Blockchains
Subtitle

Guilherme João Bidarra Breia Lopes

Master's disseration planning
Engenharia Informática
(2nd degree cycle)

Supervisor: Prof. Doctor Simão Melo de Sousa

janeiro de 2023

Contents

Acronymms

NFT	Non-fungible token

Chapter 1

Introduction

1.1 Intoduction

Blockchains have gained significant attention in recent years for their potential to revolutionize various industries by providing a secure and transparent method for storing and transferring information and assets. At their core, blockchains are decentralized and permissionless ledger systems that leverage cryptography, consensus algorithms and State Machine Replication to maintain a shared history of all transactions on the network.

The increasing popularity of blockchains can be attributed to the growing demand for secure and transparent systems in various industries, particularly finance, supply chain management, digital identity and recently digital assets with the use of Non-fungible tokens.

In addition, the distributed nature of blockchains enables them to operate in a trustless environment, reducing the risk of single point of failure and ensuring the integrity of data stored on the network.

Consensus algorithms play a crucial role in the functioning of blockchains, as they determine the process by which transactions are validated, new blocks are added to the chain, and other details such as the selection of the next block producer.

Different consensus algorithms offer varying levels of security, scalability, and decentralization, and choosing the right algorithm for a specific use case is critical for the success of a blockchain network.

The main objectives of this dissertation are to compare different consensus algorithms for blockchains, evaluate the trade-offs between security, scalability, decentralization, and explore the potential for consensus algorithms to be plugged and changed on demand. Additionally, the dissertation will cover the implementation and testing of a blockchain network with a pluggable consensus algorithm and a thorough examination of the Tezos(REFERENCIA) blockchain as a testbed for consensus algorithms.

This dissertation will provide valuable insights into the field of blockchains and consensus algorithms, contributing to the development of more secure, scalable, and efficient blockchain systems, such as tools and methods to test consensus algorithms, to develop algoritms and to include them in already developed blockchain nodes.

1.2 Motivation

Despite the growing popularity of blockchains, there is currently a lack of tools and methods for easily swapping and testing different consensus algorithms in a live environment. This makes it difficult to determine which algorithm is best suited for a specific use case, particularly in the context of the blockchain trilemma, where it is challenging to achieve optimal

balance between scalability, security, and decentralization.

Additionally, developing consensus algorithms can be a complex and time-consuming task, and there is currently no standardized method for describing and implementing them. This presents a significant barrier to the adoption and evolution of blockchains, as it limits the ability of developers to experiment with new and innovative consensus algorithms. Also, the decentralization being one of the strenghts points of blockchain networks falls in the hands of a few selected developers.

This dissertation aims to address these challenges by providing a comprehensive analysis of different consensus algorithms for blockchains, and exploring the potential for consensus algorithms to be plugged and changed on demand. The results of this study will serve as valuable feedback for the development of a domain-specific language for describing consensus algorithms, and will provide insights into the most effective methods for testing consensus algorithms in a live environment.

The importance of this research cannot be overstated, as the ability to easily swap and test consensus algorithms is critical for the continued growth and evolution of blockchains. This will not only benefit the academic community but also industry stakeholders, who will be able to make informed decisions about which consensus algorithm is best suited for their specific use case.

In conclusion, this dissertation is motivated by the growing need for a comprehensive analysis of consensus algorithms for blockchains, and the desire to explore new and innovative methods for swapping and testing consensus algorithms in a live environment. The objective of this study is to contribute to the development of a standardized method for describing and testing consensus algorithms, ultimately advancing the field of blockchains and improving the security, scalability, and decentralization of blockchain networks.

1.3 Problem Statement

Consensus algorithms, being a fundamental aspect of blockchain technology, are subject to a range of emerging problems. Hardforks, network Denials of Service attacks, centralization of networks, and other relevant issues are challenging the stability and security of these systems. The choice between scalability, security, and decentralization, as described by the Blockchain Trilema, can lead to trade-offs in the design of consensus algorithms.

One of the challenges of consensus algorithms is the difficulty of changing the rules of the game after a blockchain network has been established. The hard forks of the Bitcoin, Bitcoin Cash, and Bitcoin SV networks have highlighted this issue. Similarly, the Ethereum network has also undergone several hard forks, leading to a split in its community.

