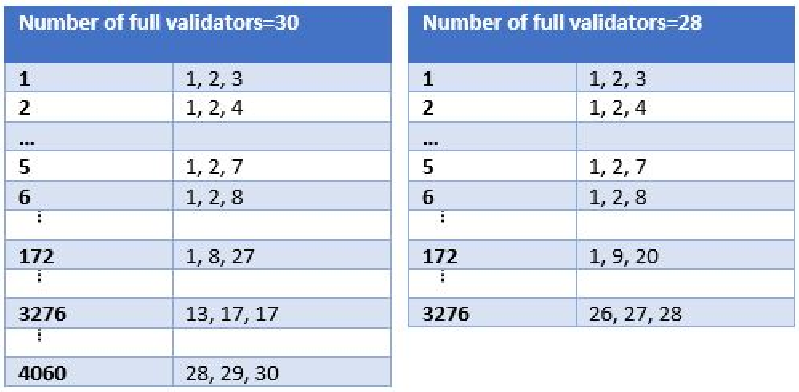
$$cv_5 = 5 mod 4060 = 5 \tag{4.12}$$

The next time that a validator v5 takes part, its sequence number will be equal to 6. Supposing that the validator v5 has participated only once in the first day and it will participate for consensus in the second day. This second day consists of 28 full validators, as a result there will be c = 3276 combinations. And the result of cv_5 now is equal to 6 through the expression 4.13 . According to the figure 4.2 this number corresponds to 1, 2, 8 full validators.

$$cv_5 = 6 mod 3276 = 6 \tag{4.13}$$

After a long time, the sequence number of v5 will be equal to sn=10000 and there will be 28 full validators for this specific day. So, the result of the expression 4.14 for cv_5 will be equal to 172 and 1, 9 and 20 full validators will be responsible for the measurement of v5 validator.

$$cv_5 = 10000 mod 3276 = 172 \tag{4.14}$$



Number of full validators=30	
1	1, 2, 3
2	1, 2, 4
...	
5	1, 2, 7
6	1, 2, 8
⋮	
172	1, 8, 27
⋮	
3276	13, 17, 17
⋮	
4060	28, 29, 30

Number of full validators=28	
1	1, 2, 3
2	1, 2, 4
...	
5	1, 2, 7
6	1, 2, 8
⋮	
172	1, 9, 20
⋮	
3276	26, 27, 28

Figure 4.2: Procedure of selecting candidates

The list of the possible combinations will change every time according to the overall number and the name of the full validators that will be able to take part in the consensus procedure during each week.

5 Conclusion

Overall, it has been analyzed how powerful can be a blockchain leading people to make their life much easier than before. In the past, people were implementing different kinds of transactions through a central authority. For example, someone who executed a transaction through the banks had to wait some days for the implementation and verification of its transaction as well as to pay some fees. Nowadays, the blockchain is widely used in many fields such as financial, healthcare and so on in order to eliminate this cost and time of the implementation of the transactions.

So, blockchain provides a trusted and more secure network where the peers interact directly each other by making transactions without the need of third parties. In this work, it has been explained how the trust and the security can be achieved through the consensus algorithms. Compared to all the consensus algorithms that were analyzed in this thesis, it seems that SCP is one of the most popular and effective algorithms at the moment. In a SCP protocol, the peers have the freedom to choose whatever peer they trust to validate the transactions through the quorums and quorum slices. In addition, SCP is one of the most fast consensus protocols in public networks for the validation of the transaction and the insertion of the block in the chain compared to other consensus algorithms.

As it was explained, the Stellar is implemented in a public network and it is based on quorums and quorum slices. When at least two quorums intersect and there is an unanimity on agreeing for a specific block between all the peers that belong to these quorums, then the consensus is achieved and the block is appended to the chain. Furthermore, the peers have the freedom to choose which peer they trust in the network. In that way, it was proved by some researchers [16] that the peers that enter the network trust only those ones which belong to big organisations leading the network to be centralized, which is the main limitation of the SCP protocol. According to our proposal, this drawback should be overcome by giving more incentives to the peers to be validators and full validators. Furthermore, through the reputation mechanism,the peers will trust and follow peers in the network that they don't belong only to the big organizations. As a result, the trust and the decentralization of the network increases.

