Chapter 4
BLOCKCHAIN CONSENSUS ALGORITHM

4.1. CONSENSUS ALGORITHM

4.1.1. Concept

A blockchain consensus algorithm is a set of rules and mechanisms that nodes in a blockchain network use to confirm and agree on the validity of new transactions/blocks, a mechanism that helps ensure all nodes (nodes) in the network agree on the current state of the digital ledger (ledger), thereby ensuring integrity and reliability as well as maintaining stability and security for the blockchain distributed system that No need for any intermediaries.

4.1.2. Working mechanism of the consensus algorithm

The consensus mechanism in blockchain operates based on the principle of consensus between nodes in the network. The main goal is to ensure all nodes agree on the current state of the ledger and confirm the validity of new transactions before they are added to the blockchain. How the consensus mechanism works includes:

Step 1. Send transaction: When a transaction is created, it is sent to the blockchain network.

Step 2. Confirm transaction: Nodes in the network (miners or validators) record the transaction and begin the confirmation process. The system will check if the transaction is valid. Such as does the sender have enough balance, is the transaction in the correct format,...

Step 3. Consensus algorithm: To achieve consensus between nodes in the network, a blockchain network uses a specific algorithm. Each blockchain may use a different algorithm. Some commonly used algorithms are:

- Proof of Work (PoW - Proof of Work)
- Proof of Stake (PoS - Proof of Stake
- Delegated Proof-of-Stake (DPoS - Delegated Proof-of-Stake)
- Proof-of-Weight (PoWeight - Proof-of-Weight)
- Historical evidence (PoH - Proof of History)
- Proof-of-Authority (PoA - Proof-of-Authority)
- Proof of Capacity (PoC - Proof of Capacity)

…

Step 5. Add block to blockchain: Once a transaction is confirmed and agreed upon, a new block containing that transaction is added to the blockchain. This block is linked to the previous block through a chain of hash codes, forming a continuous chain that cannot be changed.

Step 6. Update and spread: Once the new block is added, the information is updated and spread to all nodes in the network. This ensures that every node has the same copy of the blockchain, maintaining consistency and ensuring transparency.

4.1.3. Requirements of a Blockchain consensus algorithm

The consensus algorithm in blockchain is used to confirm and agree on the validity of new transactions/blocks. The consensus algorithm in blockchain must meet basic requirements to ensure the network operates stably, securely and efficiently. Therefore it must meet some of the following main requirements:

1. *Consistency:* All nodes in the network must agree on the state of the blockchain. Data in the blockchain must be consistent across the entire network.

2. *Integrity*: Only valid transactions are added to the blockchain. Each block must comply with the rules defined by the protocol. Transactions cannot be edited or re-added once they have been recorded on the blockchain.

3. *Decentralization:* The consensus process does not depend on a central entity. Any node can participate in authentication without asking for permission. This ensures transparency and eliminates risks from a single point of failure.

4. *Byzantine Fault Tolerance (BFT):* The system must operate correctly even if some nodes in the network fail or behave maliciously. The consensus algorithm needs to be able to cope with attacks from dishonest nodes.

This problem comes from a situation in the Roman army, where the Byzantine generals were just a group of generals of the army, these generals commanded troops stationed in different positions around a city of enemy. They communicated with each other through messengers, and the generals had to agree on a common battle plan. However, one or more generals may be traitors, trying to confuse others about the general plan. The problem is to find an algorithm that ensures that loyal generals can reach consensus.

5. *Security:* Blockchain must be protected from attacks such as:

- Double-spending: Spending a transaction twice,
- 51% Attack: When one entity controls more than 50% of the network power.

The consensus algorithm must ensure that no attacker can modify the blockchain or change the transaction state.

6. *Performance:* The consensus algorithm must be able to quickly and efficiently process large numbers of transactions. Transaction confirmation and block generation times must be optimized to ensure user experience.

7. *Scalability*: The system needs to support an increasingly large number of participating nodes without affecting performance. Blockchain must handle increasing transaction volumes while maintaining speed and security.

8. *Fairness*: Each node must have a fair chance to participate in the validation process and add new blocks. Nodes should not be excluded or favored based on geographical location or owned resources (unless explicitly stated as PoS).

9. *Finality*: Once a transaction has been confirmed, it cannot be undone or changed. This ensures the integrity and security of the blockchain.

10. *Energy Efficiency*: Especially important for new generation consensus algorithms (PoS, PoC). The algorithm should minimize energy consumption while ensuring security and efficiency.

So: *A consensus algorithm in blockchain needs to ensure: consistency, security, decentralization, high performance, scalability, and Byzantine fault tolerance. Depending on the application and goals, blockchains can prioritize certain factors to optimize operational efficiency..*

4.2. BYZANTINE FAULT TOLERANCE SYSTEM (BFT)

As with most distributed computing systems, cryptocurrency network participants need to agree on the current state of the blockchain, and that is what we call consensus. However, achieving consensus on a distributed network securely and reliably is not easy. So how can a distributed network of computer nodes reach consensus when processing a decision, if some of those nodes are likely to be untrustworthy? This is the fundamental question of the so-called Byzantine generals problem, from which was born the concept of Byzantine fault-tolerant systems.