Scalability issues are another challenge faced by some blockchain networks, including Bitcoin and Ethereum, which have struggled to accommodate increasing demand. In addition, some networks, such as Solana, have experienced outages, further highlighting the need for a more robust and reliable consensus system.

This dissertation aims to address these challenges by exploring the development of consensus algorithms and ways to test them. The Tezos blockchain provides a suitable platform

for this work, as it has a pluggable consensus system and allows for the implementation of different consensus algorithms. The goal is to learn how to develop consensus protocols in Tezos, develop the main entry points for a possible connection with a Domain Specific Language, and develop ways to test consensus algorithms. The first subject of such tests will be a proof-of-work consensus algorithm.

The question we want to answer in this dissertation is the following: How can we make a blockchain network that is flexible in terms of consensus algorithm selection and provides a platform for testing and comparison of different consensus algorithms? By addressing this question, we aim to provide a solution that will allow blockchain networks to be more scalable, secure, and decentralized, thereby addressing the challenges posed by the trilema. Additionally, this dissertation will serve as input/feedback for later work on a Domain Specific Language that will be used to describe consensus algorithms, making it easier to develop and test consensus algorithms in the future.

1.4 Document Organization

Chapter 2

Core Concepts and State of the Art

INTRODUÇÃO

2.1 Core Concepts

This section focuses on the fundamental building blocks of decentralized systems like blockchains are. This section covers State Machine Replication (SMR), Blockchain Data Structure and Networks, Network Models, and the Concept of Consensus. Understanding these concepts is crucial for understanding how blockchains networks function and the different trade-offs and design considerations involved in creating and operating a blockchain network. The section starts with an overview of SMR and how it provides consistency and availability in decentralized systems. Then, the concept of Blockchain Data Structure and its application in decentralized networks is explored. The different Network Models in distributed systems, such as synchronous, asynchronous, and partially-synchronous, are also discussed. Finally, the section concludes with a discussion on the importance of consensus in blockchain networks

State Machine Replication

State Machine Replication (SMR) is the concept of replicating data in a distributed system, where nodes in a system maintain the same state of a machine.

In this approach, a node broadcasts an update to its state using a total order broadcast, and all other nodes receive the same updates in the same order and apply it to their own state.

Total order broadcast refers to notion of broadcasting messages in a specific order, such that all nodes in the network receive the messages in the same order, even if the messages are sent from different nodes at different times.

The main advantages of using SMR, like mentioned before, are that it **can** provide both **consistency** and **availability** (in some cases it only provides availability), making it a popular choice for use in blockchain technology, databases, file systems, and other decentralized systems.

Consistency, because it guarantees that every node contains the same state of the machine (not necessarily at the same time), and guarantees availability because, in case of a node failure, there are other nodes in the system that have the information.

Overall, the concept of State Machine Replication is a fundamental component of decentralized systems that plays a crucial role in ensuring the consistency and availability of shared data. In case of Blockchains, to ensure that every node has the same chain and that there's no single point of failure of the whole system. This makes Blockchain Networks reliable, since

in case multiple nodes fail in the network, the network can still operate, and that no single node has the power over the whole information, making the network decentralized.

Blockchain Data Structure and Networks

Blockchain is a data structure made of blocks that is often compared to a linked list, where a blockchain is made of blocks (the nodes of the list) that are connected to a previous block in the chain by a reference to it, where that reference is usually the hash of the block it is pointing to.

That is, every block, except for the first block (sometimes called "Genesis" block), contains the hash of it's previous (or parent) block.

Once a block is added to the blockchain, it cannot be altered or deleted, making the ledger tamper-resistant and immutable, since, in order to change a block's information, one would need to change every child's block hash. It's also possible to say that, the older the block, the more "immutable" it is, or the bigger the number of child blocks a block has, the harder is the tampering of such block.

Since it is used to record transactions, a blockchain can be considered a "ledger", and it's transparent, as all transactions are recorded.

There are networks, like Bitcoin and Ethereum that store and replicate the blockchain (where the chain is the State Machine), with no central authority or entity controlling the network, like mentioned in the previous section. This makes the state of the blockchain highly available, since there are multiple nodes that contain it's information.

