Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Mohamed Khider - BISKRA
Faculty of Exact Sciences, Natural and Life Sciences
Computer Science Department

Order Number: GLSD01/M2/2022

Thesis

Presented to obtain the diploma of academic Master in

Computer Science

Option: **Software Engineering and Distributed Systems**

Modelling and verification of a blockchain consensus protocol

---

By:
Abdelnaim Dikki

Defended the June 26th 2022, in front of the jury composed of :

Bennoui Hammadi	Prof	President
Hmidi Zohra	MAA	Supervisor
Ramdani Mohamed	MCB	Examiner

University Year: 2021/2022

Acknowledgement

First things first, I want to thank Allah the almighty, who gave me the strength of will to reach the end of this project and finish it.
I would like to thank MRs. Hmidi Zohra for helping and being there for me whenever I needed guidance, and for that you have my deepest gratitude.
Furthermore, I would like to thank all the teachers of the software engineering and distributed systems and all the teachers of all the department in general for their hard work, and the fact they didn't skimp us of any of their experience.

Dedication

*This work is dedicated for my loving **mother** and **father** who have always supported me, and provided only the best of things for me unconditionally, and trusted whatever decisions I made as if its their own.*
My deepest gratitude to my friends, who I call brothers, my classmates, and to everyone who I met on my journey here.

Abstract

Blockchains have lately found a use for themselves in practically every field, either in finance, health or data management and many more. This spread is due to their impeccable ability to store data securely and prevent any tampering or attempts of changing its content, also due to their decentralized nature where no central authority gets to make the definitive decision.

To better manage and oversight these blockchains, many consensus protocols have been created. Each protocol enforcing security and integrity in the blockchain in its own way, and its own rules. All of this is to make sure that the sensitive data within the blockchain is highly secured and to eliminate any attempt of fraud or malicious behaviour.

Therefore, this project aims to model, verify, and evaluate one of these consensus protocols in particular which is the proof of work protocol, make a model for it and test some property's satisfaction, which we will do after going through blockchains and other consensus protocols briefly.

Key works: Blockchain, Consensus protocol, Consensus algorithm, Proof of work, Formal verification, Modeling, Model checking, Uppaal, Temporal logic.

Résumé

Les blockchains ont récemment trouvé une utilité dans pratiquement tous les domaines, que ce soit dans la finance, la santé ou la gestion des données, et bien d'autres encore. Cette propagation est due à leur capacité impeccable à stocker des données en toute sécurité, et à empêcher toute falsification ou tentative de modification de leur contenu, ainsi qu'en raison de leur nature décentralisée où aucune autorité centrale ne peut prendre de décision définitive.

Pour mieux gérer et superviser ces blockchains, de nombreux protocoles de consensus ont été crées. Chaque protocole renforce la sécurité et l'intégrité de la blockchain à sa manière et selon ses propres règles, tout cela pour s'assurer que les données sensibles de la blockchain sont hautement sécurisées et pour éliminer toute tentative de fraude ou de comportement malveillant.

Par conséquent, ce projet vise à modéliser, vérifier et évaluer l'un de ces protocoles de consensus en particulier, le protocole de preuve de travail, en faire un modèle et tester la satisfaction de certaines propriétés, ce que nous ferons après avoir parcouru brièvement les blockchains et les autres protocoles de consensus.

Mots clés: Blockchain, Protocole de consensus, Algorithme de consensus, Preuve de travail, Verification formelle, Modélisation, Vérification de modèle, Uppaal, Logique temporelle.

Contents

List of Figures

List of Tables

General Introduction

General Introduction

One of the leading fields in the present time is the Blockchain technology. The sudden interest in it comes from its immutable ledgers which are capable of providing platforms for any entity with autonomously data driven needs. In 2008 Satoshi Nakamoto proposed using blockchain as the backbone of a new decentralized cryptocurrency called Bitcoin [2]. By combing the technology of distributed systems which is used by blockchain, the peer to peer network, and cryptography, what better framework for a cryptocurrency where the data of the transactions in the form of digital tokens made between peer to peer users cannot be tampered with, and providing data transparency while maintaining security of the system.

To ensure the integrity and the correct behaviour of these distributed systems, a pseudo authority needs to be present. Therefore consensus protocols were introduced, making blockchains the go-to when it comes to transaction-driven resource management in communication networks and distributed autonomous systems, and trusted virtual computers on a decentralized network.

In a nutshell blockchain technology has earned the name of a game changer hailed by both the industry and the academic field in the decentralization of digital infrastructures in financial industry and expanding to a multitude of domains.

Insuring the good functioning of any blockchain system means insuring that the consensus protocol is working to its fullest potential and that it satisfies the awaited standards which is an indispensable task.

Quality and performance requirements can be met by formal methods seeing as they have a solid mathematical background to develop proofs of correctness and absence of failure as well as to develop computer tools that facilitate verification tasks. Among these, there are the Automata and their variants which are good candidates for modeling and verification of distributed systems.

One of the tools to do such verification is the UPPAAL tool which we will be using for our project and the objectives of this work are :

- Do a formal modeling of the PoW protocol with timed automata.
- make a qualitative and quantitative verification of this protocol though the Uppaal tool.

Therefor, this thesis is organized as follows :

Chapter1 is devoted to the state of the art on blockchain. It presents blockchain networks types, nodes, data organisation, applications and consensus protocols.

Chapter2 concentrates on proof of work protocol, its operating principle, block building and main problems.

Chapter3 describes our contribution. First it presents the modeling of PoW protocol with timed automata, then the verification of important properties and finally the evaluation of performance metrics using model checking.

The thesis ends with a conclusion which evaluates the results, and discusses some perspectives.

Chapter 1

Blockchain and consensus protocols

1.1 Introduction

Bringing a solution to the fickle nature of trust problem, emerged a new technology first proposed for the famous cryptocurrency Bitcoin, bringing trust to a decentralized system, without needing to know the identity of the entities within it, this technology is called the blockchain.

Blockchains first came to fame in October 2008, in the hopes of replacing banks with money circulating on a peer to peer network. It stirred up quite the noise especially in the market of finance, and being highly sensitive, it needed some governing rules or as its called consensus protocols to maintain integrity and security within the blockchain [9].

In this chapter, we will be reviewing the blockchain, giving its definition, its network, and its types and components, and after we will go to consensus protocols, go through some examples and dissect them thoroughly [24].

1.2 Blockchain

1.2.1 Definition

Blockchains are generally associated with the cryptocurrency term, it is a secured database of record of transfers and transactions which involves unique instances of values where all these operations are distributed, validated and maintained by a network of separate computers without requiring a third-party to intermediate such as a bank or a government, so the lack of a single person in control and supervision by a large community makes it almost impossible to change any previous records or delete the history of a transaction transaction all thanks to blockchain's build in distributed nature, blockchain also allows computers on the network to access each other's entries which makes it impossible for one central entity to attain control of the network.

When a transaction is made, it goes to the network to be determined as authentic or not. Once the transaction is authenticated it is linked with the previous one forming a chain of transactions, this chain is called the blockchain [33].

Blockchain technology operates on a peer to peer network, seeing as it is based on a decentralized network. The blockchain could handle several types of informations such as identity credentials, money, or contracts of property, but when Satoshi Nakamoto came up with the digital cryptocurrency bitcoin it took the blockchain technology society by surprise [2].

It combining cryptography, peer to peer network, and distributed system technology that the blockchain uses, it provided a secure framework to the cryptocurrencies, where the transactions could never be tempered with thanks to the immutable data organization which are the

blockchains, and and the network of the blockchain that has a set of approaches on how data should be deployed and maintained [21].

1.2.2 Network of the blockchain

In order to guarantee a swift organisation of the blockchain and the entities and nodes within, identities must be managed especially if the network is public, where any blockchain user can simply join this blockchain network and start working on it [19], these users are called nodes who can be working on the same physical machine but different identities, so a node doesn't reference the machine but the a single user [18].

The nodes are organised on a peer to peer network, on the other hand on a non public network meaning a node needs permission to participate in the consensus process or propagating data. The main reason for a blockchain network protocol is to have a topology that is randomly fair for all nodes to make the function on the blockchain more efficient and to replicate the blockchain on all nodes leaning synchronisation and replication.

Blockchains use mostly a peer to peer network with a ready to use function and make slight changes in the way data is communicated between nodes and the topology on the network. the participating nodes in the blockchain network and particularly in bitcoin gather a list of DNS servers to get the IP addresses of the new nodes entering the network for the participating nodes lists to be initialized, and based on these lists they can request addresses.

If a node starts to send malformed messages its IP address will be banned and is considered malicious through a mechanism called the penalty score where a faulty node has a high penalty score. For the synchronisation and replication of the blockchain across all nodes the transaction's messages and the mined blocks are broadcasted through the links of the peer to peer network in a gossip-like way. A peer to peer link in the network of blockchain consists of a persistent TCP connection protocol with a handshake of level three, taking each node's state and exchanging it's replica, once the connection betwwen nodes is established transactions or blocks could be exchanged.

In a permissionless blockchain network The role of each node is not specified, but they can be put to categories such as a light node, nodes of consensus or full function node,but to do that all nodes need to enable the function to route messages in order to verify or do a maintenance to the established connection [21]. Figure 1.1 shows the blockchain peer to peer network.

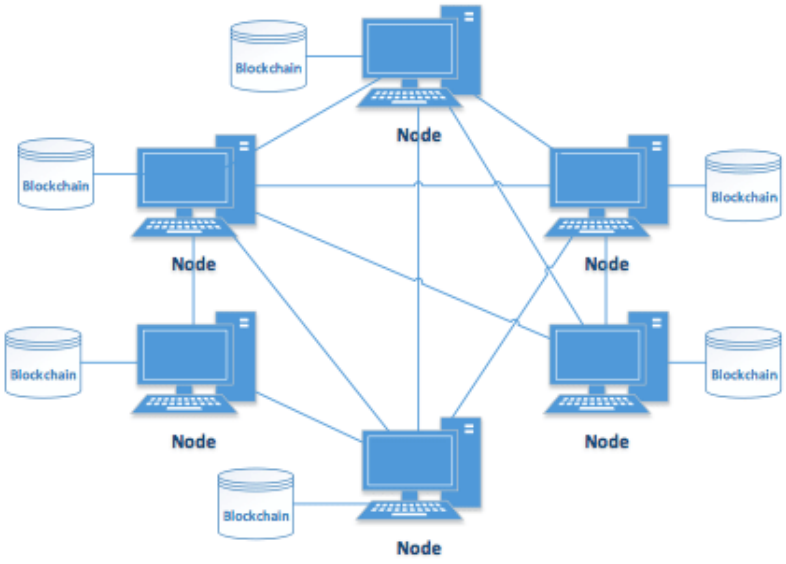

Figure 1.1: P2P blockchain network

1.2.3 Blockchain Networks types

Model of the Permission in blockchain networks determines the type of the network that the blockchain is running on, and categorizes the function of nodes, for example who can publish new found blocks, who can participate in the consensus process.

If publishing new blocks can be done by any participating node we are talking about permissionless blockchain networks, if the publishing can only be done by certain users, then the type of the network is permissioned, in small words to start, a permissioned network of blockchain is like an internet of a corporation that is controlled and secured for the purpose that not anyone can alter and certain functionalities are disabled for certain users, while a permissionless network of blockchain is like a public internet that is open for every available user to partake[34]. Figure 1.2 shows the layout of a public vs a private blockchain network [25].

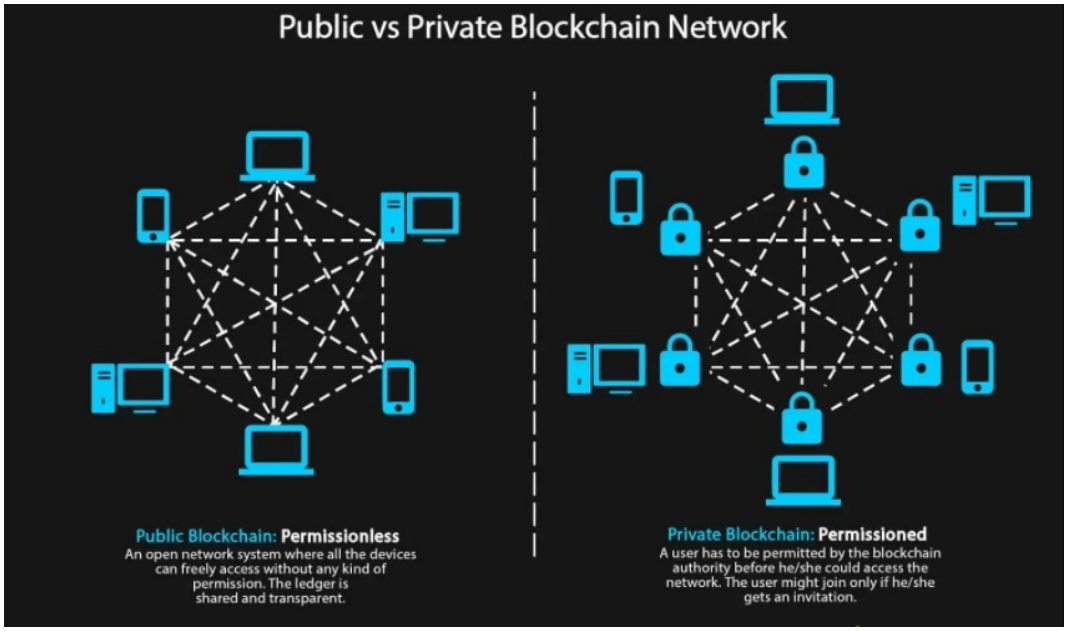

Figure 1.2: Public and private blockchain networks

1. Permissionless Blockchain Networks

   Blockchain networks that are permissionless are platforms open for any individual with a ledger that is decentralized and anyone can partake in the blocks publishing process without the need to get permission from any other governing authority. They come often as software with open sourcing, available for anyone to freely download them and in these types of blockchain networks any node has the functionality to read the the current blockchain since they already have the property to publish blocks, and also transaction issuing and including within the blocks [33].

   A user inside the blockchain network with a permissionless blockchain has the ledger at its disposal to read and write on it, since it's open to every user, other parties that are malicious may try to publish new blocks in a manner that undermine the integrity and security of the blockchain, and to avoid getting in such predicaments a multi participant consensus system has been utilized to either hold or expand their resources when publishing a new block, such mechanisms include Consensus model of Proof of Work and Proof of Stake. Permissionless blockchain networks usually utilize the reward incentive when finding new blocks with some of the native currency to promote for a behavior that is non-malicious [34][27].

2. Permissioned Blockchain Networks

Blockchain networks that are permissioned set a term that a node must be authorized in order to publish new blocks, for the reason that the blockchain is being maintained by authorized individuals, either by restricting the read only access or restricting which nodes get to issue transactions, in simple terms permissioned blockchains could have anyone be able to get a copy of the current blockchain or restrict it to a certain degree for authorized nodes with less suspicious intents [33].

These Blockchain networks that require permission could be maintained and instantiated with a closed software source or an open one.

Permissioned blockchain networks also employ consensus mechanisms in the publishing blocks process, but these mechanisms are less needed to require such immense expanses and maintenance in the resources department, contrary to permissionless ones, this is due to the fact that if an individual wants to participate in the permissioned blockchain network he needs to establish his identity before joining.

Less security threats and more trust result in needing only a lighter versions of consensus protocols which leads to them being more fast and less expansive when it comes to computational power. These type of blockchain networks are more used by tight control and protective corporations and organisations that need a certain level of discreet in their blockchain [33].

Permissioned blockchains could be used also by corporations that is doing business with one another but less trustful of one another, so they use the permissioned blockchain networks where every transaction is recorded and only authorized people who are well identified are present can issue them, on a ledger with a shared distribution [25].

The consensus model that is used in these situations is determined based on the level of trust between participating organizations. Beyond the problem of trust permissioned networks offer a more transparent and insightful way to avoid any misbehaving and have accountability. Revealing the user's transactions and it's information exists in some Blockchain networks with permission needing only the credentials of this user or his identity, providing a level of privacy [15].

In a few permissioned blockchain networks there is no anonymity or pseudo-anonymity meaning all users are required to be authorized to be able to receive or send transactions [33]. In these systems if individuals while incorporated in organizations try to commit fraudulent business or achieve a shared process of malicious intent can be instantly identified and the legal consequences of such behavior is well known and actions are established to pursue them judicially [34].

Permissioned Blockchain vs Permissionless Blockchain		
Category	**Permissioned**	**Permissionless**
Speed	Faster	Slower
Privacy	Private membership	Transparent and open - anyone can become a member
Legitimacy	Legal	Allegal
Ownership	Managed by a group of nodes pre-defined	Public ownership - no one owns the network
Decentralization	Partially decentralized	Truly decentralized
Cost	Cost-effective	Not so cost-effective
Security	Less secure	More secure

Figure 1.3: Comparison between permissioned permissionless blockchains

1.2.4 Blockchain Network Nodes

A node is an entity using its device on the blockchain network, it is considered as a sort of stake holder depending on which protocol is being used on network, it keeps track of the ledger that is distributed among other nodes, it also does other tasks such as serving as a hub for communication.
the main functionality of a node is to confirm and valid whether a batch of transactions on the blockchain network is worth being legalised or not, meaning if they are worthy of being put in a block. Each node has a unique identifier which permits to distinguish easily one node from other different nodes on the blockchain network.

1.2.5 Types of Blockchain Network Nodes

- Light node
  Stores only the block's header in its local storage.
  An example is the Wallet of a client.

- Full function node
  Stores a complete and up to date replication of the current blockchain, and is able to do a verification of transactions without referencing externally.

- Node of Consensus
  Participates in the consensus process and mining blocks process and influences the state of the current blockchain, also it can either go the Light node storage mode or the Full function node storage mode.
  A consensus node is known also as the Miners or Mining nodes. Figure 1.4 illustrates roles of nodes in a permissionless blockchain network [13][10].