The Byzantine Generals Problem was proposed by computer scientists Leslie Lamport, Robert Shostak and Marshall Pease in a scientific paper titled "The Byzantine Generals Problem" in 1982. [??]. This problem describes the resulting problem **consensus** in a distributed system, even if some components in the system do not function properly or intentionally introduce false information.

The problem is expressed metaphorically by the situation of a Byzantine army (Roman imperial army), besieging a city. The generals need to discuss to reach an agreement on the plan. In the simplest case, they agree on what should be done **attack** nice **withdrew**.

The issue of attacking or retreating is not important but the consensus of all generals, that is, agreement on a common decision to coordinate implementation. Therefore, we can consider the following goals:

- Each general must decide: attack or retreat (yes or no);
- The decision cannot be changed once made;
- All generals must agree on the same decision and proceed in sync with each other.

The communication problems mentioned above relate to the fact that a general can only communicate with other generals through messages transmitted by messengers. The central problem of the Byzantine generals problem here is that messages can be delayed, dropped, or lost. Additionally, even if the message is sent successfully, there is still a possibility that one or more generals may choose to take harmful action and send a false message to cause interference to other generals, leading to a Complete failure.

If we apply this problem to the presence of blockchain, each general will represent a network node and the nodes need to reach consensus on the current state of the system. In other words, the majority of participants in a distributed network must agree and take the same action to avoid a complete failure.

The mechanism of operation of the Byzantine fault-tolerant system

BFT's operating mechanism is based on consensus and information separation. When and only when the elements in the system reach consensus, a transaction or decision is made. And reaching consensus is not an easy story. BFT must use some more complex algorithms, this is to ensure all components in the same system will agree on the most accurate information before giving the final result.

The important point here is the separation of information in the system. The system does not need to achieve all consensus, but only some of it. The purpose of this action is also to prevent fraud by some elements. Another important aspect of BFT's working mechanism is the allocation of work and responsibilities to nodes. Each node in the network can be assigned a specific piece of work to ensure that no node has too much control over information or too much influence over overall decisions. This distribution helps minimize the risk from malicious nodes and ensures that even some nodes are hijacked.

BFT provides a high layer of security and reliability, so implementing and maintaining these fault-tolerant systems presents many challenges. These challenges require careful consideration and continuous development to ensure that BFT systems can meet the demands of real-world operating environments.

The BFT model is one of the core foundations for building secure and decentralized blockchain systems. Many modern blockchain projects have integrated or developed variations of this model to ensure greater integrity, security and performance.

In short, a Byzantine fault-tolerant system is a system that can solve the problem of the Byzantine generals problem. This means that the BFT system can continue to operate even if some nodes fail or perform harmful actions. There are many possible solutions to the problem of the Byzantine generals problem. Therefore, there are many ways to build a BFT system. Likewise, there are different ways for a blockchain to achieve a Byzantine fault-tolerant system, and what we have here are consensus algorithms. In the following sections, some consensus algorithms that have been implemented in practice are presented.

4.3. PROOF OF WORK ALGORITHM

Proof of Work (PoW) algorithm, the main principle of this algorithm is that network nodes must solve a complex cryptographic problem to find a valid hash function that satisfies the given condition. It requires nodes to spend huge computational costs (work). Some networks use this algorithm such as Bitcoin, Ethereum 1.0, ...

4.3.1. Historical development of the proof of work algorithm

In 1999, in [??] Markus Jakobsson and Ari Juels proposed the Proof of Work initiative, marking an important turning point in the cryptocurrency sector. This is a new way to authenticate transactions on a decentralized blockchain network. The initial idea was to build a system that operates on the inherent P2P (peer to peer) network platform. Jakobsson and Juels used a combination of hashing and Proof of Work to achieve decentralized consensus among nodes on transaction ordering. This system is called “Proof of Work”. This idea comes from the desire to decentralize the transaction authentication process as much as possible. This means that every participant can effectively confirm transactions without needing to access any central database. The idea was then refined and deployed on the Bitcoin network. On the Bitcoin network, miners need to solve a complex mathematical problem before adding transactions to the blockchain and receiving rewards. The problem is designed to consume a lot of computing power, making it difficult for a single individual to control the mining process. This prevents one individual from having the majority of hashing power and controlling transactions.

4.3.2. PoW working mechanism

Step 1: Create a new block

A node in the network (miner) prepares a new block to be added to the blockchain. The new block contains the following main components:

- List of valid transactions that have been authenticated.
- The hash of the previous block to link to the previous block in the chain.
- Nonce: A random number that will be searched to solve the hash problem.
- Metadata information: Includes block times and other information.

Step 2: Solve the hash problem

Each miner performs millions of hash calculations to find the nonce value such that the hash of the block satisfies the difficulty requirement (a certain number of zeros at the beginning of the hash string).