The network reaches consensus on the contents of a new block through the use of consensus mechanisms, such as Proof of Work or Proof of Stake. The blockchain is a distributed ledger, with all nodes in the network having a copy of the ledger, ensuring availability of the state.

The blockchain can be considered the state of a machine, where copies of the state of this machine is stored in multiple nodes, and newer states, that is, the process of appending a new block to the chain is replicated to every machine in the network.

Beyond cryptocurrency, blockchains have a wide range of potential applications in various industries, such as supply chain management, voting systems, and identity verification. The transparent nature of blockchains ensures that all transactions are publicly accessible and verifiable. The cryptographic functions used in blockchains, such as hashing and consensus mechanisms, provide a high level of security to the ledger and its transactions, and the replication of it ensures the availability of the information.

In conclusion, blockchain is a data structure that leverages the concepts of state machine replication to provide availability, leverages the concepts of cryptographic concepts to make it secure, imutable and tamper-resistant ledger where all transactions are recorded, making it transparent. The decentralized nature of blockchains networks and the use of consensus mechanisms ensure that all nodes in the network have the same state of the chain, providing the availability to access of this information.

Network Models

The concept of network types is a fundamental aspect of distributed systems and plays a crucial role in the design and implementation of these systems. There are three main types of networks in distributed systems: synchronous, asynchronous, and partially-synchronous. Each type of network has different properties and characteristics that affect the behavior of distributed algorithms and protocols, and it is important to understand these differences in order to design effective and efficient systems.

Synchronous networks are characterized by having a common notion of time and upper bounds on message delay. That is, in the Synchronous model, a finite time limit Δ is established and known. The adversary can only delay the delivery of a message sent by at most Δ time. This means that the network can guarantee that messages will be delivered within a certain amount of time and that all nodes in the network have a consistent view of the current time. This type of network is relevant when strict timing constraints are necessary, such as in real-time systems.

Asynchronous networks, on the other hand, do not have a common notion of time and do not guarantee upper bounds on message delay. In this type of network, messages may be delayed indefinitely, and nodes in the network may have a different view of the current time. Asynchronous networks are relevant when timing constraints are not strict, such as in data-centric systems. The unpredictability of message delays makes asynchronous networks more challenging to design and implement, but they are also more robust and can handle failures and network partitions more effectively.

Partially-synchronous networks are a hybrid of synchronous and asynchronous networks. They have a common notion of time but only provide partial guarantees on message delay. In this type of network, some messages may be delivered within a certain amount of time, while others may be delayed indefinitely. Partially-synchronous networks are relevant in scenarios where some timing constraints are necessary but not all. This type of network is often used in blockchain systems, such as Tendermint (used in Cosmos Network and others), and Tezo's blockchain take on this protocol, Tenderbake, which balance the need for speed and reliability with the need for resilience and robustness.

In conclusion, the concept of network types is critical in the design and implementation of distributed systems. Understanding the differences between synchronous, asynchronous, and partially-synchronous networks can help protocol designers make informed decisions about the type of network that best suits their needs and can ensure the success of their systems.

Consensus

Consensus in Distributed Systems refers to the process of achieving agreement among all participants in a network on the state of a shared database or system.

While State Machine Replication is the concept of broadcasting the update of a state, consensus is the concept of how the nodes replicate and decide on a replicated value.

One way to implement total order broadcast, that is, to broadcast messages or updates to

the state machine in a specific order, is by sending messages via a designated leader node. But if the leader becomes unavailable, the approach fails.

So nodes have to reach an agreement together, in a decentralized manner, on what is the next state of the replicated machine (or who's going to be the next leader), and not decided by a single node, and that's the coequally the concept of consensus.

Consensus protocols are **critical** to the blockchain context because they are the mechanisms that allow the network participants to agree on the state of the distributed ledger. In a blockchain network, transactions are submitted and processed by nodes in a decentralized manner, and the consensus protocol ensures that all nodes have the same view of the ledger and agree on which transactions are valid and should be included in the next block. Also, consensus algorithms enable the selection of the next leader, that is, the node that takes the transactions, builds the block and broadcasts the new block to the networks. This is important for maintaining the integrity and reliability of the blockchain, as well as for enabling trust in the network among participants who may not necessarily trust each other. A well-designed consensus protocol is essential for ensuring that the blockchain is able to process transactions efficiently and securely, even in the face of network failures, attacks, or other challenges.