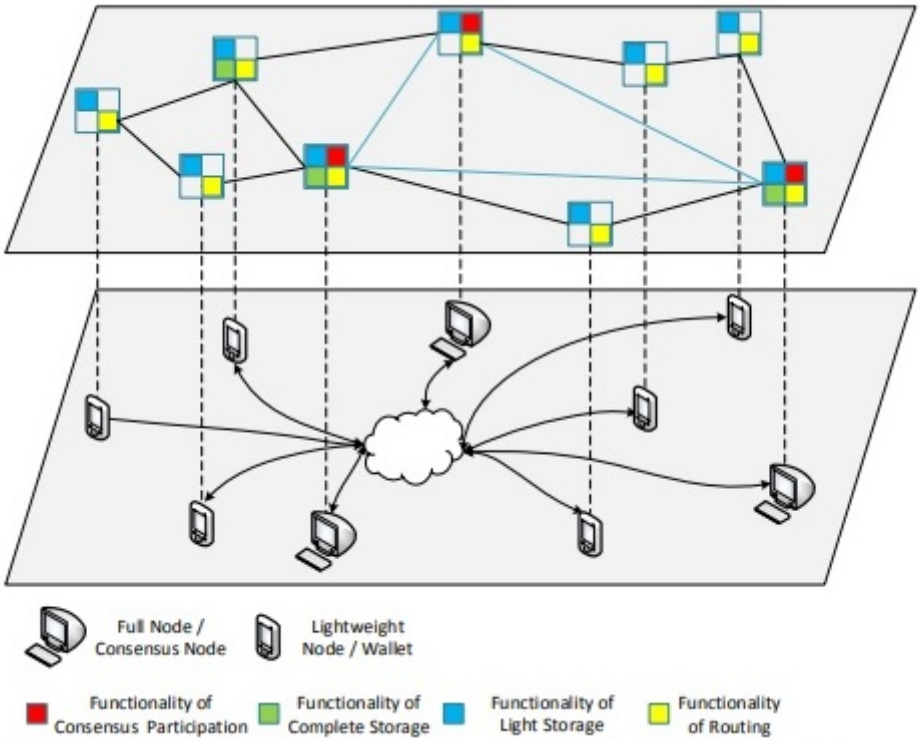

Figure 1.4: Illustration of roles of nodes in a permissionless blockchain network with P2P links between consensus nodes shown in blue

1.2.6 Data Organisation

The blockchain can be viewed as a structure of infinitely growing string that the nodes on the blockchain network have all agreed upon. In regards to the organisation of data there are three levels of hierarchical data structures which are first of all the blocks, second the transactions inside the blocks and third the chain of blocks. These three levels each need a cryptographic set of functionalities that are required when dealing with data in regards to its integrity and protecting its authenticity.

1. Transactions:
   A blockchain's atomic data structure is the transactions inside of it, in almost all cases, a transaction is made by a collection of users or autonomous objects (i.e., smart contracts) to point the transfer of tokens from the senders to the desired receivers. it also specifies list of inputs that associates the values of the tokens with the identities or adresses of the sending entities. It also specifies a non-empty list of outputs designating the redistribution results of the input tokens among the associated identities of the receivers.

   A transaction could be considered as a static record where the sender's and the receiver's identites are kept unchangeable, also the token value that is redistributed and therefore the token reception's state, to guard the authenticity of a transaction record, the functionalities of cryptographic hashing and asymmetric encryption are activated.

2. Hash Function:
   A cryptographic hash function maps indiscriminately an arbitrary-length binary input to a singular, fixed-length binary output (i.e., image). With a secure hash function (e.g., SHA-256), it's computationally very improbable to recover the input from the output image. Also, the probability to come up with the identical output for any two different inputs is negligible. Asymmetric Key: Each node within the blockchain network generates a pair of personal and public keys. The private secret's related to a digital signature function, which outputs a fixed length signature string for any arbitrary length input message. the general public secret's related to a verification function, which takes as input the identical message and also the acclaimed signature for that message.

   The verification function only returns true when the signature is generated by the signature function with the corresponding private key and also the input message. The nodes of the network or any other autonomous objects reveal their public keys as an identification method, meaning they reveal the hash code of their public keys, as their permanent addresses (also called their pseudo-identities) on the blockchain. Since each input tuple in a very transaction is signed by the associated sending account, the network is ready to publicly validate the authenticity of the input through verifying the signature supported the sender's public address [9].

3. The Block:
   A block could be seen as a container of an arbitrary subset of records about past transactions where the later could only be created by a participating node in the process of consensus.

   A block also has some other arrangements to to keep the transactions safe from alterations and record manipulation and also to keep track of the order of past blocks which is called a hash pointer of a block which is kept within the structure of the data of a block, which organises the blocks and their predecessors[9].

   The genesis block is a block recognised by every blockchain which is the beginning meaning it has no reference and no predecessor and it is the ancestor of all the blocks

that comes after in the chain. Figure 1.5 shows how each block is linked to its previous block number

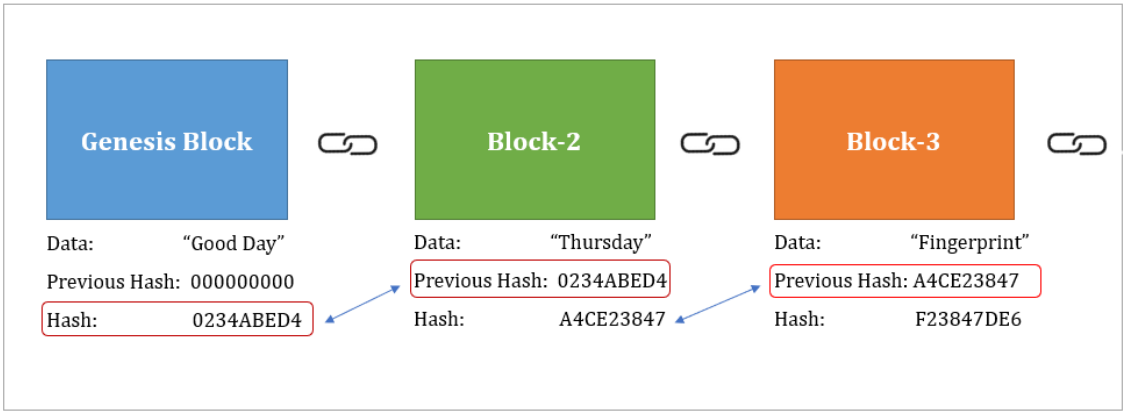

Figure 1.5: Chain of blocks

4. The Block Hash Pointer:
   A block hash is the hashcode of the data that is stored in a block. As shown in Figure 1.6 a hash pointer is the hashcode that represents current block , we find this pointer at the header of each block. The reference blocks store their hashes as a hash pointer of a block, so so when we are doing a local view, the transactions inside a block could be attached to a reference block, meaning they were created earlier on than the transactions that are stored in the latest block [23][12].

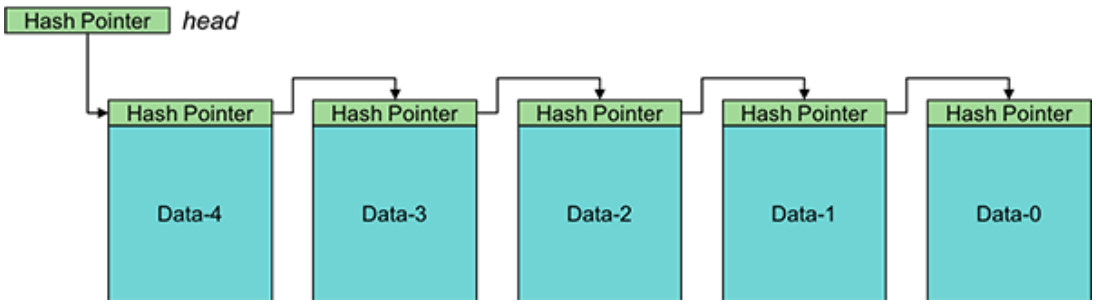

Figure 1.6: Hash pointers

5. The merkle tree:
   Simply put, meaning representing transactions in the form of a binary tree, as shown in Figure 1.7, where each transaction is represented by a leaf with the hashcode of said transaction, and the nodes with no leaf is represented by the hashcode of the child nodes. The merkle root represents the root node, and a block that only stores a transaction's merkle root is considered to be in a form that is light, which helps when it comes to synchronising and validating in a rapid way [23][12].

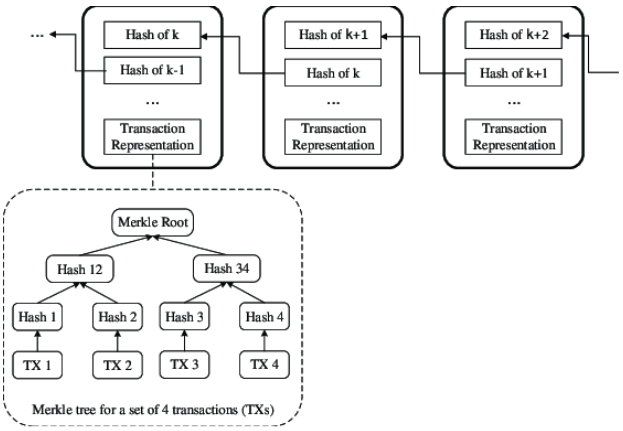

Figure 1.7: Illustration of a chain of blocks where transactions in a single block are represented by a merkle tree

1.3 Blockchain Applications and Future Uses

1.3.1 Applications

Type	Application	Description	Examples
Financial applications	Crypto-currencies	Networks and mediums of exchange using cryptography to secure transactions	Bitcoin Litecoin Ripple Monero
	Securities issuance, trading and settlement	Companies going public issue shares directly and without a bank syndicate. Private, less liquid shares can be traded in a blockchain-based secondary market. First projects try to tackle securities settlement	NASDAQ private equity Medici Blockstream Coinsetter
	Insurance	Properties (e.g., real estate, automobiles, etc.) might be registered using the blockchain technology. Insurers can check the transaction history	Everledger
Non-financial applications	Notary public	Central authorization by notary is not necessary anymore	Stampery Viacoin Ascribe
	Music industry	Determining music royalties and managing music rights ownership	Imogen heap
	Decentralized proof of existence of documents	Storing and validating the signature and timestamp of a document using blockchain	www.proofofexistence.com
	Decentralized storage	Sharing documents without the need of a third party by using a peer-to-peer distributed cloud storage platform	Storj
	Decentralized internet of things	The blockchain reliably stores the communication of smart devices within the internet of things	Filament ADEPT (developed by IBM and Samsung)
	Anti-counterfeit solutions	Authenticity of products is verified by the blockchain network consisting of all market participants in electronic commerce (producers, merchants, marketplaces)	Blockverify
	Internet applications	Instead of governments and corporations, Domain Name Servers (DNS) are controlled by every user in a decentralized way	Namecoin

Figure 1.8: Applications of blockchain

As seen in the Figure 1.8 above, there are two distinguishable applications where the blockchain could prove useful. This glamorous innovation has not only the potential to change the interaction's nature in the field of financial business. Surely the most noticeable feat where blockchain is used is in the crypto-currency field, such as Bitcoin, Ethereum, Tether, Binance

Coin, USD Coin, Terra and Solana and many others, but it could also be used in solving daily life problems.One of the biggest noticeable fields is the health care field. Figure 1.9 shows an example of how data is handled [3][11].

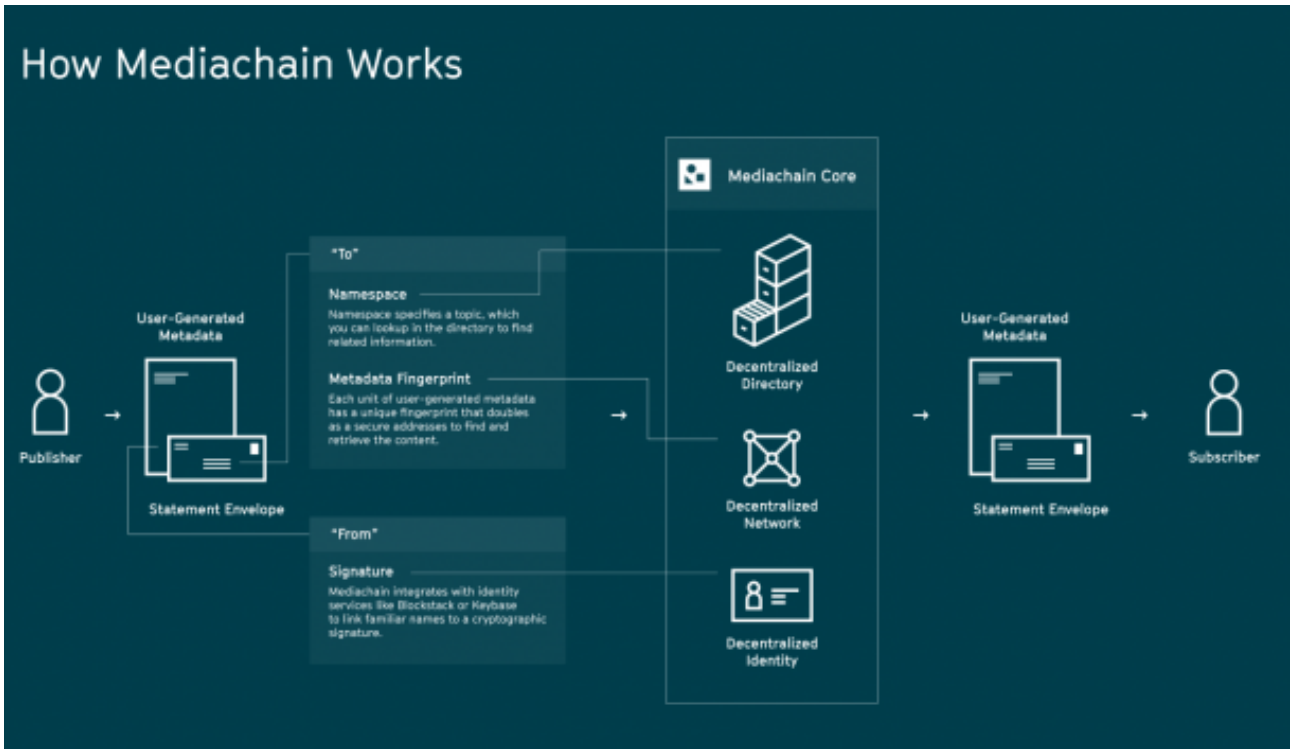

Figure 1.9: Mediachain blockchain applications

Researchers are now using the blockchain ledger to securely transfer the medical records of patients which facilitates uncovering the genetic code of these patients, supply chain management for prescripted drugs, genomic projects management, protecting health data and easy digital tracking of any outbreaks or issues. A few more examples are in Figure 1.10 [14].

Figure 1.10: Blockchain in the health department

Other applications of blockchain right now are [25]:

- Voting mechanisms.
- Advertising.

- Platform for real estate processing.
- Creation of original content.
- Anti-money laundering tracking system
- Supply chain and logistics monitoring
- Personal identity security.
- Music royalties tracking.
- NFT marketplaces.

1.3.2 Future uses

The limitless applications of blockchain seems to be branching even more, for the most part in fields that over the past years used third parties to enforce a certain level of trust, this may have the consequence of reconstructing both politics and the whole world by the blockchain technology. Using decentralized platforms may render certain functions obsolete if people started organizing and protecting society with it, as mentioned by Atzori in 2015 that "decentralization of government services through permissioned blockchains is possible and desirable, since it can significantly increase public administration functionality [38].

Using this technology would efficiently help in the protection of wealth which would help the third world countries to reorganize and enforce order to their societies, for example proving ownership of land would be an easy job in the case of government wrongfully seizing the lands from their rightful owners all these existential problems could be resolved using the integration of land titles into blockchain which would make them inerasable, however the weak link might be in the interface linking the physical and the digital realm where the digital trust which the blockchain offers would be damaged as mentioned by Glaser in 2017. Researchers are also debating whether cryptocurrencies that rely on blockchain systems could in reality replace real physical money as mentioned by both the European Central Bank in 2012, the argument for this saying is that money is defined as anything that permits doing payments and is accepted by society whether to buy goods or purchasing services, but cryptocurrencies are generally used as a method of exchange and not really as a method of payment [3].

Going from this blockchain might really be the gate to a future where transactions are totally trusted and errors re significantly reduced as well as manual labor, in a nutshell it might end certain functions and create new ones and researchers whether computer scientists or business ones are trying to predict the long effects of a new and fast growing technology on the not so distant future [20].

1.4 Blockchain Consensus Protocols

1.4.1 Blockchain Networks Consensus

Maintaining the state of the blockchain canonically across a peer to peer network can be described as a problem of replication of a non fault tolerant state machine, meaning each node needs to have a replicate of the blockchain for the minimum of viewing purposes, for that an agreement needs to happen between all the nodes on how the blockchain should be viewed and built. Failures in consensus leads to malicious attacks such as double spending attacks, coming from faulty nodes showing arbitrary behaviour, or even nodes making mistakes due to forks or problems in connection [7].

Blocks gathered in a sequence is basically the state of the blockchain, and this state goes through a transition once a transaction is confirmed, and a protocol to update the blockchain while maintaining agreement between nodes is very well needed [22].

1.4.2 Blockchain Consensus protocol definition

A consensus protocol or algorithm is a mechanism that enables a group of entities to reach an agreament on the same data value across a distributed network, it is designed to solve a sort of problem to achieve reliability and agreement between all the nodes, this consensus problem can be of multiple dimensions either by solving a puzzle or putting stakes meaning coins on the line or storage space, the important thing is that at the end all the nodes reach an agreement over the true state of the blockchain, thus which transactions are valid [37].

Consensus protocols differs from one another depending on the blockchain network it is used on. Seeing as permissioned blockchain networks practice control among nodes and their synchronization on a more tightly approach, they adopt protocols that provide consensus properties more to their requirements, meaning non fault tolerant protocols.

Contrary to permissionless blockchain networks they don't require identity verification or use schemes of synchronization explicitly meaning there is a tolerance for faulty nodes, meaning the protocols used practice a tolerance to pseudo identities and poorly synchronized nodes [23].

1.4.3 Properties of consensus protocols

1. Termination: the process of achieving mutual consensus on a given data value must come to an end; meaning eventually, every correct node must decide some value.

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

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

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

5. Egalitarian: a system that is trying to achieve consensus must be as egalitarian as possible; meaning the weight of every vote should be equal. In simple words, one vote cannot be less important than another.

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

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

8. Integrity: if x is the value that is decided by the majority of correct processes, then that same value (x) must be decided by any correct process [7].

1.4.4 Domain of application

Consensus protocols are widely applicable in various fields like state machine replication, state estimation, load balancing, control of UAVs (and multiple robots/agents in general), smart power grids, clock synchronization, opinion formation, and others where they have been

modified according to their use. But the most important application of the consensus protocols can be seen in the cryptocurrencies and blockchain based systems [7].

Seeing as the consensus algorithms maintain the security and integrity of a distributed system, so they are an essential part of the blockchain network. Blockchain, an open decentralized system, where each node acts both as a host and as a server and they must share information among all the nodes of the system to reach a consensus to be able to do transactions. some examples of the platforms that utilises consensus protocols are : Bitcoin, Ethereum, Litecoin, Peercoin, BitShaares, Steem and Steemit, EOSIO, ripple, Stellar, ardor and the list goes on. Some more platforms are in Figure 1.11.