For example: If difficulty is 4, the hash must be of the form 0000abcd1234ef....

This step consumes computational resources because it is necessary to try many nonce values until the correct result is found.

Step 3: Distribute the results

The first miner to find a valid nonce value will broadcast the result to all nodes in the network. Distributed information includes:

- Block data.
- Nonce found it.
- Hash function result of the block.

Step 4: Verify the block

Other nodes in the network receive the information and check the block's validity:

- Check if the hash function is in the required format (with enough leading zeros).
- Check the correctness of transactions in the block to ensure there is no fraud such as double-spending.

If the block is valid, the nodes add it to their blockchain.

Step 5: Add block to blockchain

Once verified, the block is added to the blockchain chain. The blockchain chain will continue to grow from this new block as an extension of the main chain.

Step 6: Receive rewards

The miner who successfully solves the problem will receive:

- Block reward: An amount of new coins generated from the protocol.
- Transaction fees: Total fees from the transactions contained in the block.

Step 7: Adjust difficulty

The blockchain network automatically adjusts difficulty to maintain a stable average block generation time.

For example: In the Bitcoin network, difficulty is adjusted every 2016 blocks (~2 weeks), to maintain an average block generation time of 10 minutes.

4.3.3. Consensus mechanism in the Bitcoin network

Step 1: Create a new block

After validating transactions, a bitcoin node adds them to a memory pool where transactions wait until they are included to form a block. Suppose currently (at the time of writing this textbook) a node X has assembled a chain up to block 878,903. Node X is listening for transactions, trying to mine a new block, and also listening for blocks discovered by other nodes. When X's node is mining, it receives block 878,904 through the bitcoin network. The appearance of this block signals the end of the competition for block 878,904 and the beginning of the competition to create block 878,905.

During the previous 10 minutes, when node X was searching for a solution for block 878,904, it was also simultaneously collecting transactions in preparation for the next block. By this time, the node has collected a few hundred transactions in the memory pool. Upon receiving block 878,904 and successfully validating it, node X will compare the transactions in that block with all transactions in the memory pool and discard those that were included in block 878,904. The remaining transactions in the memory pool are unconfirmed transactions and are waiting to be written into a new block (878,905th block).

The X button will create a new empty block, block number 878,905. This block is called a candidate block, because it is not a valid block yet because it does not have a valid proof of work. This block only becomes valid if miners succeed in finding a solution to the PoW algorithm.

Note the first transaction in any block is a special transaction, called a coinbase transaction. This transaction was created by node X and contains a reward to the owner of node X if the block is successfully mined. Node X creates a coinbase transaction as a payment to the wallet of owner

Unlike regular transactions, coinbase transactions do not use UTXO (unspent output) as input. Instead, it has just a single input, called coinbase, which creates bitcoins from "Nothing". The coinbase transaction has one output, which is a payment to the miner's bitcoin wallet address.

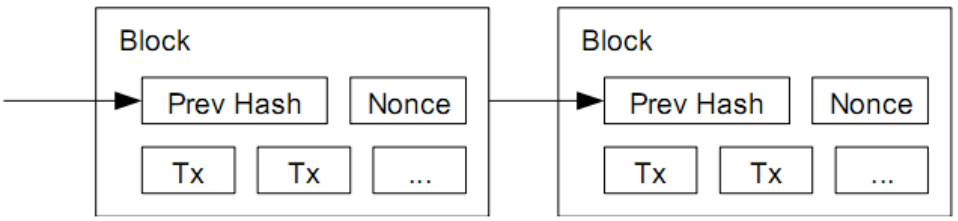

Figure 4.1. Illustration of the block structure in the blockchain

Tạo Block Header

A Block header containing no transactions will be 80 bytes in size. Includes information fields:

Size	School	Describe
4 bytes	Version (Version)	Version number used to track software/protocol upgrades.
32 bytes	Previous block hash code (Previous Block Hash)	Reference to the hash of the previous block (parent block) in the chain.
32 bytes	Merkle tree stump (Merkle root)	The hash of the Merkle tree root contains the transactions of the entire block
4 bytes	Timestamp (Timestamp)	Block creation time (in seconds since Unix Epoch).
4 bytes	Difficulty level (Target)	The goal of the PoW algorithm for this block.
4 bytes	Nonce	The counter is used in the PoW algorithm

The first field describes the version number of the block stored in 4 bytes, at the time block 878,903 was mined, the version number of the block was 0x2710e000. Describing the hash code of the previous block in 32 bytes, the hash code of the previous block of block 878,903 is:

000000000000000000001d043b2c9bd7f4bb4e41b1730381b87da0cae7b0343c2

Next are 32 bytes describing the hash of the Merkle tree root. A Merkle tree is a tree that describes the hash codes of the transactions in the block. The Merkle tree is constructed as follows: the leaves of the tree are the transactions, note the coinbase transaction is listed first, followed by the transactions added to the block. Calculate the hash code of each transaction (leaf), get the parent nodes of the leaves, next combine the 2 hash codes of 2 adjacent transactions to get the char hash code of those 2 hash codes, keep doing so. until we obtain the hash code of the root node. See the description of the Merkle tree in the picture 4.2 hereafter.