Theorems

There are several important theorems related to consensus algorithms that are relevant to the context of Blockchain Networks, as these are used to attribute characteristics to this type of networks, which include the **FLP** Theorem and the **CAP** Theorem.

The **FLP** Theorem states that, in an asynchronous network where messages may be delayed but not lost, it is impossible to achieve both reliability and consensus in the presence of process failures. In order words, it's only possible guarantee two of the following:

- Finality or Agreement, that is, if (functioning) have to decide on a value, they all decide one specific one.
- Fault Tolerance or Integrity, the system still functions in case of node failures.
- Termination or Liveness, where all the functioning nodes decide on value.

On the other hand, the **CAP** Theorem states that, in an asynchronous network where messages may be lost, it is impossible for a distributed system to simultaneously provide the following:

- Consistency - all nodes see the same data at the same time
- Availability - every request receives a response, without guarantee that it contains the most recent version of the data
- Partition Tolerance - the system continues to operate despite arbitrary partitioning due to network failures.

These (and other) theorems provide limits and trade-offs in decentralized systems, and serve as a useful reference for designers and researchers in the field. In the following sections, more insight about these theorems will be provided and why they are relevant to the topic of Blockchain networks.

Permissioned and Permissionless Blockchains

Blockchains are a type of decentralized system that uses consensus algorithms to maintain the integrity of its data. In the context of blockchains, the terms “permissionless” and “permissioned” refer to the way in which participants in the network are allowed to participate in the validation of transactions and the creation of new blocks. These terms are critical in understanding the trade-offs between security, scalability, and decentralization in blockchains. In this section, we will delve deeper into what ”permissionless” and ”permissioned” mean and the advantages and disadvantages of each approach.

Permissionless blockchains, also known as public blockchains, are open to anyone and anyone can participate as a node. No central authority or entity is in control, making it a completely decentralized system. Permissionless blockchains are most often used for cryptocurrencies such as Bitcoin, Tezos and most of popular networks, where anyone can participate in the network and validate transactions, contributing to the security and reliability of the network.

Permissioned blockchains, also known as private blockchains, are restricted to a set of trusted participants. Only approved entities are allowed to participate as nodes and validate transactions, making it a partially decentralized system. This type of blockchain is often used in business environments where the participants are known and trusted, and where confidentiality and privacy are important considerations. Usually this networks are faster and more secure, since usually the number of participating nodes is way smaller and the fact that nodes are “handpicked” they are already trusted. Known examples are Hyperledger-Fabric, an enterprise-grade network and Binance Smart Chain, a cryptocurrency network that’s a fork of Ethereum.

Processes of Block creation

In the context of blockchain networks, there’s a need for nodes selecting the next update to the state, or the next block to add to the blockchain structure. At each round/heartbeat of the network, a proposer decides on the structure of the block (in a cryptocurrency blockchain, the proposer selects the transactions to include in the block).

In this subsection, we will explore the two main ways of reaching consensus/selecting a leader in blockchain networks: proof-based consensus and committee-based consensus.

Proof-based consensus protocols rely on the concept of proof of leadership. Nodes are selected to generate new blocks through a cryptographic random algorithm. The selection of the new leader/proposer is similar to the idea of a lottery. The node that wins such a lottery has the right to propose a new block. Nodes validate transactions, generate a Merkle tree for the transactions, and package them into a new block. The leader node broadcasts the new

block and its proof of leadership to the network. The network then validates the new block and if it is found to be valid, it is appended to the blockchain.

In **committee-based consensus** protocols, nodes vote to decide the next block to be appended to the blockchain. The proposer multicasts a preparation block request to other participants, who reply with their status. If the proposer receives a sufficient number of ready messages, it enters the pre-commit phase. Participants then broadcast their votes to commit the proposed block. If the number of commit responses agreeing to the new block exceeds a threshold, the block is appended to the blockchain.