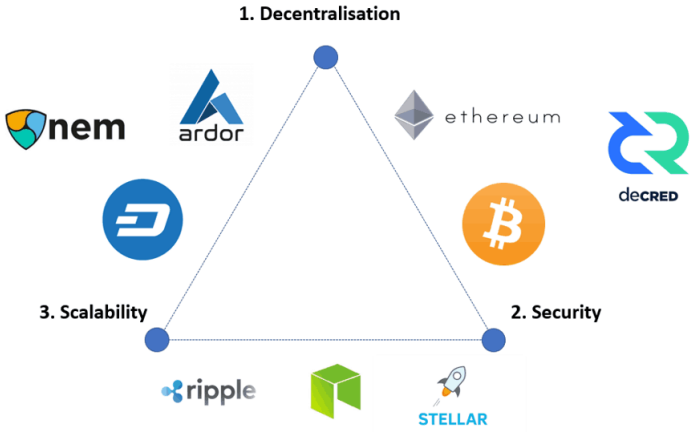

Figure 1.11: Platforms using consensus protocols

1.5 Examples of consensus protocols

The blockchain consensus protocol aims to reaching some specific objectives such as coming to an agreement, collaboration, co-operation, equal rights to every node, and the mandatory participation of every node in the consensus process, thus, a consensus algorithm aims at finding a common agreement that is a win for the entire network [25]. We will start off by discussing briefly some consensus algorithms such as The Proof of Stake (POS), The Proof of Burn(POB), The Proof of Capacity, Proof of Elapsed Time, and then focus our main attention on a specific one which is The Proof of Work consensus algorithm.

1.5.1 Proof of Stake (POS)

In this consensus algorithm we have validators instead of miners who lock up some of their coins as stake as part of an investment in the coins of the system, the way it works is the validators place a bet on blocks in order to validate them if they discover a block which they think can be addeed to the chain, and based on the actual blocks added in the blockchain, the validators get a reward proportionate to their bets and their stake increase accordingly [37]. In the end a vaidator is chosen to generate a new block based on their economic stake in the network. Thus to reach an agreement validatores are encouraged by POS through an encentive mechanism which is the stake [34][23]. Figure 1.12 shows the flow of proof of stake.

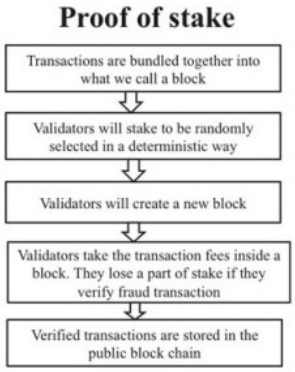

Figure 1.12: Flow of PoS

Advantages:
• Efficiency in energy
The PoS algorithms are efficient in energy comared to other consensus algorithms. Finding a block process in PoS makes it a better suited option in cutting out the energy consumption intensively.
• Security
For security in PoS, if an attack is to happen then the attackers should put their stakes on the line and the assets in order to attempt a 51% attack.
Unfortunately, that also makes them more vulnerable to attacks, and because benefits flow increasingly to the largest coin holders, in a POS system, the richer you are, the richer you get.

1.5.2 Proof of Burn (POB)

Instead of doing an investment into some expensive hardware equipments, with POB validators send coins to an adress from which they are irretrievable, or the correct term is burn coins, and by committing the coins to an unreachable adress, validators earn the privilege of mining the system with a random selection process, So Burning coins permits the validators to have a long term commitment in exchange for their short term loss.

Depending on how POB is implemented validators may burn the native currency of the application that is in the blockchain or the currency of an alternative chain such as Bitcoin. to summarize the more coins the validators burn the more they have a chance for being selected to mine the next block, however the power of burnt coins “decays” or reduces partially each time a new block is mined. This promotes regular activity by the miners, instead of a one-time early investment [23], as shown in Figure 1.13.

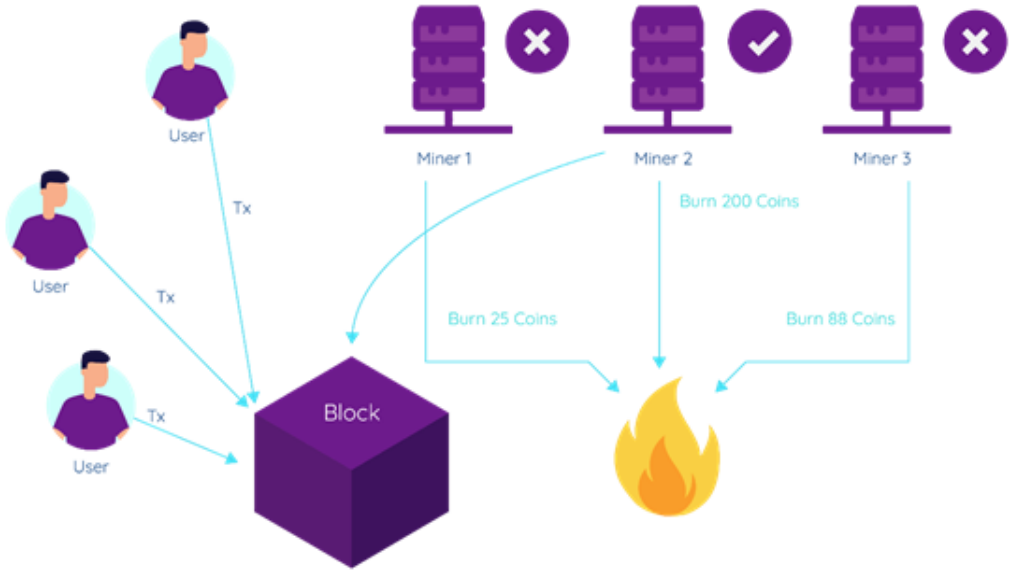

Figure 1.13: Flow of PoB

While POB has an interesting approach to mining, it has its advantages such as the needless waste of resources and it is at best questionable seeing as the mining power goes to those who are willing to burn more coins or money.

1.5.3 Proof of Capacity (POC)

It emerged as one of the alternative solutions for high energy consumption In the Proof of Capacity consensus, the validators instead of investing equipment or burning coins they they invest invest the space in their hard drives, and just like in the Proof of Stake where the more coins you burn the better chance you have to being selected for the mining of the next block, same goes for Proof of Capacity, the more hard drive space validators have, the better are their chances of getting selected and earning a block reward.

Proof of capacity allows the mining devices, on the blockchain network to use empty space on their hard drive to mine the available cryptocurrencies. Instead of repeatedly altering the numbers in the block header and repeated hashing for the solution value, POC stores a list of possible solutions on the mining device's hard drive even before the mining activity begins. In other terms, the larger the hard drive, the more possible solution values one can store, which increase the chance to match the required hash value from his list, resulting in more chances to win the mining reward [23]. Figure 1.14 illustrates the workings of this protocol.

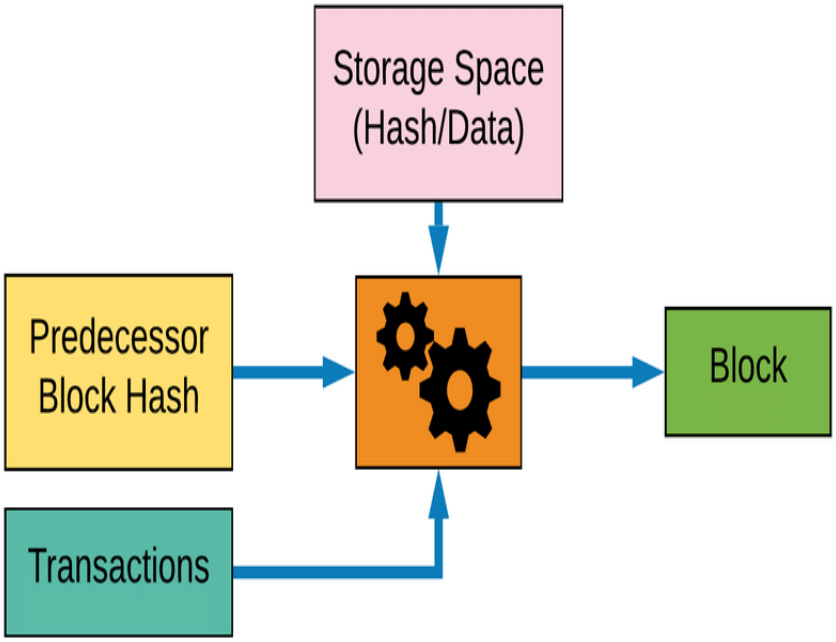

Figure 1.14: Flow of PoC

1.5.4 Proof of Elapsed Time (POET)

The PoET network consensus mechanism needs to ensure two important factors. First, it ensures that the participating nodes genuinely select a random time emphasis on random and not a shorter duration chosen purposely by the participants to ensure the win.

Second, the genuine completion of waiting time. POET was based of the principle of a fair lottery system where all nodes have an equal chance to be chosen, so in a nutshell the PoET mechanism spreads the chances of winning across the largest possible number of network participants. Under PoET, each participating node in the network must wait for a randomly chosen period and the first one to complete the pre-established waiting time wins the new block. Each node in the blockchain network generates a random wait time and sleeps for that specified duration.

The one with the shortest waiting time is the one that wakes up first up and adds a new block to the blockchain, and broadcasts the necessary information to the whole network. And the process gets repeated for the mining of the next block and so and so forth [23][37], as shown in Figure 1.15.

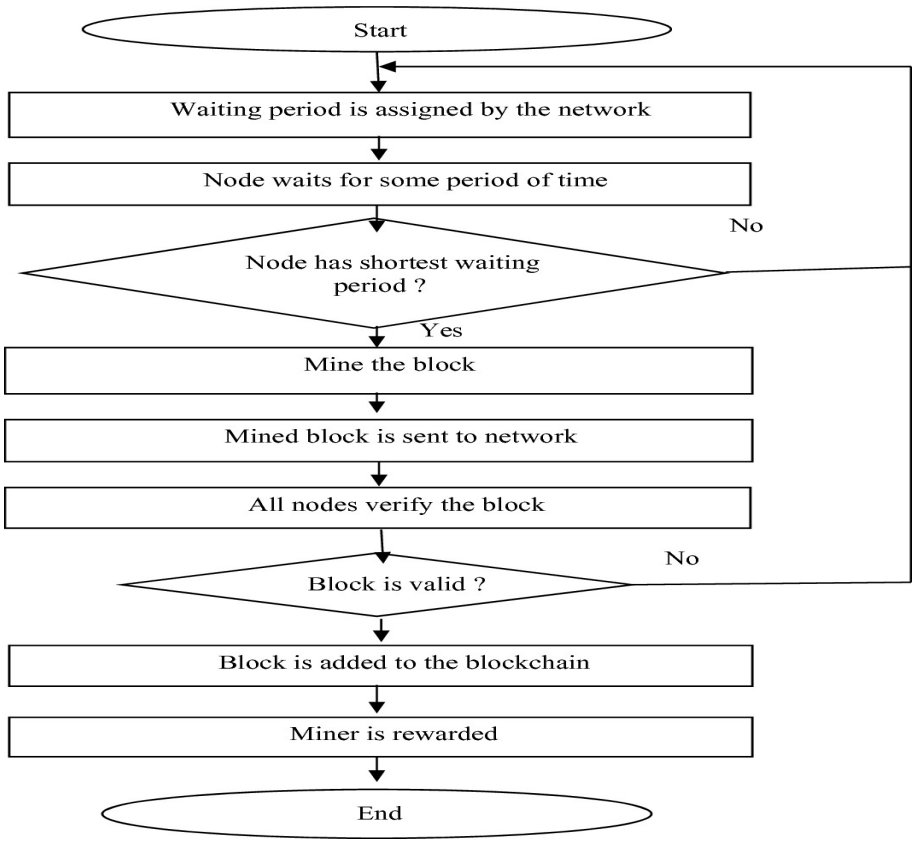

Figure 1.15: Flow of PoET

1.5.5 Proof of Authority (AOA)

Proof of Authority (PoA) is a reputation-based consensus algorithm that introduces a practical and efficient solution for blockchain networks. The PoA consensus algorithm puts on leverage the value of the validators identities,meaning they are not piting on staking coins but their own reputation instead. Therefore, only trustworthy entities are selected as validating nodes to secure PoA based blockchains .relying on a limited number of block validators and this is what makes it a highly scalable system as shown in Figure 1.16. Also blocks and transactions are verified by moderators of the system who are pre-approved participants.

PoA consensus algorithm may be applied in a variety of domains and is considered a high-value option for logistical applications such as supply chains. The Poa algorithm allows companies to keep their privacy while still benefiting the from the blockchain technology, for example Microsoft Azure implements the PoA algorithm [23].

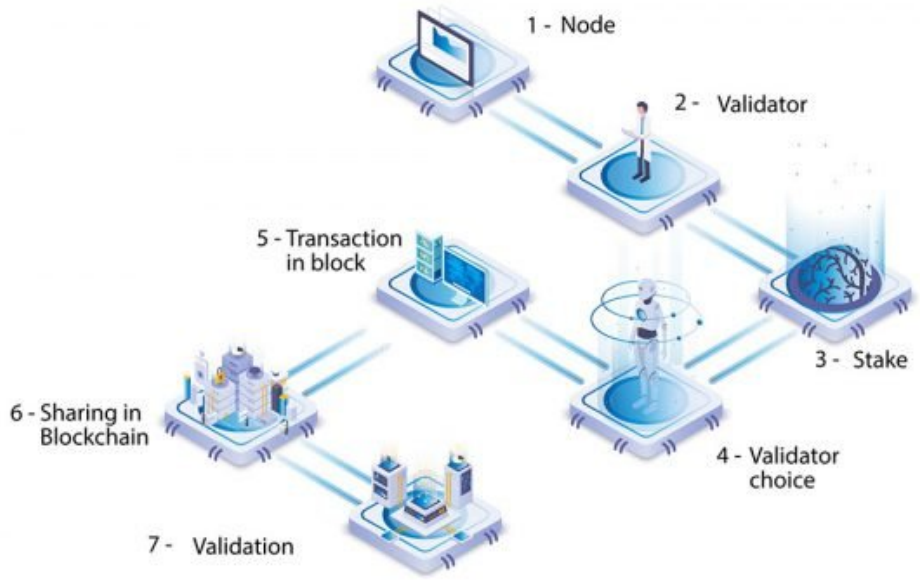

Figure 1.16: Proof of authority

1.5.6 POW Proof of Work protocol

Firstly used by bitcoin, it is the first mechanism for crypto-currency consensus, proposed by Satoshi Nakamoto as a means to organise the mining process, it takes its name the proof of work from the very large processing power [27] needed to satisfy the protocol's condition, which is mainly solving a math or the crypto puzzle and claiming the reward from the network with a pre-established crypto-currency amount [26][23].

1.6 Conclusion

Blockchain is a revolutionary new technology that has taken the world by surprise due to its various uses in different fields such as cryptocurrency, health, finance, business management, and secure data storing an many more other applications. Due to the fast growing interest in it, consensus protocols have been set in order to secure and manage blockchains.

We have gone through the different concepts of blockchain and consensus protocols, the next chapter is describing the Proof of Work consensus protocol where we will dissect it more thoroughly.

Chapter 2

In depth proof of work protocol analysis

2.1 Introduction

A consensus protocol is considered to be highly effective, if it maintains the integrity of the blockchain network that's running it, and what better way to show this effectiveness than to not rely on a third party but to make the members of this blockchain network themselves keeping the security of the system by having them fulfill a certain criteria, to eliminate the trust issue between untrusted parties. In this chapter we will be going through the proof of work protocol, stating its operating principle, the incentive behind it, and entities involved in the functioning of this protocol, its problems, and everything that revolves around this protocol, then we are going to talk about some related work to our project.

2.2 Proof of work protocol

2.2.1 Definition

The proof of work consensus protocol was applied for the first time by Satoshi Nakamoto based on his famous white paper [2] in 2008 that outlines the technology of blockchain and revolutionized the decentralization as a term, which led to him applying his paper to create the Bitcoin cryptocurrency.

The aim of this consensus mechanism is to reach a coordinated agreement or a trust between all the nodes in the blockchain network, which is by default an environment with multiple entities, where no entity trusts another, on the validity of the blocks and the validity of transactions between bitcoin users to put inside a block [28].

2.2.2 Operating principle of PoW

The Proof of Work consensus protocol revolves around a process of solving a puzzle that is challenging both mathematically and computationally, and verifying transactions made by entities called Clients, which would lead in the creation of new blocks and adding them to the blockchain of Bitcoin [17]. This process is called mining, and the nodes of the blockchain network that are responsible for finding new blocks and adding them to the chain are called miners [1].

The miners engage in the process of competing to find the next block, in the hopes that they claim the reward of finding said block, and they do that by mining transactions that is made between two clients, by verifying the validity of these transactions that are to be added

to the block, and then organizing them in an order, and announcing that a new mined block has been found to the entire blockchain network. Verifying transactions isn't necessarily the hard task for miners since it doesn't consume much time and energy, the hard part is solving the mathematical puzzle, after that it becomes easy to link the found block to the previous one in the blockchain as shown in Figure 2.1. Once a miner comes to a right solution of the mathematical problem or puzzle, this node broadcast the block to the entire network, simultaneously, and claiming the cryptocurrency reward [32], keeping in mind that the prize of finding a new block decreases by approximately half every four years in the bitcoin network [8].

The process of mining has been attracting newer and more miners, with this increase in competition the time to find a new block is bound to get shorter every time, and in order to keep a consistency in the approximately needed time to find a new block, and the difficulty level of the bitcoin blockchain network keeps changing the difficulty of the computational problem to find a new block accordingly, until a maximum number of blocks of 21 million is reached which is expected by bitcoin researchers to be around the year 2140.

The principle behind the Proof of work mechanism is finding a solution that that is easy to verify but difficult to find [6].