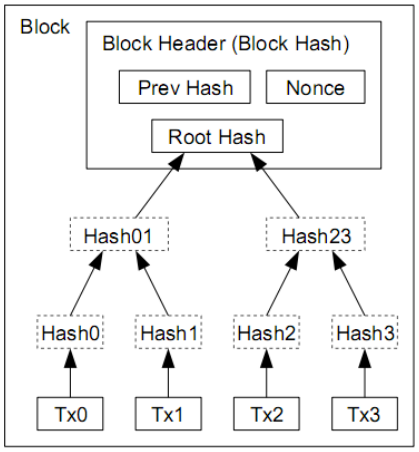

Image 4.2. Merkle tree structure

The node's timestamp is described in 4 bytes, encoded as Unix epoch time, based on the number of seconds that have elapsed since midnight January 1, 1970 UTC/GMT. The difficulty (target) of the PoW algorithm is stored in 4 bytes and finally the nonce number, initially initialized to 0. After all other fields have been filled in, the block header is complete and the process Mining is started. The goal now is to find a nonce value such that the hash of the block header is less than the target value. The mining node performs step 2.

Step 2: Solve the hashing (mining) problem

Once the candidate block is created by node X, it will mine the block to find the solution to the PoW algorithm to make the block valid. That is, at this point, node X must find a nonce value so that the hash of the block header is smaller than the target value.

Simply put, mining is the process of hashing a block header over and over, with the change of a single parameter, the nonce, until the resulting hash value satisfies a specific target. The result of a hash function cannot be determined in advance, nor can it be possible to create a pattern that can generate a specific hash value. This property of the hash function means that the only way to produce a hash result that matches the target is to try again and again, randomly changing the input until the desired hash result appears. .

A hash algorithm takes arbitrary input data and produces a fixed-length output. Such hashing algorithm is called **digital fingerprint** of input data. For a given input, the hash result will always be the same and can be easily calculated and verified by anyone using the same hash algorithm. With the SHA256 hash algorithm, the output is always 256 bits long, regardless of the size of the input data.

In the bitcoin network, the SHA256 hash function is used to find the hash value of the Block header that is less than the network's established target. The process of finding the nonce value to create a satisfactory Block header hash will take a lot of effort. Therefore, it is called proof of work (PoW).

So PoW must create a hash value smaller than the target. The higher the target, the easier it is to find a hash value smaller than the target. Conversely, the smaller the target, the more difficult it is to find a hash value smaller than the target.

Step 3: Distribute the results

When the nonce value is found and written to the block header, it creates the hash value of this block header. Immediately, node X transmits this block to all its peers. Those nodes receive the block, validate it, and then continue to propagate this new block to other nodes. As the block propagates throughout the network, each node adds the block to its own copy of the blockchain, increasing the new block height to 878,905. When mining nodes receive and validate this block, they give up searching for a block at the same height and immediately start calculating the next block in the chain using the block generated by node parent block". Miners who add node X's block to their blockchain have "voted" for X's block and the chain it extends.

Step 4: Verify the block

When the new block finds a node in the network that finds a nonce such that the block header's hash meets the difficulty of propagating it through the network, each node that receives the block performs a series of checks to validate the block. before continuing to propagate to other nodes. This ensures that only valid blocks are propagated within the network.

This independent validation ensures that miners act honestly and their blocks are added to the blockchain, thereby receiving rewards. Conversely, miners who act dishonestly will have their blocks rejected and not only lose their rewards but also waste the effort and electricity costs spent searching for PoW solutions without compensation. .

When a node receives a new block, it validates the block by checking against a list of criteria that must be met; otherwise, the block will be rejected. These criteria can be viewed in the source code **Bitcoin Core Client** via the CheckBlock and CheckBlockHeader functions, including:

- The block's data structure must be syntactically valid.
- The hash of the block header must be smaller than the target (ensuring PoW).
- The block timestamp cannot exceed two hours from the current time.
- The size of the block must be within allowable limits.
- The first transaction (and only the first transaction) must be a coinbase transaction.
- All transactions in the block must be valid.

Independent validation of each new block by all nodes in the network ensures that miners cannot cheat. In the previous sections, we saw that miners have the right to create a transaction to receive the new bitcoin reward created in the block while also collecting transaction fees.

So why don't miners create their own transaction with a reward of up to a thousand bitcoins instead of the correct reward? Because every node in the network validates blocks according to the same general rules. An invalid coinbase transaction will cause the entire block to become invalid, resulting in the block being rejected and the transaction never being recorded in the ledger. Miners are forced to build a perfect block based on common rules that all nodes follow and mine that block with a valid PoW solution. To do this, they have to consume a lot of electricity during the mining process. If you cheat, all your effort and electricity costs will be wasted. That's why independent validation is an important component of a decentralized consensus mechanism.