Blockchain "trillema"

Another subject that is usually used to compare different blockchain networks, with much discussion is the topic of the "Blockchain trilemma". This "trilemma" refers to the trade-off between scalability, security, and decentralization. In other words, it is hard if not impossible to achieve all these three characteristics at the "same time" in a public/permissionless blockchain network:

- Decentralization: refers to the distribution of power and decision-making authority across all participants in the network. In blockchain, this means that there is no central authority or single point of control, allowing for a more equal and democratic system.

- Security: refers to the robustness and reliability of the network, ensuring that data and transactions are protected from tampering, hacking, or other forms of malicious activity.

- Scalability: refers to the ability of the network to handle a growing number of transactions and users without sacrificing performance or speed.

Even though that in the context o distributed systems, the topic of decentralization isn't relevant, when talking about blockchain networks it's important, as it's one of the main reasons why people use these systems in the first place.

For example, Bitcoin is a highly decentralized and secure blockchain, as there are many nodes (Around 40 thousand (TODO: Add citation)), yet lacks on scalability, since, the creation of a block takes 10min and a transaction takes about 1 to 1.5 hours to complete (TODO: Add citation).

Therefore, when designing a blockchain network, it is important to understand the trade-offs between scalability, security, and decentralization and to make trade-offs based on the specific goals and requirements of the network.

2.2 State of the Art of Consensus Algorithms in Blockchain Networks

2.2.1 Proof of Work

Proof of Work (PoW) is a consensus algorithm used by many blockchain systems, like Bitcoin and Litecoin, to secure the network and confirm transactions. Bitcoin or "Nakamoto Consensus" was the first instance of a Proof of Work Consensus.

The idea behind PoW is to solve cryptographic hard problems to create a block of transactions and add it to the blockchain. The solution to these problems requires significant computational power, which is provided by nodes called "miners". These miners compete to be the first to solve the cryptographic puzzle, and the winner is selected as the leader to create a block and add it to the blockchain.

In a permissionless environment, it's not possible to make a democratic system where each entity could cast a vote, since a malicious entity could create many fake identities or nodes to manipulate the network, known as a "sybil attack". PoW or "Nakamoto Consensus" was the first to be tolerant to sybil attacks in a permissionless setting. It prevents these attacks by making it computationally expensive for an attacker to try to disrupt the network, since an attacker would have to do as much work as the rest of the network, and that could imply problems to the malicious entity, like large electricity costs in case of mining.

By requiring a significant amount of computational power to mine blocks, PoW ensures that only legitimate nodes with real computational resources will be able to participate in the network.

Mining is what is called coequally to the process of using computational power to solve cryptographic puzzles and adding blocks to the blockchain.

It is used for several purposes in PoW, such as leader selection and preventing sybil attacks, like mentioned before, but also enforcing block timing with difficulty adjustment and in case of cryptocurrency blockchains, to add new value/mint new currency into the system.

In leader selection, the miner who solves the puzzle first becomes the leader to add the next block to the blockchain, and since one can only add blocks to the blockchain by solving the puzzle, this prevents sybil attacks, as, no matter a mallicious entity tries to create as many entities it can, this doesn't affect the mining. The difficulty adjustment ensures that the time between blocks is consistent, and the computational difficulty to mine new blocks adjusts accordingly. When in a epoch (in Bitcoin, that is 2016 blocks), the median time taken to mine a block was lower than it should've been, the difficulty is increased proportionally to make it harder to mine, and vice versa. The mining also introduces/mints new currency into the system, since the system doesn't have any value in it (it's a closed system), and that currency is awarded to the participants of the mining process.

Properties of Proof of Work Consensus

The properties of Proof of Work (PoW) consensus can be analyzed through the FLP theorem and the CAP theorem, which were discussed in the previous section.

Regarding the FLP theorem, PoW exhibits Probabilistic Finality. This means that all functioning nodes eventually agree on a single block, but the process is not deterministic and involves a certain degree of probability. The waiting for confirmations adds an additional layer of security to the consensus process. For example, in Bitcoin this takes about 6 confirmations (or 60 minutes (TODO: add reference)). This happens because mining is a process independent from the state of the network and independent from other nodes, and two or more nodes can solve the cryptographic puzzle/build a block "at the same time" (until the whole network agrees on the same block). When this happens, a node can receive more than one valid candidate extending the same chain, and each protocol handles this in a specific way. This is what is called a "Fork". For example, in Bitcoin, a node handles this case by using the rule of the longest chain (Longest Chain Rule), where, when a node has two competing blocks, where each makes a branch, it waits until one of the branch is bigger than the other. This is done because, the branch with the most computational power is the one with the bigger chance of being extended.