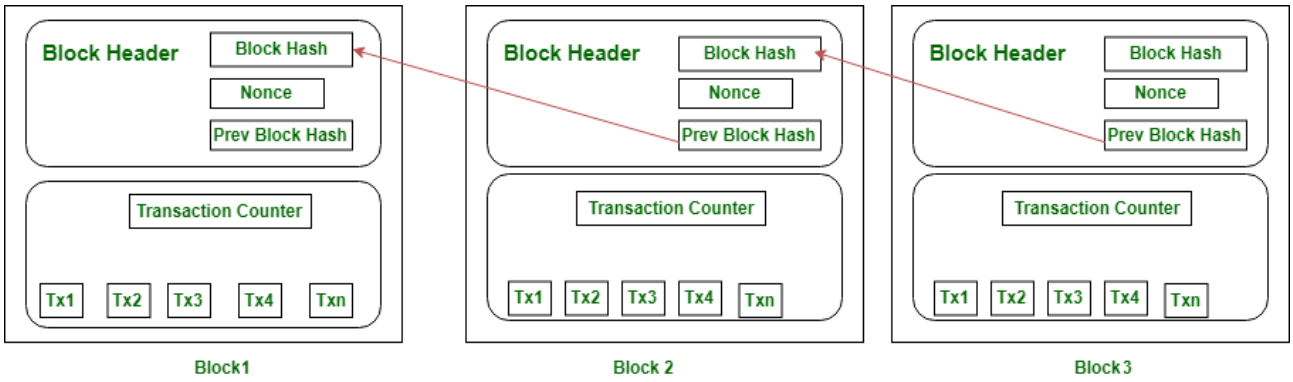

Figure 2.1: Block linking

As seen in the figure above each block is connected to the previous block through a particular number called the hash number, this number is very important to ensure the security and organization of the blockchain, in a way that if the hash number of Block1 is different from the hash number of block1 in block2 no link between Block1 and Block2 would be established, and Block2 will not be verified and considered to be a valid block, so any miner that finds a new block has to put in the information of the mined block the hash number of the last one in the chain.

The first ever block that was found is called the Genesis block, it was established by the creator of bitcoin and the implementer of Proof of Work on it, Satoshi Nakamoto, this Genesis block has no previous block number or previous hash value. To protect the safety and security and to avoid tampering with the chain of blocks, in order to change a block, a miner has to create a block that has the same predecessor as the one that he wants to change, and to do so to the entire successors of this block on the chain, meaning doing the solving of the computational problem the number of times as the blocks that has been changed, which is computationally and mathematically practically impossible. Figure 2.2 shows the flow of the proof of work protocol [8].

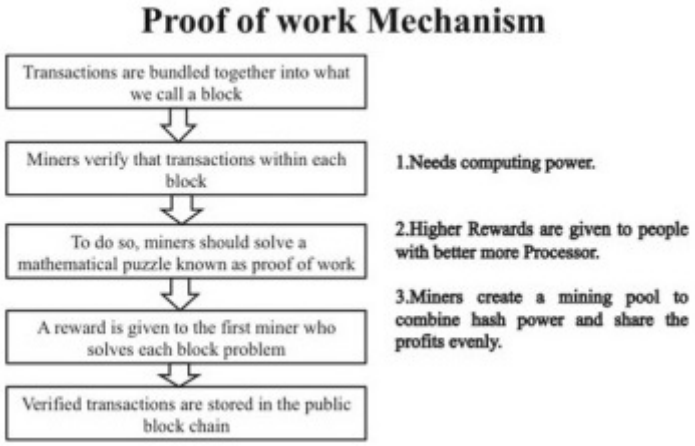

Figure 2.2: Flow of PoW

2.2.3 Incentive

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended. The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free [2].

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth [2].

2.2.4 Entities involved in the Mining Process

1. Miners:

   Mining is the process of creating new blocks, this activity performed by the nodes in the network who are called Miners, who impose great significance on the system's functionality. Miners collect transactions from a transaction ledger that is shared by every miner on the network, verify the validity of these transactions to make sure there is no fraud and that this transaction really exists, by checking its information, like the sender and the receiver, and the amount sent [4].

   after the validation of these transactions, miners put them into a block to start the next and most important step. All the miners keep an identical ledger and work on the same transactions.

   Right after the verification step is complete they start doing the computational puzzle and they attempt to brute force as many possible values of results for the mathematical problem as possible, which is known as the nonce, until such a solution is fetched that

makes the hash of block n+1 below a specific threshold and once that is done, the mining process is considered to be complete, a new block is found and generated and broadcasted by the winning miner to the rest of the nodes in the network [4].

The winning miners claims the reward for finding the n+1 block, which is the main incentive why miners waste such computational power and time consumption in the process. The steps are illustrated in Figure 2.3.

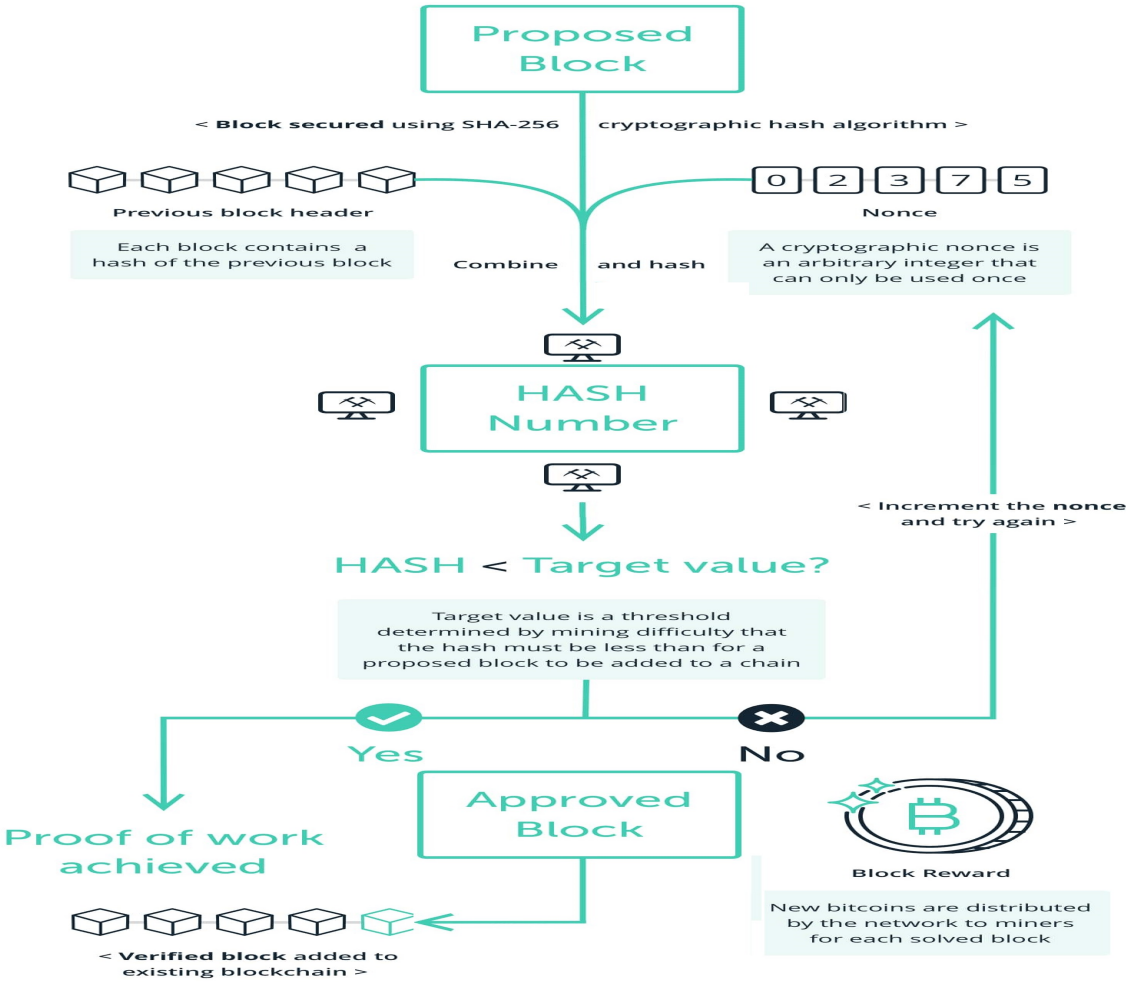

Figure 2.3: Miner function

2. Clients:

The ones responsible for making transactions where they sign off from their own wallets applications, either by sending a certain amount of cryptocurrency or in some cases tokens to another client, and by which this transaction is stored in a ledger called the transaction pool, a ledger that contains unconfirmed transactions waiting to be validated by miners [38], and put into blocks [22]. These clients making transactions is the basics and the core of mining for a block, since a block's main purpose is to store data of handlings between users, whether it is bitcoin or any other cryptocurrency, so in simple terms they trigger the process of mining for blocks, by handling the give and take transactions [22].

Transactions between clients contain the sending user's address and the receiving user's address and the amount that was transferred between the two. Some clients attempt to practice fraud in their transactions by sending an amount that they have already spent in a different transaction, and this is why any transaction made by clients is considered not verified until its put in a block and the transfer is finalized and legitimate.

Clients also keep a ledger of the current blockchain, to stay tuned to any increase or decrease or modification of the blockchain as shown in Figure 2.4 [35].

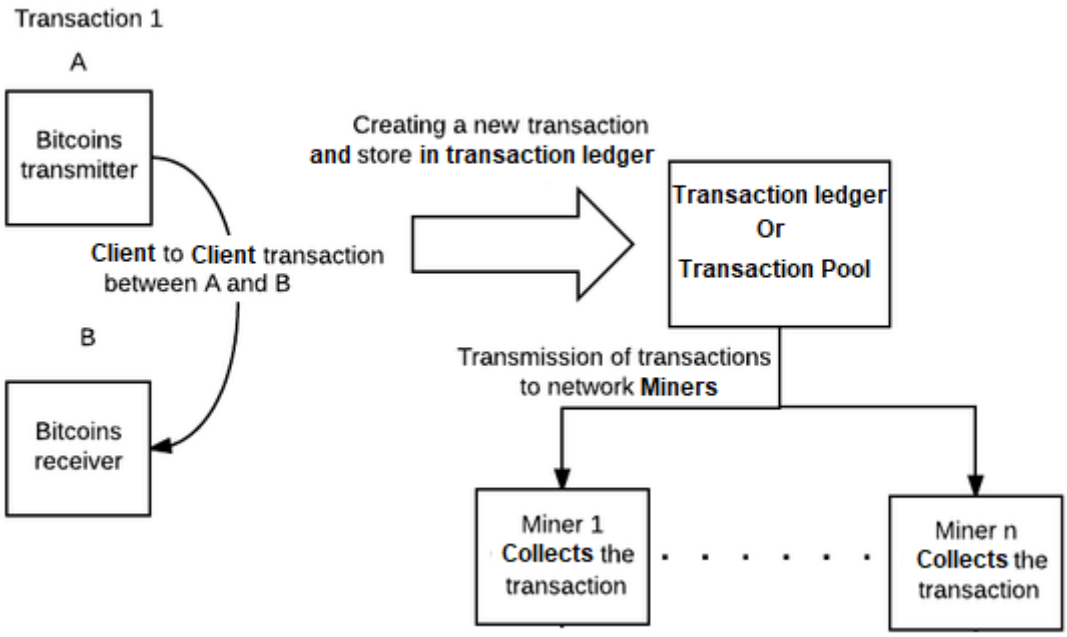

Figure 2.4: Client function

2.2.5 Transactions

A transaction is an electronic coin consisting of a chain of digital clients signatures or addresses, the owner of this coin transfers it digitally to the intended receiver, by signing it with the address of the next owner as shown in Figure 2.5, and the hash of a previous unspent transaction, and linking these information to the coin. The receiver of this coin may track the chain of ownership, but he may never know for sure whether the sender has spent this coin before or not, meaning there may be a double spending. There is a common solution to put an intermediate authority devoted only to check transactions for double spending, and only coins that are validated and verified to be unspent may be included in transactions [2].

There is a catch with this solution, which is whether this intermediate authority can be trusted, and if the fate of an entire system of coins can depend on the organization that is running this double spending check, after all they will end up doing the same function of a bank, where every transaction must go through them. The client that receives the coin must then check all earliest transactions containing the address or signature of the sending client [30], meaning he has to be aware of all earliest transactions, and for that they need to be publicly announced, and an order of transactions must be set, for if a transaction has been spent at a particular time has a unique number, and trying to spending it again would cause a contradiction, and the double spending attempt would be revealed [29].

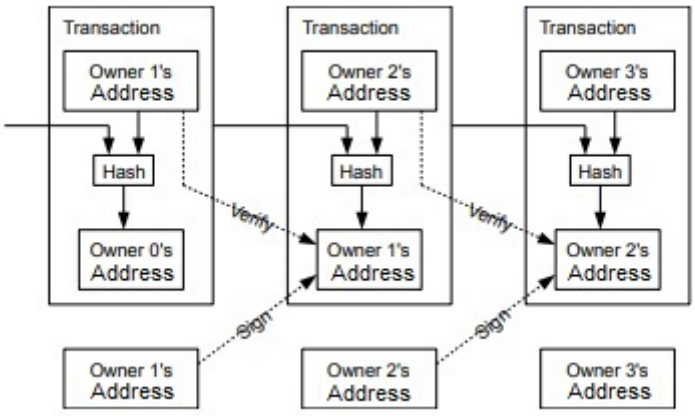

Figure 2.5: Transaction linking

2.2.6 Network running Proof of work

A blockchain network running a proof of work protocol would need to follow these steps:

- Every new transaction made by clients must be shared by all nodes at all times, in a shared access ledger.
- A block is filled with transactions collected from this shared ledger, which is called a transaction pool.
- Nodes attempt in a competitive way to find the difficult solution to a mathematical puzzle for the current block.
- Every block found by a particular miner, is broadcasted to all nodes across the network.
- The block is accepted by the rest of the nodes, after making sure all the transactions contained in it, are valid and spent for the first time.
- A sign that the block has been accepted by nodes on the network, is when they start working on the next one in the chain, basing the accepted block's hash as the previous hash in the new one.
- The longest chain is considered to the nodes as the correct one, and will be built upon, and extended.
- When two miners find the next block at the same time, and broadcast it simultaneously to the rest of the network, a group of nodes may receive one version, while another group may receive another version, the one that is considered and worked on would be the first one that is received, but the other branch is saved in case it becomes the longest chain.
- The tie breaker would be after a condition is met, whether it be a time period after that the longest chain becomes the definite chain, or after a certain amount of blocks is reached, and the other branches that certain nodes were working on will be abandoned and orphaned and they will switch to the definite chain.
- New transactions won't necessarily reach the entirety of the network, because of the massive amount of transactions coming to play at every moment, they only need to reach many nodes to make sure that they get included in a block [2].

2.2.7 Solving the Crypto Puzzle

The basics of this puzzle is guessing at random a hash value.Predicting the output is made impossible by using the hash function. considering this difficult task, the minors predict a random value and apply it to the hash function and the data stored in a particular block. The number of zeros that are pre-established will be the start for the resulting hash.If a particular miner, in the peer to peer network solves the crypto puzzle or finds the hash within the time period of finding a new block, the announcement takes place to all the nodes connected to the network to stop mining, for the n+1 block has already been found by a certain miner.

Then the other miners will stop their work, add the new found block, and proceed mining for the next one and solving their crypto puzzle. The Crypto puzzle is the main handicap standing between miners and getting claiming the reward . The consistency of the mathmatical problem aka the puzzle is maintained in a decentralized manner by the peer to peer network.

2.3 Cryptography and Block Building

2.3.1 Hash function SHA-256

The proof of work consensus protocol used in bitcoin and many other cryptocurrencies uses the hash function SHA-256 from the well known family for hashing security SHA-2, created by the United States National Security Agency in 2001, it is the standard security hashing for Federal Information, and for the institution of Technology and Standards.

The hash function SHA-256 is widely used at the moment in a lot of applications of encryption and system security due to its high effectiveness. The way it works is that given as an input any size of any number or a string, manipulates the given data and outputs a sequence of a fixed size, it has a binary value of 256 bits.

Like any other crypto hashing function the SHA-256 has all the basic properties, calculating at a fast rate when it comes to complexity of computation, and when given an output of a 512 bits, it is impossible to calculate the input, meaning it is a one way function, and also when given two different inputs it almost infeasible to find an output generated by the two that are the same.

To sum up the SHA-256 properties in a list:

- SHA-256 is a one way function.
- Fast computation of an output y, when given an output x, with the hash function as H as the following equation y = H(x).
- Resistance to image (input) detection, for if y = H(x), x meaning the image is uncomputanable.
- while x and x' are two images, x' is uncomputanable from x with H(x) = H(x').
- No two images x and x' produce the same output meaning there is no H(x)=H(x').

With the help of the hash function, the miners try to find the random nonce (small size random data) and find the block that holds the hash (binary values), with particular 0's.

A hash with the needed number of zeros is a rare hash to find, but in all cases miners have to try many times to find a perfect nonce. The number of zeros were based on the difficulty faced by the miners to search for a perfect block [36]. The basics of SHA-256 is illustrated in Figure 2.6.

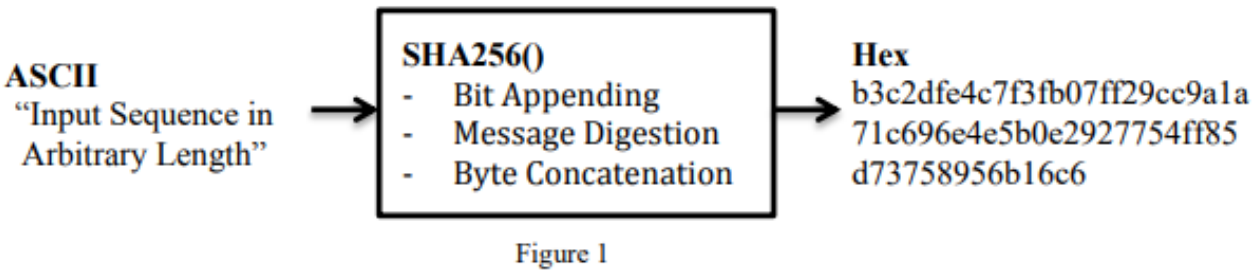

Figure 2.6: Logic behind SHA256

+

2.3.2 Nonce

Cryptocurrencies and blockchain adopting the proof of work have a central part which is the concept of the nonce.
A nonce, abbreviation for number only used once, which produces a hash that is lower than or equal to the hash value where its difficulty is fixed by the network called the difficulty target. To find these values, miners do the mining and compete against each other. In that network, the miner who finds a nonce is named as the golden nonce, and he gets to claim the reward for finding a new block and adds it to the blockchain.

Now the next part shows us how the process of finding a nonce is done in mining. In Bitcoin, PoW makes sure that there is no majority dominion of the blockchain network's mining, if a nonce is found and is yet the block is not validated in a time period defined by the network, and also if the nonce is found in a shorter amount of time that it is noticeable, the hardness of the puzzle is adjusted accordingly, works also if a nonce is not found within the network's set period the hardness of the puzzle is diminished, meaning depending on the fast or slow obtainment of the nonce, a default rate is established for the system, and is adjusted routinely accordingly.