Step 5: Add block to blockchain

The final step in Bitcoin's decentralized consensus mechanism is to assemble blocks into a chain and choose the chain with the greatest total PoW effort. Once a node has successfully validated a new block, it attempts to assemble into a chain by connecting that block to the existing blockchain. Each node maintains three main sets of blocks:

1. Blocks are connected to the main chain.
2. Blocks form branches from the main chain (secondary chains).
3. Blocks that have no known "parent block" in the current chains (orphans).

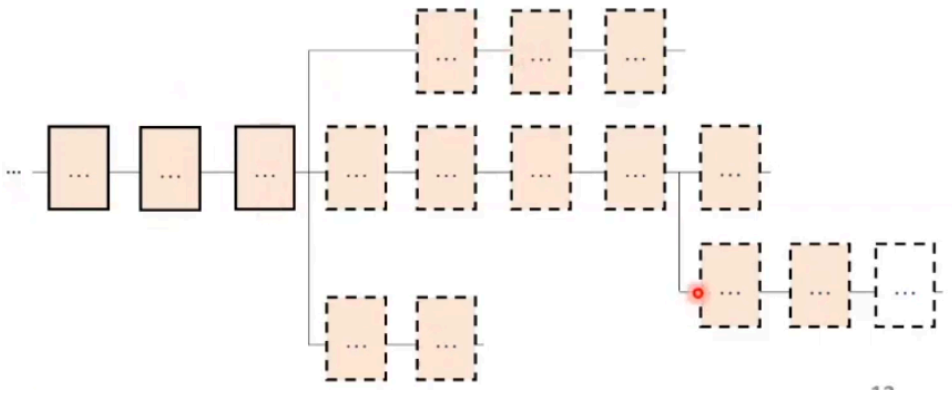

Figure 4.3. Illustration of blockchains in a node of a blockchain network

Invalid blocks are rejected as soon as they do not meet a certain validation criteria and are therefore not added to any chain.

The “main chain” at any given time is the valid blockchain with the greatest total cumulative PoW effort. In most cases, this is the chain with the most blocks (longest in

number of blocks), unless there are two chains of equal length but one has a greater total PoW effort. The main chain also has branches that contain "sister" blocks to blocks on the main chain. These blocks are valid but are not part of the main chain. These are kept for future reference, in case one of the branch chains expands and surpasses the main chain in total PoW effort.

When a node receives a new block, it tries to insert that block into the existing blockchain. The node will check the “previous block hash” field of the block, which is a reference to its parent block. The node will then try to find that parent block in the current blockchain.

The majority of the parent block will be the last block of the main chain, meaning new blocks will be added to the main chain. However, occasionally, a new block will be added to a chain that is not the main chain. In that case, the node will attach the new block to the side chain and compare the total PoW effort of that side chain with the main chain. If the sub-chain has a total PoW effort greater than the main chain, the node will switch to the sub-chain, meaning it will choose the sub-chain as the new main chain, and the old main chain will become the sub-chain.

If a valid block is received but the parent block is not found in existing chains, the block is considered an orphan. Orphaned blocks will be stored in the orphan blocks cache, where they will stay until the parent block is received. Once the parent block is received and linked into existing threads, the orphan block can be removed from the cache (orphan pool) and linked to the parent block, becoming part of the chain. Orphaned blocks typically appear when two blocks are mined almost simultaneously and are received in reverse order (child block before parent block).

By choosing the valid chain with the largest cumulative total PoW effort, all nodes eventually reach network-wide consensus. Temporary differences between sequences will be resolved as additional computational effort expands one of the possible sequences.

When miners mine a new block and extend the chain, that new block is their vote.

Step 6: Receive rewards

To create a coinbase transaction, node X first calculates the total transaction fees by adding up all the inputs and outputs of the transactions added to the block. Transaction fees are calculated according to the formula:

Total Fee = Total (Input) - Total (Output)

Next, node X calculates the correct reward for the new block. Initially the reward is set at 50BTC for mining a new block, after every 210,000 blocks the reward will be halved. So by mining the 878,905th block, node X will receive a reward of 3,125BTC.

The maximum number of halvings allowed is 64, so if 64 halvings are exceeded, the reward will be set to 0. Miners then only receive transaction fees when a new block is mined.

Step 7: Adjust difficulty

When creating an invitation block, the node's task is to find the number of nonce to make the hash of the block header smaller than the target. With the current difficulty on the Bitcoin network, miners have to try millions and billions of times before finding a nonce that creates a block header hash value small enough to satisfy the goal. We can see that increasing the difficulty by 1 bit will double the solution search space.

On average, the network needs to perform more than 1.8 septa-hashes (trillions of trillions of hash calculations) every second to find the next block. This may seem like an impossible task, but fortunately, the network now has the processing power of up to 3 exa-hashes per second (EH/s, equivalent to 3 trillion hashes per second), allowing to find a block in about 10 minutes on average.

As we have seen, the goal determines the difficulty and therefore the time to find a solution for the PoW algorithm. This leads to obvious questions: Why does difficulty need to be adjusted, who adjusts it, and how?