PoW also demonstrates Fault Tolerance, making it a form of Byzantine Fault Tolerance (BFT). This is achieved through the complex and expensive mining process, which makes it economically infeasible for a malicious node to alter the data in a block and catch up with the rest of the network. The incentive to follow the longest blockchain also adds to the fault tolerance of the consensus mechanism.

Finally, PoW also exhibits Termination, meaning that every functioning node reaches a decision, even if it's not the final one (because of the Probabilistic Finality).

Regarding the CAP theorem, PoW does not provide Consistent results, as a node might not have the latest block when requested, and like mentioned before, forks can happen, and so different nodes can contain different chains (for a period). However, it does provide Availability, as miners are continuously trying to mine new blocks, the difficulty level of mining is adjusted to ensure the steady addition of blocks to the blockchain and there are multiple nodes in the network. PoW is also Partition Tolerant, meaning that even if a portion of the nodes stop functioning or the network splits, the blockchain can still function and recover.

Nakamoto Consensus

In the previous subsection, we discussed the consensus mechanism used in blockchain technology where participants in the network compete to solve complex mathematical puzzles in order to validate transactions and generate new blocks. We introduced the concept of miners, who play a crucial role in this process, and how they are incentivized through rewards for their efforts.

Building on this foundation, we now delve into the specifics of the Nakamoto Consensus, named after the pseudonym used by the unknown creator(s) of Bitcoin. The Nakamoto Consensus is a milestone in the history of blockchain technology and a major innovation in the field of consensus algorithms.

In the Bitcoin network, miners are responsible for generating blocks by solving complex puzzles, which require a large amount of computational power. This acts as a barrier to entry for malicious actors, as it becomes extremely difficult for them to launch Sybil attacks by

generating many fake nodes. The puzzle in Bitcoin involves computing a nonce that meets certain conditions, as outlined by the equation H(hi−1,nonce,tx,par) < Target, where hi−1 is the hash of the previous block, tx represents the set of validated transactions, and par represents other parameters such as blockchain version and cryptographic parameters. The usual workflow of a node is the following:

- Node gossip transactions in the network (They are stored in a data structure that is called “Mempool”).
- When a node receives a block from the network, it appends it to it’s own blockchain stored.
- To start the mining process, a node selects multiple transactions for the transaction part of the block. There’s a size limit in bytes for this part, and the selection parameters are irrelevant, they only have to be valid.
- It creates a new block header with information, like, the hash of the previous block, a nonce (a integer number) and a Merkle Tree Root Hash of the transactions and other information (that is irrelevant to this).
- The node starts trying out every possible value for the nonce until the resulting hash of the header has a value bellow the current target.
- If a node receives a new block from the network before it finds itself the nonce, it stops the mining and appends such block. If it finds the nonce, it shares the block with the network

The difficulty of the puzzle is adjusted periodically (every 2016 blocks like mentioned before), depending on the actual block generation intervals and the expected block generation intervals of around ten minutes. This allows Bitcoin to maintain a consistent block generation time of around ten minutes, regardless of the computational power in the network.

Improving Proof of Work

The Nakamoto Consensus protocol, despite being the first consensus mechanism for blockchains and widely adopted, has limitations when it comes to scalability, security, and decentralization.

There are multiple ways of improving the Nakamoto Consensus protocol, and doing so has resulted in the creation of new blockchain networks. These changes move the consensus protocol within the triangle formed by the ”trillema”, but they also come with new challenges compared to the original consensus for blockchains.

This topic is relevant to this dissertation since it’s these characteristics differentiate consensus even further.

Improving Scalability

Blockchain networks have faced scalability issues since their inception. A consensus mechanism must maintain security and decentralization, so it’s challenging to scale the network

and improve the transaction processing speed. To overcome this challenge, several solutions have been proposed, including decoupling blockchain functions, parallel chains, and DAG-based protocols.