This method to solve the puzzle prevents malicious behavious like cheating at the mining process from multiple aspects, where there is no leveraging machines with highly powerful computational powers, forming colluding partners, or faking the proof of correctness. Easy verification of the correctness of the solution is also an innovative feature, and here in Figure 2.7 is an example of the nonce of the genesis block.

Genesis Block	
Previous Hash	0
Data	Welcome to Blockchain
Hash	000018035a828da0…
Nonce	56551

Figure 2.7: Nonce example in the genesis block

2.3.3 Build blocks using nonce

Proof of work in blockchain after setting up the the difficulty target, as explained in the previous paragraph, the miners choose a random nonce and add it to the block header in the attempt to build a new block.
If the number of zeroes needed to be lower than the difficulty target is not reached, then keeping the header and looking for a new nonce is what the miners will keep doing until reaching a valid nonce that produces the right combination between the header and the nonce. The hashcode (i.e, the target block header bh) of the concatenation of x and the nonce is smaller than a target value D(h): bh = H(x+nonce) <= D(h).

2.3.4 Block Reward

When miners in the network mine successfully a new block they get rewarded with bitcoin.

- The number of bitcoins a miner gets, depends on the number of rewards from mined blocks.
- The reward becomes half for every four years or every 210,000 blocks.
- each transaction put in a verified block, augments the reward of mining this block by a certain percentage, while some transactions add no increase to the reward.

2.3.5 Flowchart of Proof of Work

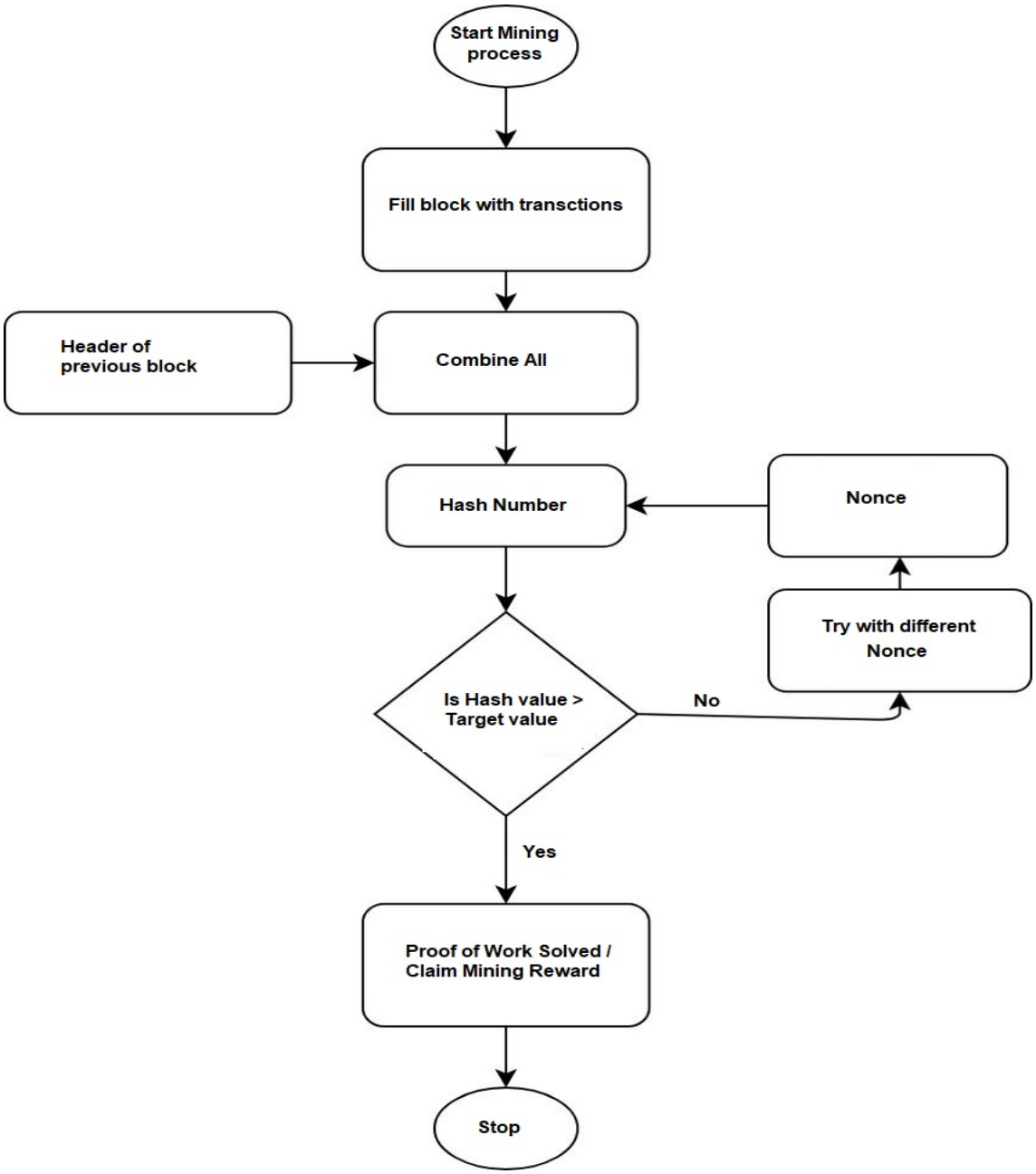

Figure 2.8: Flow chart of proof of work

2.4 Proof of Work Main Problems

The following are some of the issues for the Proof of Work consensus mechanism.

2.4.1 Computation Power

1. Time consuming: for solving the crypto puzzle, the miners have to check for the nonce which must be solved to mine the block. This helps in time [33].

2. Resource consumption: in order to solve the mathematical crypto puzzle, the miners consume high computational power. This results in wastage of resources like hardware,

space, money, and energy. It is estimated that the world's electricity spent for verifying transactions in 2018 was 0.3% [33].

3. Carbon footprint: 34.73 Mt CO2 In comparison with the carbon footprint of Denmark over 723,140 VISA transactions with 48,872 h of time watching YouTube [33].

4. Electrical Energy: 73.12 TWh In comparison with the power consumption of Austria, the equivalent power consumption has an average of over 20.61 days in US.

5. Electronic Waste: 11.49 kt In comparison with the e-waste generation of Luxembourg, as shown in Fig. 4, the equivalent weight of 1.48 "C"-size batteries or golf balls around 2.09 [33].

System function	Proof of work [POW]
Mining power	The work (solving the crypto puzzle) done by the miner
51% attack	Level incentive to avoid 51% attack
Energy consumption	Higher
Decentralized versus centralized	Becomes powerful when nodes which tend to be centralized over a period of time
Target time for a block	For every 10 min

Table 2.1: PoW analysis

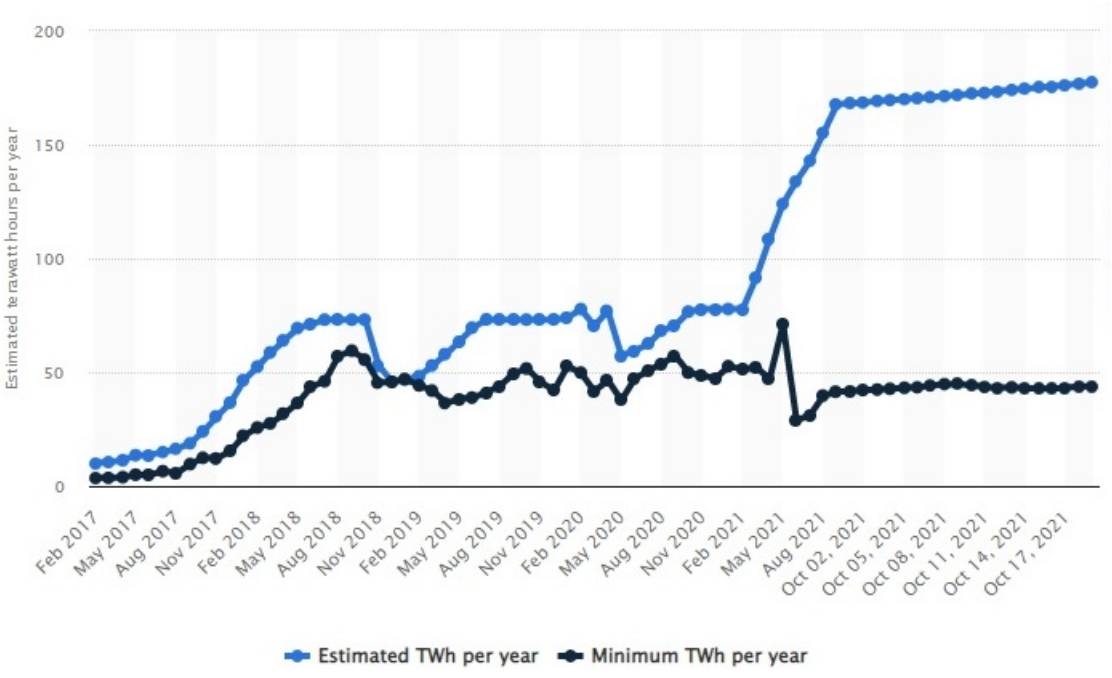

Figure 2.9: Energy consumption chart

The energy consumption chart shown in Figure 2.9 is taken from statista.com website it shows the Bitcoin energy consumption worldwide from February 2017 to October 19, 2021 in terawatt hours. It clearly manifest the increase in energy consumption with each year passed which is one of the biggest issues of the Proof of Work protocol as we have seen in the previous paragraphs not only high resource consumption but electrical energy highly consumed in order to keep up the mining process [16].

2.4.2 The 51% Attack

In the blockchain network, if any node gains 51% or more than 51%, the nodes could influence the blockchain by gathering most of the network. Usually the attacks are mainly by a group of miners who control more than 50% of the mining hash in the network or by the computational power. Meaning the attackers create their own private branch of the blockchain by combining their computational resources, and once this fraudulent chain is the longest, they broadcasts it to the entire network which leads the other nodes to abandon the actual chain because its blocks had lesser blocks [7]. The visual working of this attack is depicted in Figures 2.10, 2.11 and 2.12.

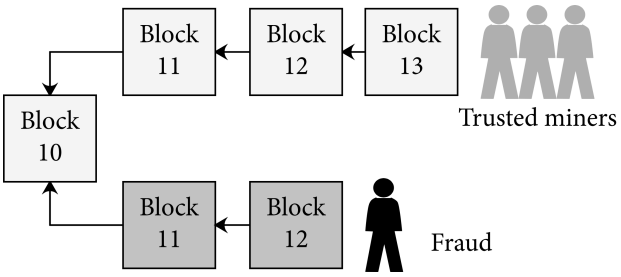

Figure 2.10: 51% attack: fraudulent chain creation

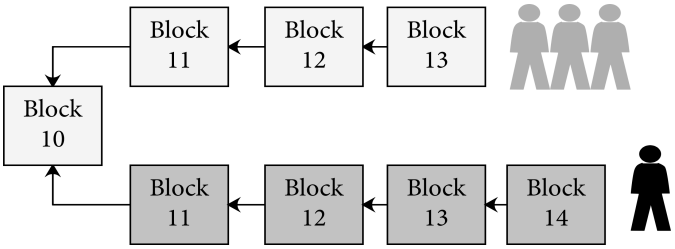

Figure 2.11: 51% attack: fraudulant chain overgrow main chain

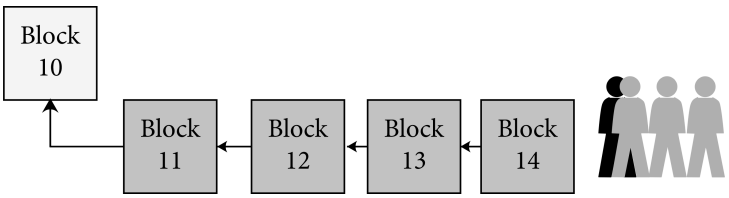

Figure 2.12: 51% attack: main chain abandoned

The ability of the attackers is to prevent the transactions that gain confirmations and can halt the payments between the users. They can also reverse the completed transactions of any

user in the network and this leads to Double Spend coins. The attackers could not alter old blocks or create new coins in the blockchain-based cryptocurrency [33].

2.4.3 Forks

Permissionless blockchain networks, consists of some heavily competitive atmosphere, due to the increasing growth of participating nodes in the mining process, in theory thousands of nodes or miners could be simultaneously looking for the next block in the chain, and trying to solve the proof of work puzzle, and the reason for making the puzzle so difficult is to avoid the chance where two nodes solve it at the same time, which would cause a change to the blockchain network called a fork.

A fork is when two miners find the proof work solution at the same time as presented in Figure 2.13, and broadcast their found block to the rest of the network, some may adopt the first block because they received it first while others adopt the second block, causing a branching in the chain [33].

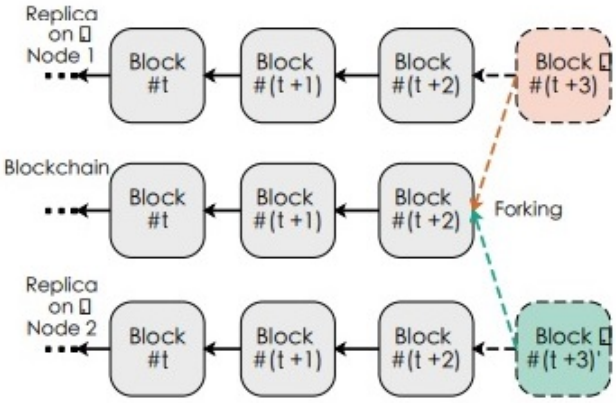

Figure 2.13: Fork between nodes 1 and 2

Every branch will be build upon by a group of miners , which in their mind are working on the definitive chain, as do the other group of miners, and these two branches will keep expanding and the change in blockchain becomes more and more apparent.

The blockchain's reliability is affected by forking, this is why in proof of work, to counter this change a mechanism must be set, either by setting a time interval, so in case there is indeed a fork, it would be resolved by that interval's ending, adopting the longest chain as the definitive chain, or by setting up a number of blocks after which the first chain that reaches that number would be considered the longest and there for the definitive chain. In the Table 2.2 below we show how complexity of the problem has a huge impact on forks manifestation:

Problem Complexity [1]	Average time to mine a block(in seconds)	Average fork length
2.10^{-4}	0.03	21.47
2.10^{-5}	0.32	8.27
10^{-6}	2.62	2.61
2.10^{-7}	8.6	1

Table 2.2: Analysis of different levels of problem complexity

We observe from this table that with the increase in complexity of the problem the time to find a new block is augmented, an expected and logical result of course, and the second observation is when complexity is increased less and less forks occurs, which is a result statistically logical, the more the problem to solve is difficult the less nodes will be able to solve it at the same exact time.

2.4.4 Double Spending

The risk of spending a digital currency twice is called Double spending. The savvy individuals, who understand about the blockchain network and the computational power needed were reproduced to obtain the digital currencies that are unique potential problem. This type of issue will not happen during the cash transactions, the parties involving in transactions can easily verify the authenticity and the ownership of the physical currency. But in digital currency, the digital token can be copied by the holder and it might be sent to any of the parties or the merchants with the original one [9].

2.5 Related Work

Related research and similar ideas to our project is presented in this section. Modeling proof of stake protocol using Uppaal, by the department of computer science in the Rutgers university, New Brunswick, by Dholakia Niraj.

The tool used in this modelization is the same tool that we will be using to model the proof of work protocol [31].

The aim of their project is to simmulate the proof of stake mining, observe its affects on the blockchain's growth and change.
The basics on which this work relies on a simple rule, the more you stake, the more you chance you have of being selected to mine the next block in the chain [31].

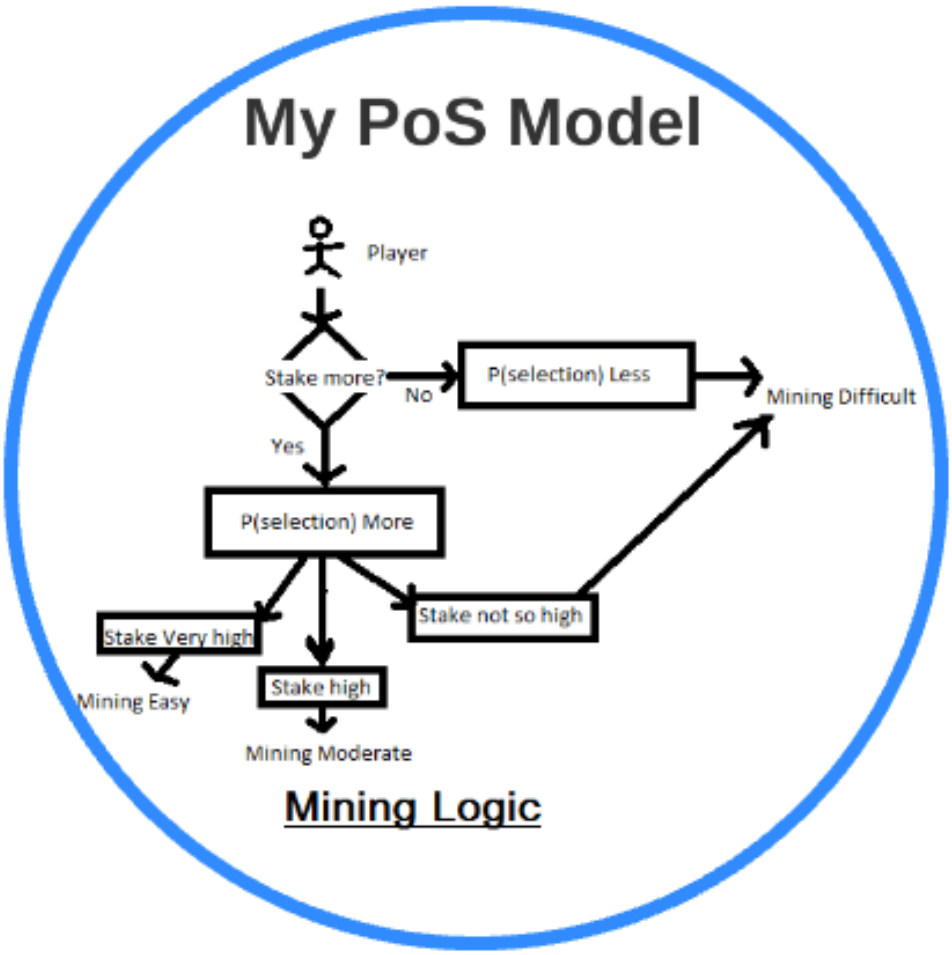

Figure 2.14: PoS mining logic

As shown in the Figure 2.14 above, the pos model is very simple, and easy to comprehend.