Bitcoin blocks are created on average every 10 minutes. This is considered the "beat" of the Bitcoin system, supporting the frequency of currency issuance and transaction processing speed. This frequency needs to be maintained stable not only in the short term but also over the course of decades. Over that long period, computing power is expected to continue to increase at a rapid rate. Furthermore, the number of people participating in mining and the computer systems they use will also constantly change. To keep the contouring time at an average of 10 minutes**,** Mining difficulty must be adjusted to compensate for these changes.

With the Bitcoin network, difficulty adjustment (retargeting) happens automatically and independently on each network node. Every 2,016 blocks, all nodes will readjust PoW. The governing equation measures the actual time it took to find the last 2,016 blocks and compares it to the expected time of 20,160 minutes. The ratio between the actual time and the expected time is calculated, and the target is adjusted (increased or decreased) accordingly.

- If the network finds blocks faster than 10 minutes/block, the difficulty increases (target value decreases).
- If the speed of finding blocks is slower than expected, the difficulty will decrease (target value increases).

The adjustment formula can be summarized as follows:

$$Target_{new} = Target_{current} \times \left(\frac{Actual\ number\ of\ minutes\ mining\ 2016\ blocks}{20160\ (minutes)}\right)$$

To avoid extreme fluctuations in difficulty, difficulty adjustments must be less than or equal to a factor of 4 for each adjustment cycle. If a target adjustment beyond a factor of 4 is needed, it will be limited to a factor of 4 and no higher.

Any further corrections will be made during the next adjustment cycle, as the imbalance will continue to exist through the next 2,016 blocks. Therefore, the large difference between computational power and difficulty may require several 2,016 block cycles to be fully balanced.

The difficulty of mining a bitcoin block is set so that it takes the entire network approximately 10 minutes to process a block, based on the mining time of the previous 2,016 blocks and adjusted every 2,016 blocks. This is achieved by decreasing or increasing the target.

Note that the target does not depend on the number of transactions or the value of the transactions. This means that the amount of computing power, and therefore the amount of electricity consumed to secure the Bitcoin network, is completely independent of the number of transactions.

4.3.4. Advantages and disadvantages of PoW consensus algorithm

4.3.4.1. Advantages of PoW

- **High security**: PoW requires miners to solve very difficult, but easily testable, cryptographic problems. Changing the data in a block changes the entire linked hash chain, forcing the attacker to re-solve the entire chain, which requires enormous computing power. A 51% attack is theoretically possible but in reality is very difficult with large networks like Bitcoin.

- Decentralization: PoW helps ensure that no central entity controls the network. Anyone with a computing device can join the network and become a miner without needing permission from a third party.

- Anti-fraud and double-spending: PoW ensures that transactions cannot be overwritten or modified once they have been added to the blockchain. Prevents double-spending because confirmed transactions become a permanent part of the blockchain.

- Immutability and transparency: Once data is written to the blockchain via PoW, it is almost impossible to change or be deleted. The PoW blockchain makes all blocks public, allowing users to transparently audit and verify them.

- Simple in operating mechanism: PoW only requires solving cryptographic hash calculations (in the Bitcoin network, SHA-256), making it easy for participating nodes to check validity without the need for complicated mechanisms.

4.3.4.2. Disadvantages of PoW

- Energy consumption: Solving the hash problem requires enormous computing power and consumes a lot of electricity.

- Strong hardware requirements: Miners need to use specialized equipment such as ASICs to compete in finding nonce values, making hardware investment costs very high. It is difficult for ordinary users to participate in mining due to insufficient hardware and resources, leading to a focus on large mining pools.

- Centralization ability: Large mining pools can concentrate computing power and dominate the network. Mining pools – where many miners cooperate to share rewards – can reduce the decentralization of the network. Large mining pools can take up the majority of computing power, posing a risk of centralization.

- Low transaction processing speed: Due to the fixed block generation time (average 10 minutes/block in Bitcoin), the number of transactions that can be processed per second (TPS) is limited. Bitcoin can only process about 7 transactions per second (TPS), while centralized payment systems like Visa can process tens of thousands of transactions per second.

4.4. PROOF OF Stake (POS) ALGORITHM

- Proof of Stake (PoS - Proof of Stake), instead of using computing power like PoW, PoS chooses validators based on the number of coins (tokens) they stake in the network. People who hold more coins (or stake longer) have a higher chance of being chosen to create a new block. Some networks use this algorithm such as Ethereum 2.0, Cardano, Solana, Algorand, ...

4.4.1. Development history of the proof-of-stake algorithm

One of the main issues we need to consider with the protocols of PoW-based blockchains is the energy required to execute them. Currently, the electricity consumed to maintain the Bitcoin blockchain network is comparable to that of a small country. This

situation has prompted research into alternative blockchain protocols, which aim to eliminate the reliance on PoW by replacing it with a mechanism that is more energy efficient but still provides similar guarantees. [??].