Decoupling of Blockchain Functions: The functions of blocks in a blockchain network can be separated into key blocks for leader election and microblocks for transaction packing. By doing so, the miner who successfully solves the puzzle becomes the leader of an epoch and generates key blocks and microblocks. This method helps to improve the throughput of the network, but it doesn't significantly reduce the transaction confirmation latency. Also, it comes with problems such as the fact that the leader can be compromised during the epoch.

Parallel Chains: The parallel chains method involves miners extending parallel chains simultaneously to improve the blockchain throughput. In this method, miners compete to solve puzzles, and the generated block is appended to one of the chains based on the random hash value of the puzzle solution. While this improves throughput, a malicious entity could still target a single chain.

DAG-based Protocols: instead of using a traditional blockchain structure, DAG-based protocols utilize a tree-like structure, called a Directed Acyclic Graph (DAG). The DAG structure allows for concurrent block generation and operates differently than traditional blockchain structures. Unlike parallel-chain protocols that have multiple independent genesis blocks at initialization, there is only one genesis block in DAG-based protocols. If the DAG-based blockchain follows the longest chain rule, there will only be one longest chain, instead of multiple chains in the parallel-chain scheme, and blocks on the forks will be pruned. This structure provides a unique approach to improving scalability in blockchain networks

Improving Security

Blockchain networks are vulnerable to various security attacks, such as selfish mining attacks, double-spending attacks, and liveness attacks. To improve the security of blockchain networks, various solutions have been proposed, including changing the incentive mechanism in the chain and how it's shared.

For example, some blockchain networks have adopted a new consensus algorithm that uses random incentives or that the reward of mining is shared.

Improving Decentralization

To improve decentralization, several solutions have been proposed, including de-incentivizing centralized activities like pool mining and incentivizing decentralized/solo mining using methods like eradicating ASIC mining.

Pool mining is a point of centralization in Proof of Work networks. The basic idea of pool mining is that there's a single node (connected in the network) that communicates with multiple miners, that is, entities which their only purpose is to find a nonce. These miners get together to mine a single block as if they were only a single node, so together they have more computational power, and inherently having a bigger change of finding a block. When

a block if found by the pool (the node, as viewed from the network), the reward is shared with all the miners, depending on how much effort they put into the mining. The problem comes with the fact that most mining nowadays in Bitcoin is done like this, and so, the owners of these pool nodes have a lot of power.

Also, with the increase of Hashing power, that is, the computational power of blockchains, in specific "Bitcoin", miners have improved the mining process by developing highly specialized hardware, called "Application-Specific Integrated Circuits" (ASIC).

By doing so, it's infeasible for anyone to join the mining process, and centralizing the power on the owners of ASICs. So there are multiple efforts to erradicate this type of mining like by using memory-hard puzzles, that are harder for ASIC hardware than for Personal Computer Hardware (like GPUs or CPUs), or using hash functions that are hard to implement as ASICs.

2.2.2 Proof of Stake

Chapter 3

Problem Statement, Experiments and Work Plan

3.1 The Problem

The development of blockchain technology has led to the creation of a large number of consensus protocols, each with its unique characteristics, from the selection of the leader, to the type of network, data structure, block structure, time between each block, and multiple ways to have a chain, to name just a few. Newer and innovative protocols are designed "everyday" with totally different characteristics compared to previous protocols. This abundance of choices has made it difficult for individuals or organizations looking to start their own blockchain to determine which protocol is the best fit for their needs, as there's a lot to account for.

Furthermore, there is no easy way to test and compare these consensus protocols in a live, non-simulated environment. There is no blockchain node software that allow for a seamless swap of the consensus algorithm, making it challenging to properly evaluate the performance of each protocol. This presents a major obstacle for researchers and developers looking to experiment with and improve upon existing consensus protocols.

Additionally, the process of designing and coding a new consensus algorithm is also challenging and requires a significant amount of expertise and resources. There is currently no platform that allows for the easy design, swap, and testing of new consensus algorithms.

All these problems contribute to the lack of standardization in the field of consensus protocols, a lot of attributes and characteristic to take in, making it difficult for individuals and organizations to adopt blockchain technology effectively. It is crucial to find a solution that allows for the easy and efficient comparison and testing of consensus protocols, as well as the design and implementation of new protocols. This will pave the way for the further development and widespread adoption of blockchain technology.

Bibliography