A random number between 0 and 3 has been set to rotate between all participants in the proof of stake mining, this number represents how high the stake of this node is, where 3 being the highest and 0 being the lowest stake. The node with highest stake is considered to be the winner and is selected to mine the next block.

As we can see no high performing machines or computational ressources is needed for proof of stake, but as we can see from the name of the protocol the stake is cryptocurrency meaning the stake is literally money.

2.6 Conclusion

Satoshi Nakamoto's Proof of Work protocol has been widely used in many cryptocurrencies, the first one being Bitcoin where he applied his white paper.

Proof of work has many ups and downs, the most noticeable ones are the massive computational power it needs and consumes, in order to mine a single block, but from the bright side this also provide more stability to the blockchain system, where it controls the quantity of mined blocks, also it provides a secure and validated block mining.

In this chapter we have discected the proof of work protocol, and tried to cover all its aspects, in the next chapter we will be modeling and verifying the PoW protocol using the UPPAAL tool.

Chapter 3

Modeling and verification of PoW protocol in Uppaal

3.1 Introduction

After we have thoroughly gone through the proof of work protocol for blockchain, and covered all its aspects and entities and its functionning, now we will be modeling this protocol using a modeling and verification tool, which would alow us to better understand it, and would permit us to test some properties within the protocol.

For this modeling and verification task we will be using a tool called Uppaal, and in this chapter we will be modeling the two entities involved in this protocol, and verifying some properties satisfaction and doing some simulation and probabilistic calculations.

3.2 Modeling PoW protocol using Uppaal

3.2.1 The Uppaal tool

Uppaal is an integrated tool environment for modeling, simulation and verification of real-time systems, developed jointly by Basic Research in Computer Science at Aalborg University in Denmark and the Department of Information Technology at Uppsala University in Sweden. It is appropriate for systems that can be modeled as a collection of non-deterministic processes with finite control structure and real-valued clocks, communicating through channels or shared variables. Typical application areas include real-time controllers and communication protocols in particular, those where timing aspects are critical. Uppaal consists of three main parts: a description language, a simulator and a model-checker. The two main design criteria for Uppaal have been efficiency and ease of usage, to facilitate modeling and debugging [5].

3.2.2 Description of PoW protocol in Uppaal

In our description of the PoW protocol we will be modeling the miners and the clients, the two main entities involved in the Proof of Work protocol functioning, and the other different components of the PoW protocol such as blocks and transactions and their description. A transaction is only considered confirmed if a block containing this particular transaction is included in the chain.

The two parties that are needed in order for the mining process to begin are Miners and Clients. A Miner is responsible for solving the proof of work puzzle and there for creating new blocks, and each Miner maintains its own chain. A Client can be either sending Bitcoins or receiving it, there for responsible for transferring cryptocurrency from one Client to another

and creating a transaction in the process. Clients do not participate in the mining process, But they do keep a copy of the blockchain. In the next subsection we will discuss the data structures that is used to store necessary information such as transactions and blocks.

3.2.3 Proof of work algorithm

The algorithm that summs up the procedure of the Satoshi Nakamoto's proof of Work is as follows [1] :

```
/* Joining network */
1 Node join the blockchain network;
2 Start Mining;
/* Main loop */
3 while running do
4 — if Mining() returns block then
5 — — Write block into blockchain updateChain();
6 — — — /* broadcasting rule */
7 — — Broadcast block to nodes on network foundBlock!;
8 — end
9 — receiving block by nodes foundBlock?;
— /* adding block */
10 — if block received & is valid then
11 — Write block into blockchain addBlock;

12 — end
13 end
/* PoW-based block mining process */
15 Mining functions:
16 — Get transactions from pool getTransaction();
17 — Perform the double spending check doubleSpendCheck()
18 — Put transaction in the block
19 — hash of the last block in the chain, and other needed information for
20 — status of blockchain;
/* mathmatical puzzle for PoW */
21 — Find a solution nonce that satisfies the
22 — following condition:
23 — Hash(H + nonce) < target checkNonceDifficulty()
24 — return new block;
25 end
```

3.2.4 Structures

The transaction's structure designated by Tx, contains a unique identifier which is initialised to 2 since every client will have one transaction to start with, an input and an output transaction, and the status of the transaction whether it is confirmed once it is put in a block, unconfirmed when it's still in the transaction pool and haven't been taken by a miner to include in a block, or invalid if it cannot be put in a block due to it being spent already or its output is negative or equal to zero, it is designated as STATUS.
An input transaction which is designated by TXIN, it contains the number of the previous transaction where there is a positive output(positive amount of Bitcoin cryptocurency) that

this current transaction is planning to use.
An output transaction designated as TXOUT, contains the Address of the client and the Amount sent as value.

Structure of a Transaction

```
const int MAX_TRANSACTION = 100;
const int CLIENTS = 4;
const int UNCONFIRMED = 0;
const int CONFIRMED = 1;
const int INVALID = 2;
typedef int[0,CLIENTS] CLIENT;
typedef int[0,10] AMOUNT;
typedef int[0,MAX_TRANSACTION]TXNUMBER;
typedef int[0,2] STATUS;

typedef struct {
TXNUMBER Id;
TXIN TxInput;
TXOUT TxOutput[2];
STATUS status;
}Tx;

//output transaction
typedef struct{
CLIENT Address;
AMOUNT value;
}TXOUT;

//input transaction
typedef struct{
TXNUMBER id;
}TXIN;
//transaction
```

The block that transactions are placed in, has in its structure a block number, which normally is represented by a block hash, but to simplify things and for the lack of a hashing function within the Uppaaal tool, we represented it by an integer, so it will be represented by a block number and will be initialized as 2 since the genesys block is always the first block, a block also has a previous block numberthat allows the block to be linked to the chain, also a block contains transactions obviously.

Structure of a Block

```
const int HASHSIZE = 100;
const int MAX_BLOCK = 100;
typedef int[0,HASHSIZE] HASH;
HASH Block_num = 2;

typedef struct {
HASH num;
```

HASH prevBlocknum;
Tx tx;
}Block;

For a client we have the wallet structure which contains the transactions and their amount.

structure of a Wallet
typedef struct{
TXNUMBER Id;
AMOUNT value;
}Wallet;

We also defined a channel foundblock to broadcast and receive mined blocks to other miners and clients, it goes in effect when a miner has found a the solution for the proof of work problem, other variables are chain which is used to hold the details of a block meaning its number, the previous block number and the transactions within the block, and the variable Block_num is used to store block numbers and is initialised to 2 when block number 1 is the genesys block, and tempChain is used to store block details when broadcasting a block, and TxPool is the ledger that stores transactions created by clients.
In the next subsections we will discuss the Miners and the Clients models.

3.2.5 Miner

As shown in Figure 3.1 below, the miner automaton has four locations, Initial, Wait, DoubleSpendingCheck, and Mine.The local declarations for the miner automaton includes length of the chain for every miner, the length of the longest chain,and the index of the longest chain, when a block is found we use these variables to generate new blocks. We have represented the blockchain as an array of integers, its two sizes are HASHSIZE and 4, chain[HASHSIZE][4], in chain we will store the number of the block, the number of the previous block which permits the linking, the id of the miner that found a particular block, and the length of the longest chain.

We keep all details of a certain block kept in an array called blockDetails that stores transactions and their details, the current and the previous block number. When grabbing transactions from the transaction pool, they are stored in The variable tempTx, and when doing the double spending check the variable tempTx is used as well for the verification process. In the miner automaton we represent the functionality of a miner, from first to the last procedure.

The first function being taking transaction from the transaction pool verifying that there is no double spending and if the check is positive placing these transactions in a block. The second function is solving the mathematical puzzle, meaning finding the proof of work and finding the suitable nonce, and this task is the hardest one.

The third function is when the mining process is successful, the miner broadcasts the found block to every pther node, meaning miners and clients alike. The fourth function represents when a miner receives a block, stops the mining process, adds that block, and start over on the next block in the chain.

The final function is deciding when to orphan a chain, and which chain is orphaned and which is not. The location wait is the starting point of the automaton, upon transitionning to the location wait the chain is initialized, and the first block is set to 1(genesys block), meaning any miner that finds a block, it will numer it 2, and set the previous block number to 1.

There are 5 edges coming out of location wait, and 4 coming in. First every miner checks if there are any unconfirmed transactions in the transaction pool, by taking the transition with a

guard of UnconfirmedTxPool, and if there are grab them using the function getTransaction() and copy them to the local variable tempTx, and go to location DoubleSpendingCheck, and also to make sure that there is a positive amount being sent, to try to validate these transactions.

If there is no double spending found meaning this transaction hasn't been spent before, which is verified by comparing the input transaction and the amount with other transactions, and the amount is legitimate and positive then this transaction is set to be confirmed and we transition to the Mien location, otherwise if the check is unsuccessful we return back to location wait, set the status of the transaction to Invalid and get other transaction to try and confirm.

In location Mine, we have two edges, either the miner finds a suitable nonce, hence solving the proof of work problem, and broadcasting this block to the rest of the nodes through channel foundblock and updates the chain with function updateChain(), the transaction tempTx is added to blockDetails along with Blockn_num, updates the miner's own chain, and the broadcasting of the block to other nodes through channel foundBlock using tempChain, after this the tempTx is resetted and Block_num is incremented, and its index is kept in the indexLongestChain variable.

The other case if a miner is in the process of mining and another miner beats him to solving the proof of work puzzle, in this case he stops mining, and adds the block that has been broadcasted to him through channel foundBlock using function addBlock() to blockDetails, update its own chain and sets the transaction that he was trying to put in a block back to Unconfirmed and transition back to the location Wait, in this location like the Mien location if a miner receives a block through channel founfBlock he adds it as well using function addBlock() and goes through the same procedure of updating its own chain and setting the transaction he was working on to Unconfirmed.

At location Wait if there are no unconfimed transactions in the transaction pool the transition that has !UnconfirmedTxPool guard will be taken. The length of each miner's chain is stored in variable lengthChain using function getLengthChain(), this variable lengthChain is used to determine whether a particular miner's chain will be orphaned, when the block number reaches the maximum HASHISIZE, a transition is taking comparing the length of a particular chain to the the length of the longest chain if it is shorter then this miner's chain will be orphaned and this information is kept in the local variable Orphaned. Figure 3.2 shows the flowchart of the miner.

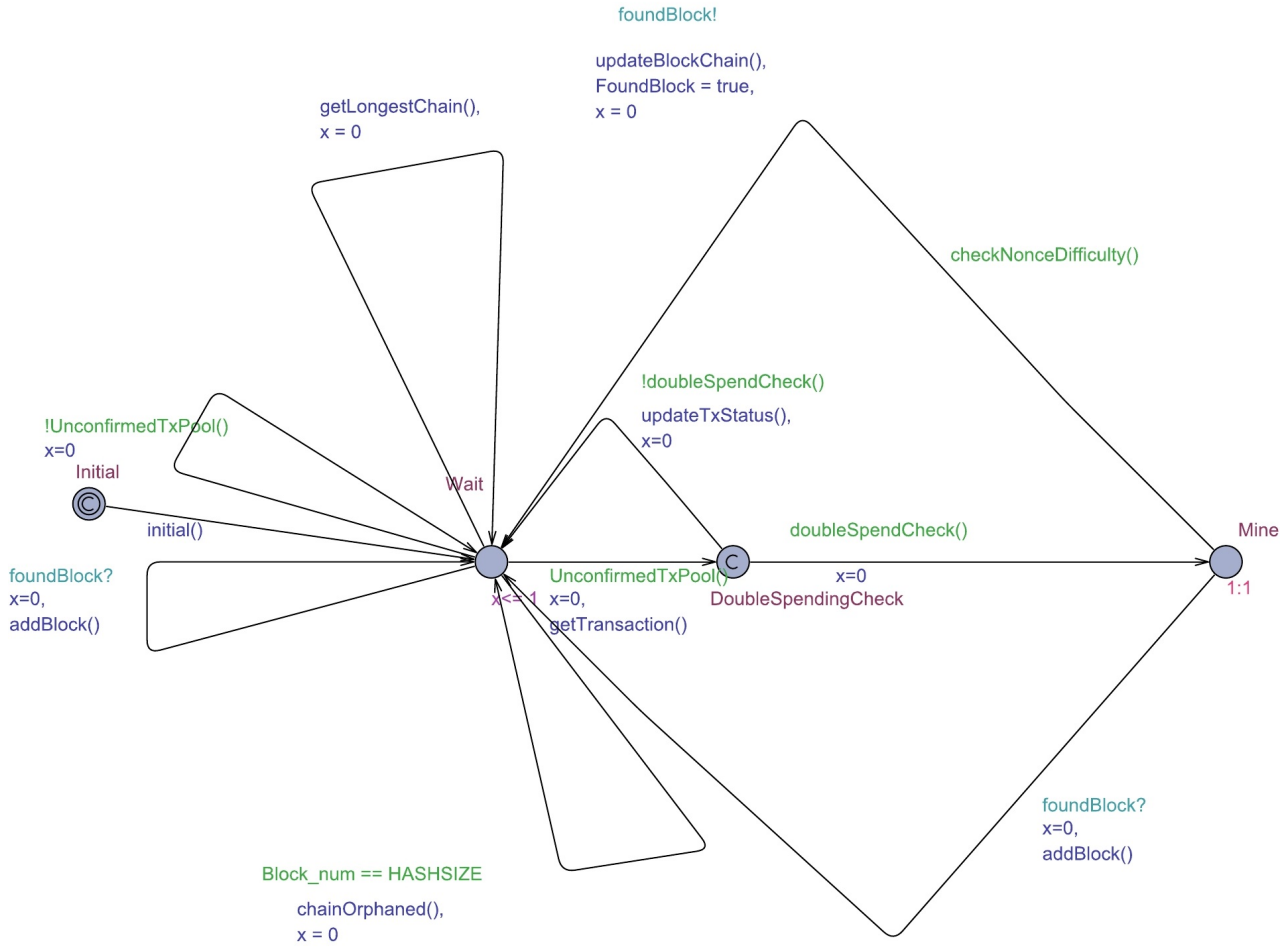

Figure 3.1: Automata of the miner

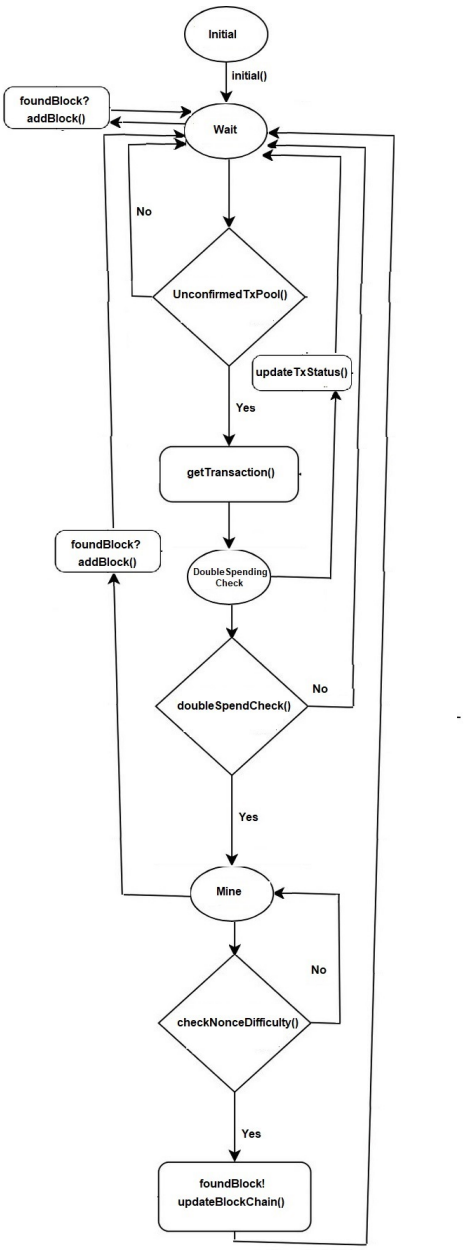

Figure 3.2: Flowchart Miner

3.2.6 Client

As shown in Figure 3.3 below, the client automaton also keeps a copy of the blockchain, and has a chain variable to store basic block information, and a blockDetails variable to store full block information such as transactions and their details.

The reason for a client to keep a copy of the blockchain is to periodilcally check if there is an output with that has its address and it is included in the chain, and this client has not claim it to its wallet yet.

The main job of a client is simply to create transactions, put them in the transactions pool, and update its wallet after these transactions are confirmed and included in a block.

The client automaton has 2 locations, the Initial location, and the Wait location. To initialize the clien's wallet and the chain that he keeps a copy of the blockchain in, a transition from Initial to Wait is taken. In order to create transactions each client will be given one transaction to start with, and 10 cryptocurrency coins to create more transactions by sending

and receiving these coins from one client to another.

After checking that the wallet is not empty and has coins in it, and that the maximum number of transactions MAX_TRANSACTIONS hasn't been reached by the transaction being made Tx_num, a client is selected through the select statement clientNumber:int[1,CLIENTS-1], which selects the client from the totality to send coins to.

The function create_transaction(clientNumber) takes the client number from the select statement as a parameter, and in the wallet's index looks for a positive amount that this client possesses, and finds an empty index in the transaction pool and places the made transaction with its number being Tx_num, and the input that has been taken from the client's wallet, and the receiving client address, and the amount sent, in the empty index of the transaction pool.

Once the transaction is made and the coins are sent, the transaction is set to 0 in the clients wallet , meaning he cannot use its input or spend it another time. The transition that has the guard !walletNotEmpty() verifies if there are any coins in the client's wallet, and the next transition is to check if there is any output that has this client's ID and has not been claimed in the wallet, if there is the function updateWallet will claim it and included in the wallet.

The transition with function updateTxPool() is used to update the transaction pool once a client creates a transaction and places this transaction in it. As said previously a client keeps track of the blockchain in order to claim any output that has its ID, and for that it can receive blocks through channel foundBlock and add it to its chain with function addBlock(), the same way as when a miner receives a broadcasted block and adds it. The last transition in this automaton is taken if the transaction number Tx_num is equal to the maximum transactions allowed MAX_TRANSACTION. The flowchart of the client is shown in Figure 3.4.