The concept of PoS has been widely discussed on the Bitcoin forum.
Blockchain designs based on proof of stake have been formally investigated by Bentov et al., both when combined with proof of work (PoW) and when PoS is the sole mechanism for a blockchain protocol. [5]. Although Bentov et al. have shown that their protocols are secure against certain types of attacks, they do not provide a formal model for analyzing PoS-based protocols or security proofs. Confidentiality is based on precise definitions. Many PoS-based blockchain protocols have been proposed (and implemented) for several cryptocurrencies, however because they are based on empirical security arguments, these cryptocurrencies are often discovered There are shortcomings from a security perspective.

The idea of PoS was first proposed on the Bitcointalk forum in 2011 by some developers in the Bitcoin community such as QuantumMechanic. The initial goal of PoS is to reduce the energy consumption caused by PoW mining and create a consensus system that does not depend on computing power. However, at that time, this idea was only at a theoretical level and had not been implemented in practice. PoS is applied for the first time in the Peercoin (PPC) blockchain network. This is a blockchain network developed by Sunny King and Scott Nadal in 2012. Peercoin uses a hybrid mechanism between PoW and PoS to ensure initial security and gradually transition to complete PoS. In 2014, the Blackcoin blockchain network was born and was the first blockchain to completely switch to PoS without using PoW after the genesis block. In the period 2015-2018, the birth of the Ethereum blockchain network was developed by Vitalik Buterin. In 2015, Ethereum launched as a PoW blockchain but planned to transition to PoS from the first versions. In 2017, Ethereum introduced a plan to upgrade Casper, a PoS algorithm to replace the PoW mechanism in the Ethereum network and also in 2017 the birth of the Cardano blockchain network, Cardano was developed by Charles Hoskinson, co-founder of Ethereum . This is a blockchain network that uses PoS from the beginning with the Ouroboros consensus algorithm, designed to enhance security and performance while still saving energy. In 2022, the Ethereum network officially switches completely from PoW to PoS with "The Merge" event. This is an important milestone, making Ethereum the largest PoS network in the world. Switching to PoS has reduced the network's energy consumption by more than 99%. Pave the way for other upgrades like sharding to increase scalability. The period 2020 up to now has been a boom period for modern PoS blockchains. New blockchains like Polkadot, Tezos, Solana, Avalanche have adopted variations of PoS from the beginning, bringing fast processing speeds and low transaction fees. Modern PoS platforms incorporate features such as Delegated Proof of Stake (DPoS), Proof of History (PoH) to increase performance while ensuring decentralization central and confidential.

4.4.2. PoS working mechanism

Step 1: Staking assets (Staking)

Users who want to become validators must deposit a certain amount of coins into the blockchain network. These coins act as a commitment and "collateral" to participate in the validation process.

Depending on the network, the amount of coin stake required may vary (e.g. Ethereum 2.0 requires 32 ETH).

Step 2: Validator Selection

The system selects validators to validate new blocks based on the following criteria:

- Number of coins to deposit: Validators with more coins to deposit have a higher probability of being selected.
- Deposit time: The longer a Validator deposits, the greater the chance of being selected.
- Randomization: Some PoS protocols apply randomization to ensure fairness in selection.

Step 3: Block Creation and Transaction Validation

The selected validator will verify the transactions in the new block and ensure that all transactions are valid and do not violate network rules (e.g. no double-spending transactions). If all conditions are valid, the validator will add the new block to the blockchain and broadcast the block to other nodes in the network.

Step 4: Consensus Verification

Other nodes in the network will check and confirm the new block. If the block is confirmed to be valid, the block will be added to the official blockchain.

Step 5: Receive rewards and trading fees (Reward Distribution)

A validator that successfully creates a block will receive a reward of the total transaction fees from the transactions in the block and depending on the protocol, the validator may receive additional newly generated coins.

Step 6: Slashing mechanism

If a validator intentionally creates the wrong block or submits invalid data, they will be penalized through a slashing mechanism. Validators who create wrong blocks or submit invalid data will lose part or all of their staked coins, and may be excluded from participating in the validation process for a period of time. This helps ensure that validators will always behave honestly to avoid losing assets.

4.4.3. Cardano network consensus mechanism

Cardano is a public blockchain platform built on academic research and scientific methodology. It was created to address the limitations of Bitcoin and Ethereum 1.0 in terms of scalability, interoperability, and sustainability. Cardano was started in 2015 by Charles Hoskinson, who is one of the co-founders of Ethereum. The core technology of Cardano is the use of an advanced PoS consensus protocol called Ouroboros and announced in 2017. It is developed using the Haskell programming language, a programming language that allows Cardano to achieve security. high security and stability. Ouroboros is the name of an ancient symbol of a snake eating its own tail, symbolizing regeneration and eternal circulation. The Ouroboros consensus mechanism allows the Cardano network to securely and efficiently reach consensus on the state of the distributed ledger. Ouroboros has been developed through 5 versions.