A future improvement of this thesis could be a practical implementation of our theoretical proposed algorithm to check if the reputation mechanism works well based on the analysis that we implemented in our thesis.

Bibliography

[1] Andreas M. Antonopoulos. Mastering bitcoin. *Book*, June 2017.

[2] Dr. Arati Baliga. Understanding blockchain consensus models. *Book*, April 2017.

[3] Charts bitcoin. Blockchain size. `https://charts.bitcoin.com/btc/chart/blockchain-size#5ma4`, August 2019.

[4] Bitinfocharts. Cryptocurrency statistics. `https://bitinfocharts.com`.

[5] BitShares. White paper of bitshares. `https://bitshares.org/technology/delegated-proof-of-stake-consensus/`.

[6] Vitalik Buterin. On public and private blockchains. `https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/`, Aug 2015. Accessed on 6 August 2015.

[7] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. *Article*, February 1999.

[8] Miguel Castro and Barbara Liskov. Authenticated byzantine fault tolerance without public-key cryptography. *Article*, June 1999.

[9] KONSTANTINOS CHRISTIDIS and MICHAEL DEVETSIKIOTIS. Blockchains and smart contracts for the internet of things. *Article*, 3 June 2016.

[10] D. Mazieres G. Losa and E. Gafni. Simplified scp. `http://www.scs.stanford.edu/~dm/blog/simplified-scp.html`, March, 2019.

[11] J. McCaleb G. Losa, D. Mazieres and S. Polu. The stellar consensus protocol (scp). `https://tools.ietf.org/id/draft-mazieres-dinrg-scp-05.html`, November 4, 2018.

[12] Elli Androulaki Ghassan O. Karame and Srdjan Capkun. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. *Article*, 2012.

[13] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Article*, 1991.

[14] ROBERT SHOSTAK LESLIE LAMPORT and MARSHALL PEASE. The byzantine generals problem. *Article*, 1982.

[15] DAVID MAZIERES. The stellar consensus protocol: A federated model for internet-level consensus. *WhitePaper*, February 25, 2016.

[16] Yujin Kwon Minjeong Kim and Yongdae Kim. Is stellar as secure as you think? *Article*, April 29, 2019.

[17] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Article*, 2008.

[18] Giang-Truong Nguyen and Kyungbaek Kim. A survey about consensus algorithms used in blockchain. *Article*, February 2018.

[19] Nt. White paper of nxt. `https://nxtwiki.org/wiki/Whitepaper:Nxt`, Feb 2016. Accessed on 7 February 2016.

[20] Peercoin. White paper of peercoin. `https://docs.peercoin.net/`, Dec 1988. Accessed on 2012-11-11.

[21] Waves platform. Leased proof of stake (lpos). `https://docs.wavesplatform.com/en/blockchain/waves-protocol/leased-proof-of-stake-lpos.html`.

[22] Hyperledger Sawtooth. White paper. `https://sawtooth.hyperledger.org/docs/pbft/nightly/master/configuring-pbft.html`, 2018.

[23] Hyperledger Sawtooth. White paper. `https://sawtooth.hyperledger.org/docs/pbft/nightly/master/architecture.html#message-types`, 2018.

[24] Stellar. Stellar lumens. `https://www.stellar.org/lumens/`, 2019.

[25] Stellar. Dashboard of stellar. `https://dashboard.stellar.org/`, August 6, 2019.

[26] W. Scott Stornetta Stuart Haber and Dave Bayer. Improving the efficiency and reliability of digital time-stamping. *Article*, March 1992.

[27] Florian Tschorsch and Bjorn Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *Article*, 2015.

[28] Wikipedia. Smart contract. `https://en.wikipedia.org/wiki/Smart_contract`, December 2016.

[29] Wikipedia. Blockchain. `https://en.wikipedia.org/wiki/Blockchain#Types_of_blockchains`, April 2019.

[30] Hongning Dai Xiangping Chen Zibin Zheng, Shaoan Xie and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. *Article*, 201.