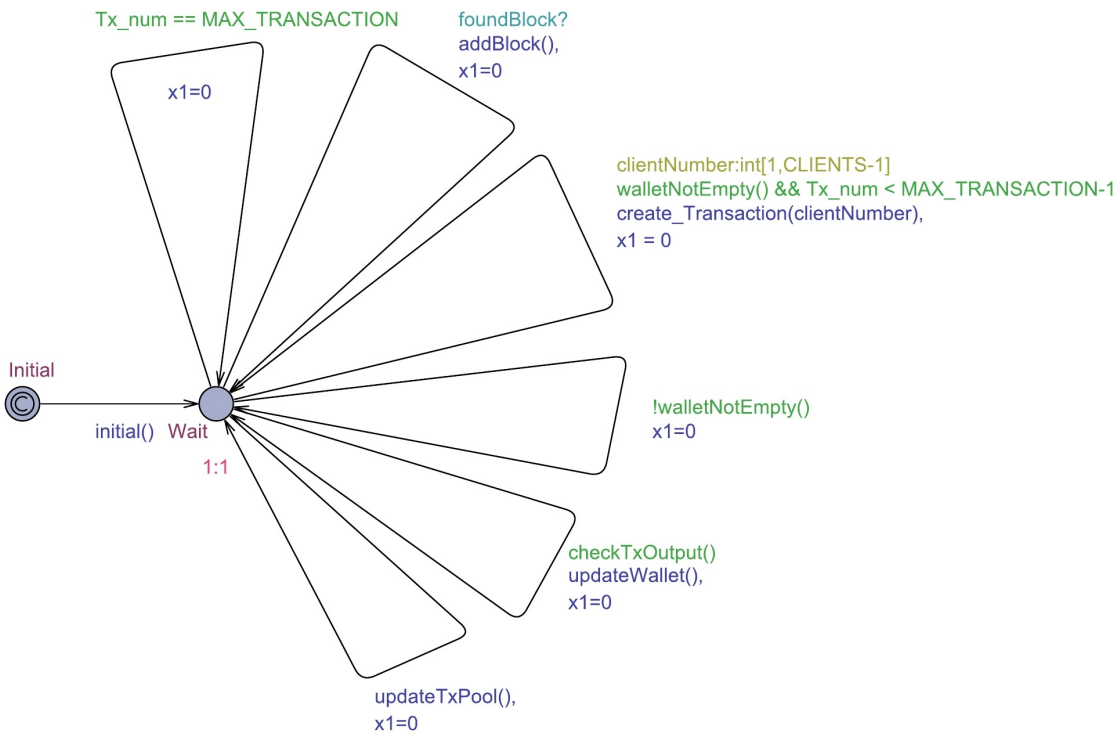

Figure 3.3: Automata of the client

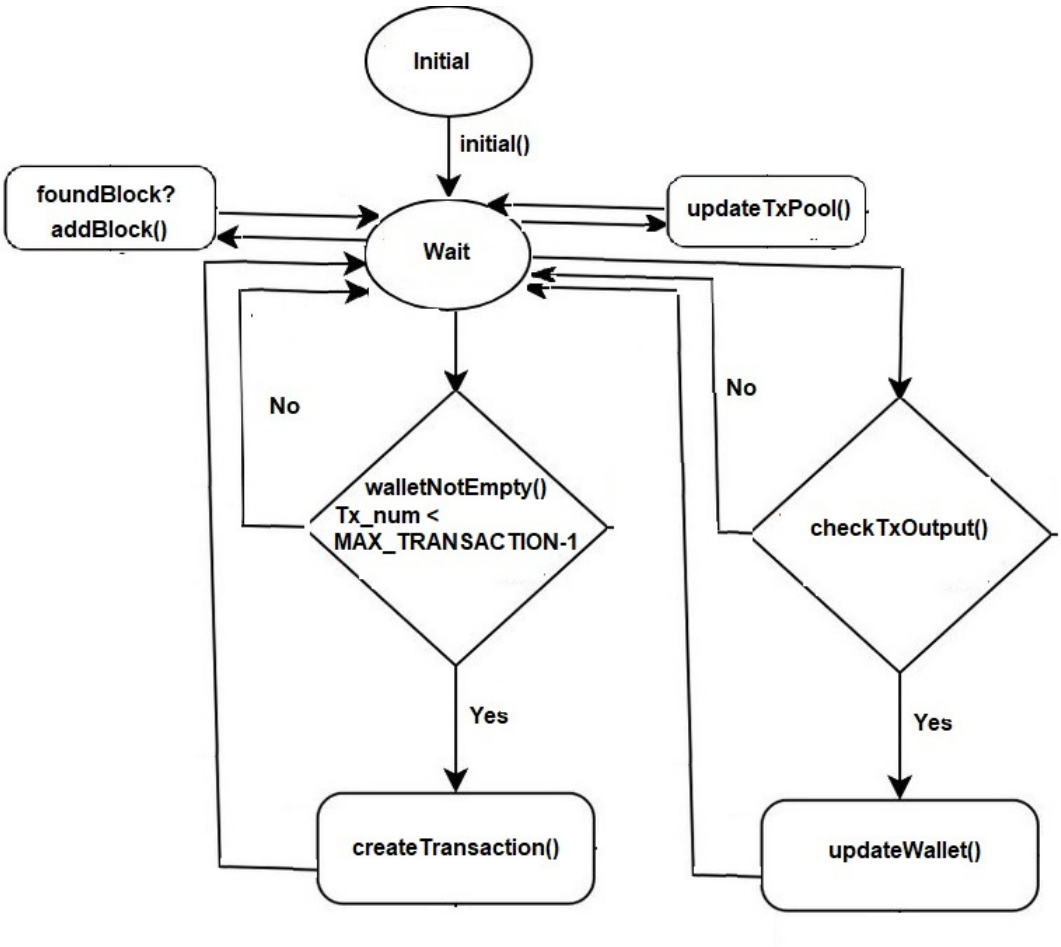

Figure 3.4: Flowchart Client

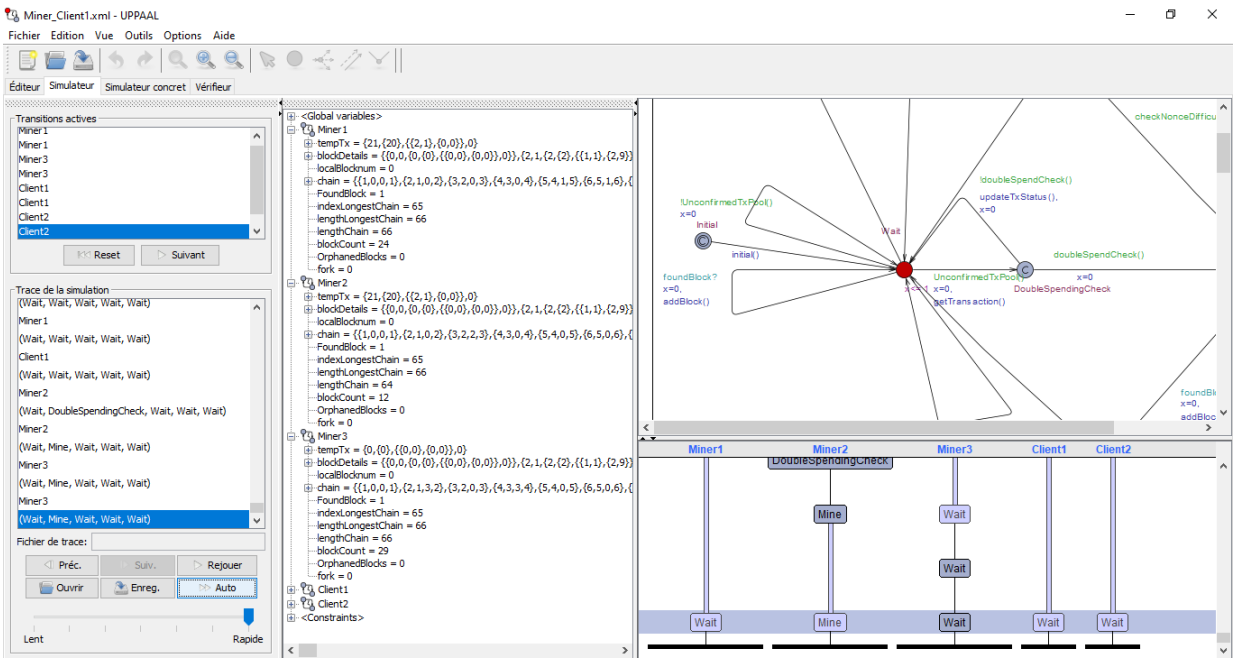

Figure 3.5: Building blocks in uppaal

As seen in Figure 3.5, all the needed information, is displayed on the interface, the length of the longest chain, the length of every miner's chain, the number of blocks mined by this

particular miner, and whether or not his chain has orphaned blocks, and if there is a fork within the blockchain.

3.3 Property verification

3.3.1 The probablity of Reachability

The reachability is the simplest form of properties, and it is the easiest one to verify, it demands if a state formula could be satisfied by any attainable state. We express that if a particular state satisfies the formula, meaning this state is reachable and accessible. For example in our Miner model, we choose a location to verify its reachability, the DoubleSpendingCheck state.

This verification could be done by this simple query in the Uppaal verifier: E<>Miner1.DoubleSpending which checks for any double spending attempts as shown in Figure 3.6.

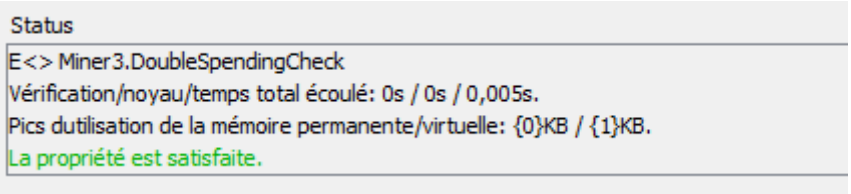

Figure 3.6: Property of Reachability for DoubleSpendingcheck state

3.3.2 The property of Safety

This property means in short words that a certain bad thing never happens, an example is the deadlock that must not accur in the model. To test this property we used the following query in the verifier: A[] not deadlock and the result is in Figure 3.7.

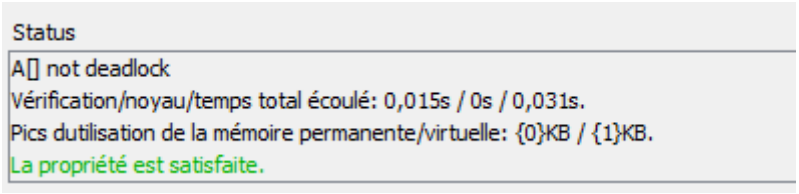

Figure 3.7: Property of safety

As shown in the figure 2 above, after a long time consuming verification of every possible state that this protocol could pass through, there is no deadlock detected for this protocol's model.

3.3.3 Property of Liveliness

This property express that something will end up arriving, sometimes or always. Liveliness could be verified through the query A<>Miner1.Mine, meaning that Miner1 will always do the mining process.

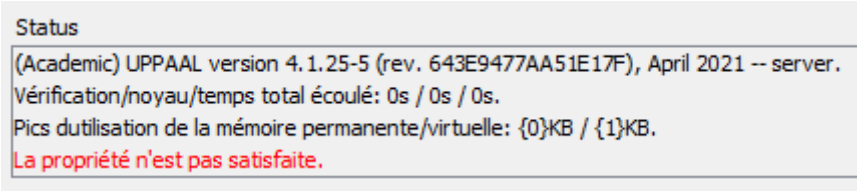

Figure 3.8: The property of Liveliness

In the Figure 3.8 above, We found out that the property is not satisfied, and it is logical because we have multiple miners and a single particular miner will not always handle the mining process. On the other side if we try the query E<>Miner1.Mine, we see that the property is satisfied meaning Miner1 will handle the mining process at least one time as shown in Figure 3.9.

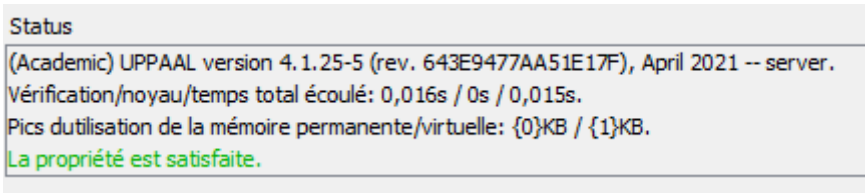

Figure 3.9: Liveliness at least once

3.3.4 Verification of qualifying properties

Using the Uppaal tool permits using the timed CTL logic. In this particular logic we are able to Add time constraints to our formulas in order for a more exact and precise verification, for example checking the following query's satisfaction for liveliness : if a particular miner could arrive to the mining state before a certain time limit x.

For the verification of this property, we will be using a clock x, which will count the time between the DoubleSpendingCheck state and the Mine State. We present the following TCTL formulas verified on our protocol:

1. E <> (Miner1.Mine and x <= 100) : this property is satisfied in our protocol, meaning the miner is able to finish the double spending check and begin the real mining process in a period of time that is lower or equal to 100.

   The following Figure 3.10 shows the satisfaction of this query.

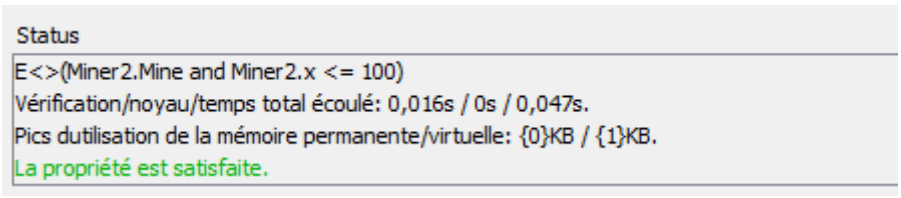

Figure 3.10: liveliness property for 100 time units at least once

2 A <> (Miner1.Mine and x <= 10) : this property is not satisfied in this protocol, which is completely normal, giving that sometimes the miner will encounter several invalid transactions or some double spending attempts and will not always pass from the DoubleSpendingCheck State to the Mine state in less than 10 time units.

The Figure 3.11 below shows the result :

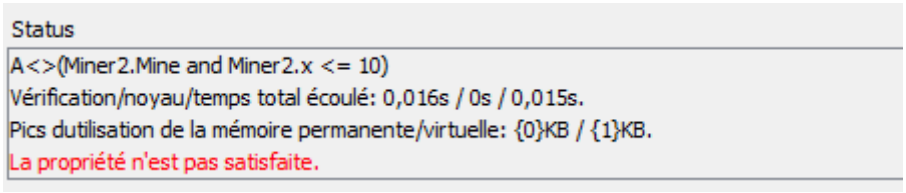

Figure 3.11: liveliness property for 10 time units always

3. A <> (Miner2.Mine or Miner1.Mine or Miner3.Mine and Miner2.x <= 10 or Miner1.x <= 10 or Miner3.x <= 10): on the other hand this property checks that at least one of three miners will always pass from the DoubleSpendingCheck state to the Mine state in a period of time that is lower or equal to 10.

   The Figure 3.12 below shows the result :

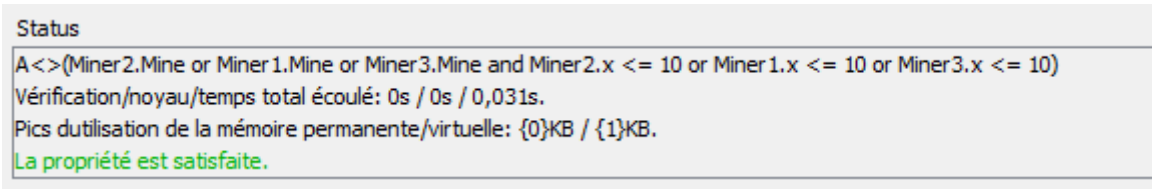

Figure 3.12: liveliness property for mining before 10 time units

3.3.5 Probabilistic verification of properties

The probability of a certain miner reaching the Mining state in a 100 runs could be verified using the following query : Pr[#<= 100] (<> Miner3.Mine) is shown in the Figure 3.13 below :

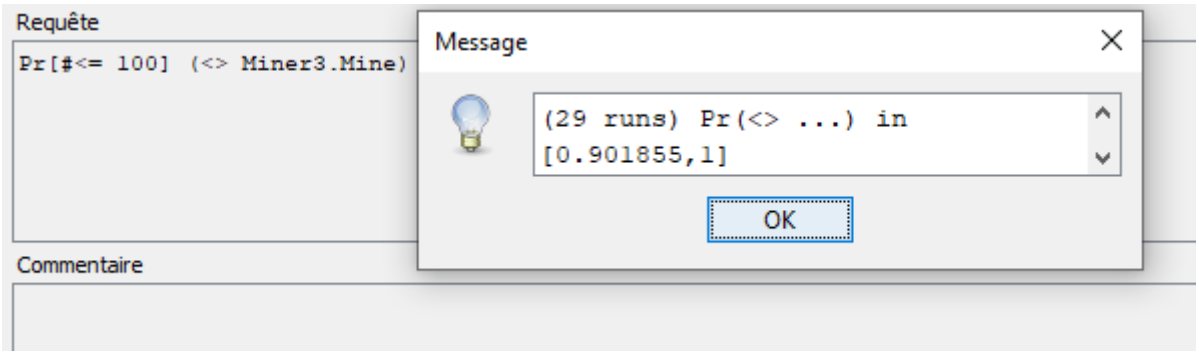

Figure 3.13: Probabilistic verification of Miner 3 reaching mining state in 100 runs

Uppaal also comes with different diagrams of probability distribution and density and others, in order for a more graphical representation of the the probability check as the one we have obtained above in Figure 10. We will be using the probability distribution for the previous query and the resulting diagram is shown in the following Figure 3.14 :

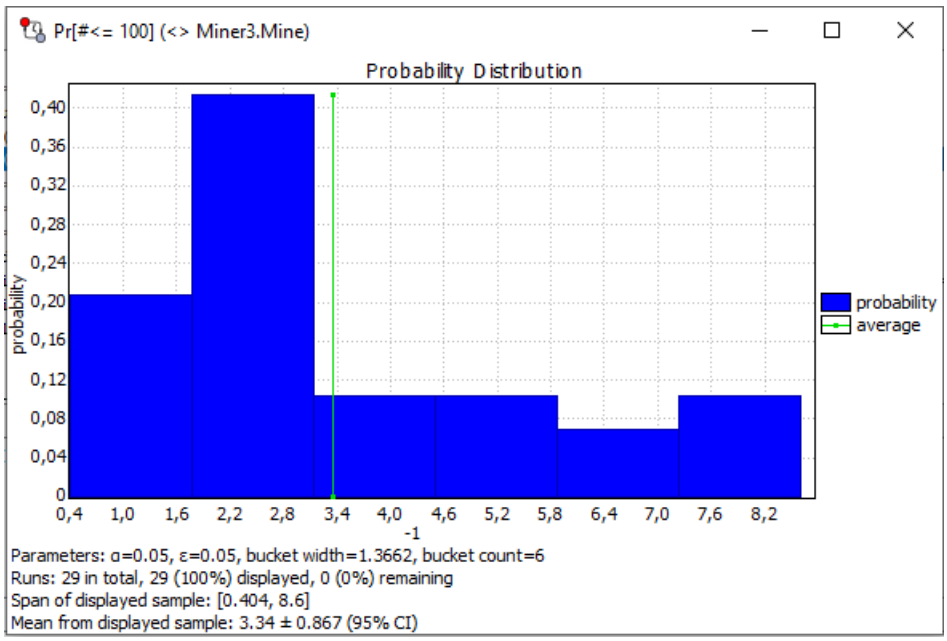

Figure 3.14: Probability distribution

The possibility of at least one of three miners will have orphaned blocks within their chains could be verified using the following query :
Pr[#<=100](<> Miner1.OrphanedBlocks == true or Miner2.OrphanedBlocks == true or Miner3.OrphanedBlocks == true).

From the the result shown in figure 3.15 below, we find that there is a small probability of 9% that at least one of the miners will have orphaned blocks within their chains, meaning the probability of a fork appearing is also 9%.

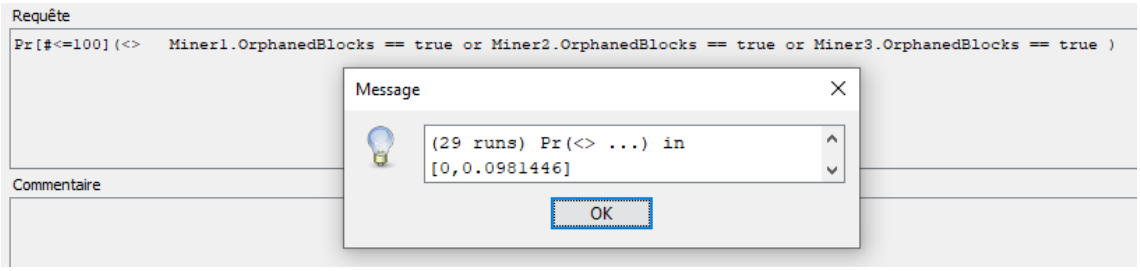

Figure 3.15: Orphan blocks probability

The probability of a certain miner building a block, not reaching the Mining state but actually finding a block could be verified using the following query : Pr[#¡=100](<> Miner2.FoundBlock==true
Figure 3.16 shows the result.

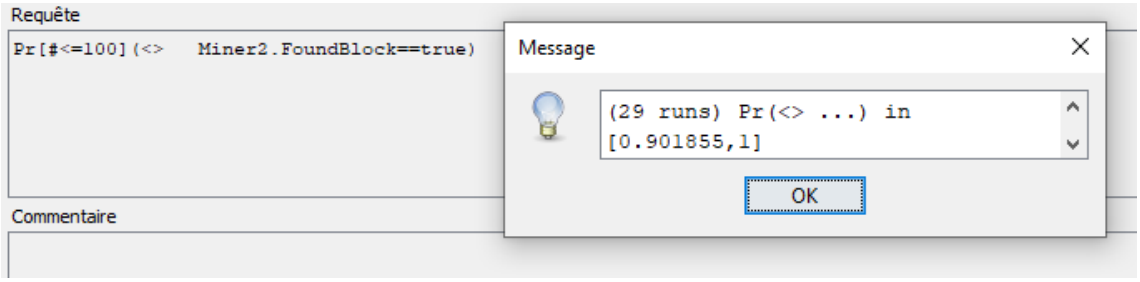

Figure 3.16: Probability of a miner building a block

The probability of miners passing the double spending check, meaning for all of them while verifying transactions there were no transactions that have been spent before is 90%. Figure 3.17 shows the above result.

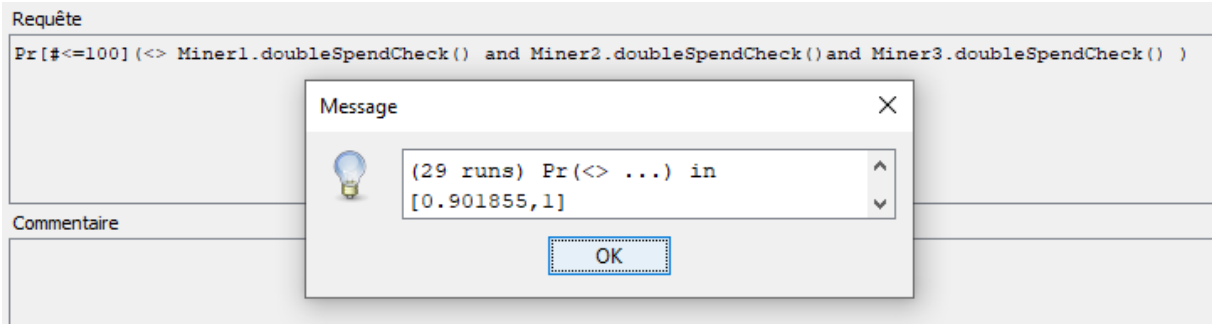

Figure 3.17: Probability of no double spending

The probability of a fork between the three miners, as shown in the Figure 3.18 below is very low, this due to the constant comparison between the length of every node's chain and the length of the longest chain recorded.

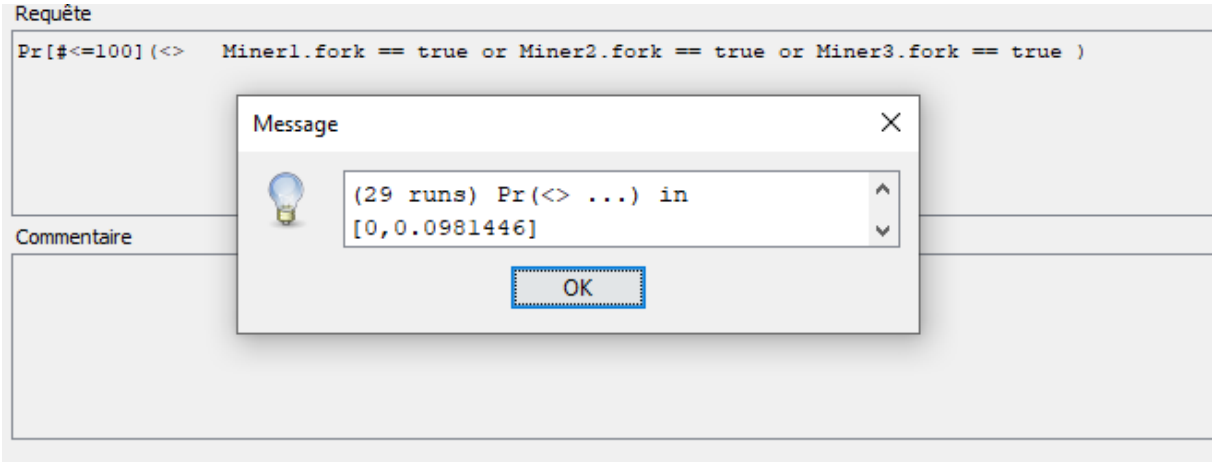

Figure 3.18: Probability of a fork

3.3.6 Property simulation

Each miner can reach the Mine state but not build a block, whether another miner has beat him to it, or he hasn't found the nonce yet. The following Figure 3.19 shows a simulation of miners reaching The Mine state.

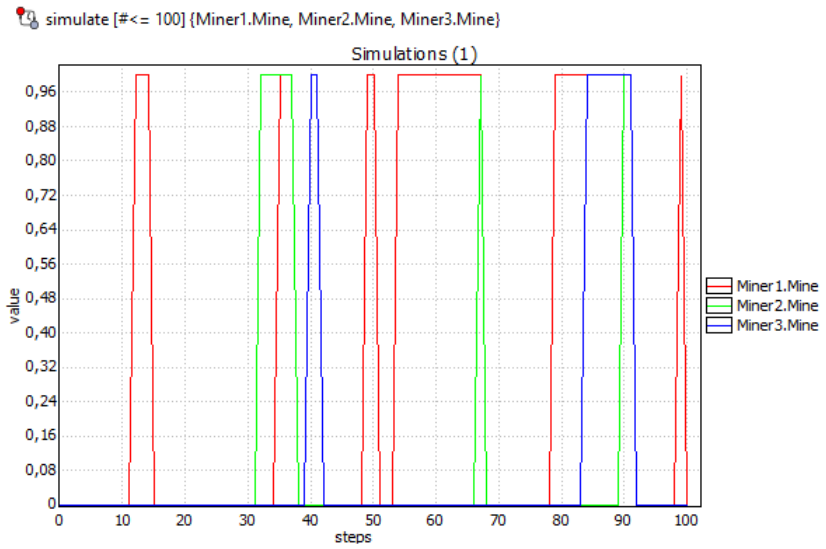

Figure 3.19: Simmulation of miners reaching Mine state

The blocks built by each particular miner in our model can be simulated by the following query : simulate [#<= 1000] Miner1.blockCount, Miner2.blockCount, Miner3.blockCount and the result is in Figure 3.20.

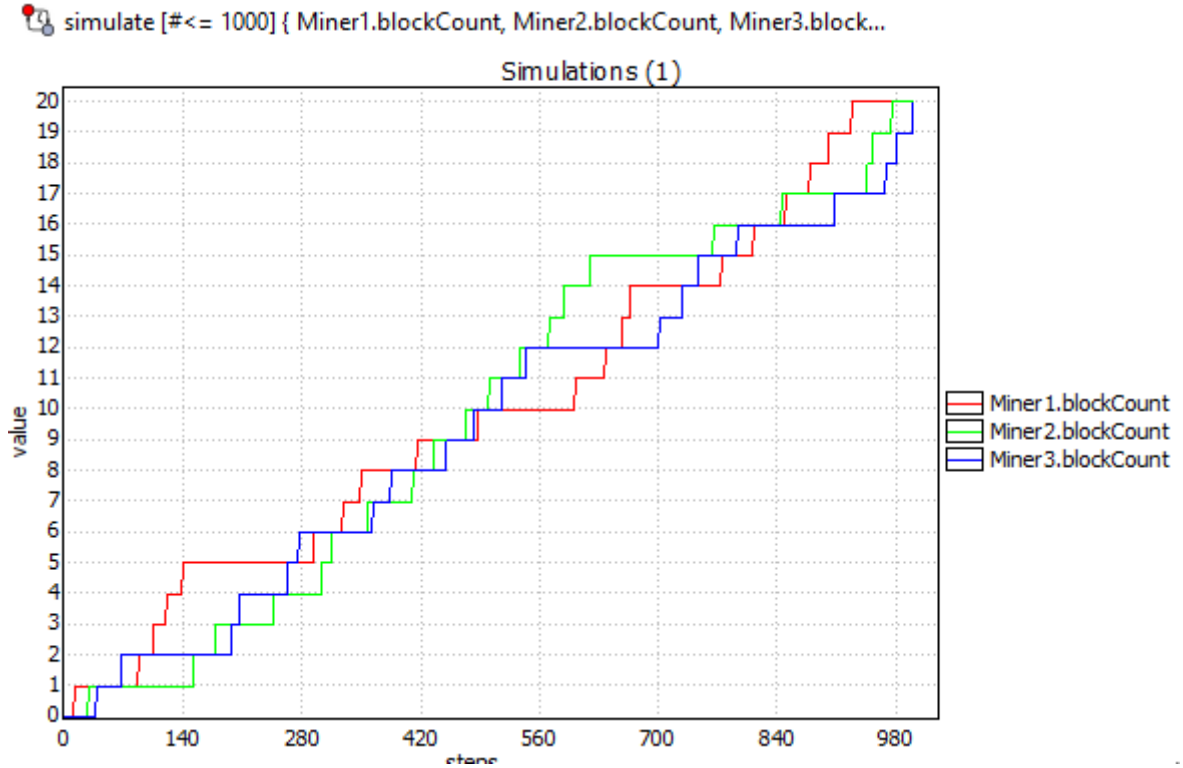

Figure 3.20: Simmulation of each miner's block count

3.3.7 Conclusion

In this final chapter we have seen the inner workings of both the entities miners and clients, how transactions are created by clients, and how they are used by miners to build blocks, how the process of building blocks is performed, all this was modeled in the uppaal tool and its properties were tested for satisfaction and some of them were simulated .

General conclusion and perspective

General conclusion and perspective

Blockchain study generally, and consensus protocols especially have been an important subject in our modern time. Therefore, many researches are investing their experience and time in order to exploit every possible angle of this technology, and make the most of it seeing how its potentials and uses are practically limitless. In this thesis, we have presented blockchain in chapter 1, and covered most of its aspects, and then we described some consensus protocols, and there are many so we only presented the most popular ones. In chapter two we picked one consensus protocol which is the proof of work protocol, presented its components, incentive and how it works, and we finished the chapter by presenting a related work to ours using the same tool, and a different consensus protocol.

In the third and final chapter we gave a description to how the proof of work will be modeled in the uppaal tool, gave the structures that we have used and presented the model for both the miners and the clients, after that we verified the most important properties that we have an interest in, simulated some others which concluded the third chapter.

From the work done here we came to a conclusion that the proof of work protocol, has well earned its reputation as being one of the most secured and integrity keeping protocols due to its hard solution to find but easy to verify approach, and as we have added double spending check and a fork eliminating method we came to a result that both forks and double spending are almost unexistable.

Therefore, it opens the possibility for some future work, to try and improve this protocol even more than it already is by lowering the difficulty level to minimize the resource consumption and still maintain the same security level which is what these consensus protocols are all about.

Bibliography

Bibliography

[1] Yang Xiao, Ning Zhang, Wenjing Lou, and Y. Thomas Hou. A Survey of Distributed Consensus Protocols for Blockchain Networks. IEEE COMMUNICATIONS SURVEYS TUTORIALS, 2020

[2] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. bitcoin.org .2008

[3] Sam Daley. 34 Blockchain Applications and Real-World Use Cases Disrupting the Status Quo. Builtin.com. August 2021

[4] Ledger. WHAT IS PROOF-OF-WORK. Ledger academy. October 2019

[5] Features of uppaal. Aalborg University in Denmark and Uppsala University in Sweden. March 2019

[6] Ren Zhang. Analyzing and Improving Proof-of-Work Consensus Protocols. ARENBERG DOCTORAL SCHOOL Faculty of Engineering Science. November 2019

[7] ivleen Kaur, Sheetal Chaturvedi, Aabha Sharma, Jayaprakash Kar, "A Research Survey on Applications of Consensus Protocols in Blockchain", Security and Communication Networks, vol. 2021, Article ID 6693731, 22 pages, 2021.

[8] Shijie Zhang, Jong-Hyouk Lee, Analysis of the main consensus protocols of blockchain, ICT Express, Volume 6, Issue 2,

2020

[9] Subhransu Sekhar Dash, Swagatam Das, Bijaya Ketan Panigrahi. Intelligent Computing and Applications Proceedings of ICICA 2019

[10] What are Blockchain Nodes. timesofindia.com. Dec 2021

[11] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang, Wen, and Dong In Kim. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. IEEE Access. Jan 2019

[12] Ayushi Abrol. What are Blockchain nodes? Detailed Guide.Blockchain council. 2022

[13] Abid Haleem, Mohd Javaid, Ravi Pratap Singh, Rajiv Suman, Shanay Rab, Blockchain technology applications in healthcare: An overview, International Journal of Intelligent Networks, Volume 2, 2021

[14] Rui Zhang, Rui Xue, and Ling Liu. 2019. Security and Privacy on Blockchain. ACM Comput. Surv. 1, January 2019

[15] Statistics. Energy concumption chart statista.com. 2021

[16] Paul Krzyzanowski. Blockchain and Bitcoin Cryptocurrency. Rutgers University. March 27, 2022

[17] Simanta Shekhar Sarmah. Understanding Blockchain Technology. Computer Science and Engineering 8(2): 23-29. 2018

[18] Stephan Leible, Steffen Schlager, Moritz Schubotz and Bela Gipp. A Review on Blockchain Technology and Blockchain Projects Fostering Open Science. frontiersin.org. 2019

[19] Blockchain: Research and Applications An official Journal of the Zhejiang University Press. 2020

[20] Bojana Koteska, Elena Karafiloski, Anastas Mishev. Blockchain Implementation Quality Challenges: A Literature Review. Sixth Workshop on Software Quality Analysis, Monitoring, Improvement, and ApplicationsAt: Belgrade, Serbia .Sep 2017

[21] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. IEEE. June 2017

[22] Leila Ismail and Huned Materwala. A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions. Symmetry 11, no. 10: 1198. 2019

[23] Dylan Yaga, Peter Mell, Nik Roby, Karen Scarfone. Blockchain Technology Overview. US National Institute of Standards and Technology. October 2018

[24] Gwyneth Iredale.Introduction To Permissioned Blockchains. 101blockchains.com. June 2019

[25] Geeksforgeeks. proof of work pow consensus geeksforgeeks.org. 2022

[26] Henrique Centieiro. Bitcoin Proof of Work. Level Up Coding. May 2021

[27] Shabnam Dhar, What is proof of work by theblockchaincafe.com. 2021

[28] Chen Zhao. Graph-based forensic investigation of Bitcoin transactions. Iowa State University. 2014

[29] Jimi S. How do blockchain mining and transactions work explained in 7 simple steps. blog.goodaudience.com. May 2018

[30] Niraj Dholakia. Modeling Proof of Stake Mining using UPPAAL. Rutgers university New Branswick. Dec 2016

[31] Ledger. What is proof of work. ledger.com/academy/blockchain. Oct 2019

[32] P. Shamili, B. Muruganantham, and B. Sriman. Understanding Concepts of Blockchain Technology for Building the DApps. ICICA 2019 (pp.383-394). 2019

[33] B. Sriman, S. Ganesh Kumar, and P. Shamili. Blockchain Technology: Consensus Protocol Proof of Work and Proof of Stake. ICICA 2019

[34] Guillaume Guérard, Zeinab Nehaï. Integration of the blockchain in a smart grid model, Léonard de Vinci Pôle Universitaire, Research Center, 2017

[35] Chen Zhao. Graph-based forensic investigation of Bitcoin transactions. Iowa State University 2014.

[36] Huaqun Guo, Xingjie Yu. A survey on blockchain technology and its security. Institute for Infocomm Research, A STAR, Singapore. June 2022

[37] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos and G. Das, "Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems," in IEEE Consumer Electronics Magazine, July 2018

[38] Michael Nofer. Peter Gomber. Oliver Hinz. Dirk Schiereck. Blockchain. Springer Fachmedien Wiesbaden. March2017