- First version (Ouroboros Classic): This is the first version of Ouroboros, announced in 2017. It is one of the first PoS algorithms to be researched and theoretically proven in terms of security and decentralization. Ouroboros Classic establishes a consensus mechanism by dividing the network into time cycles (epochs) and smaller slots (slots), where "slot leaders" are randomly selected from ADA stake holders (stakeholders). Ouroboros Classic expands blockchain capabilities without requiring PoW, saving energy.

- 2nd edition (Ouroboros Praos): This version was developed as an upgrade of Ouroboros Classic and announced in 2018. Ouroboros Praos improves the security of the system by using a randomization model initialized in each time cycle (epoch). Slot leaders are chosen not only based on stakes but also according to an additional random factor, making the system more secure against cyber attacks. Ouroboros Praos improves security and resistance to Sybil attacks and threats from high-stakes attackers.

- 3rd Edition (Ouroboros BFT (Byzantine Fault Tolerant)): This version was launched in 2019 and is an important innovation of Ouroboros Praos, aimed at enhancing error resistance and improving performance. Ouroboros BFT applies Byzantine Fault Tolerance (BFT) techniques, which help the system maintain consensus even in the presence of some dishonest or inactive nodes. This allows the network to continue operating normally even if some nodes fail or are attacked. Ouroboros BFT offers greater security, strong error recovery, and reduced transaction completion time.

4th Edition (Ouroboros Omega): This is the next version, researched and developed after Ouroboros BFT, with the goal of further optimizing security and scalability. Ouroboros Omega is designed to enhance Cardano's scalability, especially in large-scale use cases. This version can support larger networks while maintaining consensus efficiency and energy efficiency. Ouroboros Omega optimizes security and scalability, making it possible for Cardano to process high-frequency transactions without sacrificing performance.

5th Edition (Ouroboros Chronos): Is the next version of Ouroboros Omega, with some small but important changes and improvements. Ouroboros Chronos focuses on improving scalability and consensus in real-life usage scenarios, while maintaining network security.

4.4.3.1. Some main concepts of Ouroboros

- **Epoch (Era):** Time is divided into epochs, each epoch lasts a certain amount of time, which in the Cardano network is 5 days.
- **Slot (Time slot):** Each epoch is divided into shorter slots. In the Cardano network, each slot is one second, so in one Epoch there will be 432,000 slots.
- **Slot leader (Time slot leader):** In each slot, one network node i selected randomly with probability $p_i = \frac{s_i}{\sum_{j=1}^{n} s_j}$ to become slot leader. Slot leader has the responsibility (right) to create a new block. In there n is the number of nodes in the network, s_i The ADA number that the network node belongs to i stake.

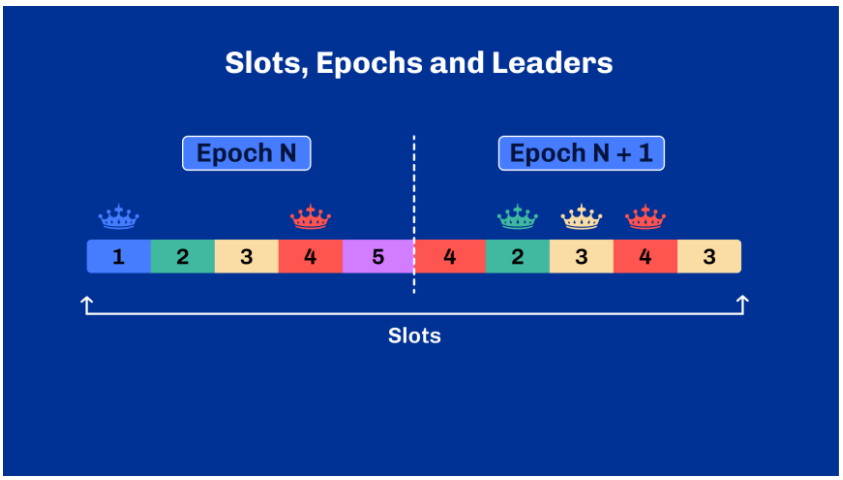

Figure 4.3. Illustrate the concepts of epoch, slot, Slot leader

- **Stake (Share):** ADA holders can "stake" their tokens to participate in the consensus process. The likelihood of a node being selected as a slot leader is proportional to the amount of ADA it has staked or is authorized to stake.

4.4.3.2. How Ouroboros works

Step 1. Stake pool: Users can stake their ADA directly or delegate it to stake pools. Stake pools are nodes operated by pool operators, who are responsible for maintaining node operations and receiving rewards on behalf of delegators.

Step 2. Select slot leader: At the beginning of each epoch, a random selection process takes place to determine the slot leader for each slot in that epoch. The probability of a node being selected depends on the total ADA staked for that node or the stake pool it operates.

Step 3. Create volume: The selected slot leader has the right to create a new block in his slot. This block contains verified transactions. A Cardano transaction includes the following components:

Inputs: Includes existing UTXOs that the sender wants to spend. Each input specifies a specific UTXO from previous transactions.

Outputs: Includes new UTXOs created by the transaction. Each output specifies a receive address and an ADA value to be passed to that address. Additionally, it may contain data script.

Transaction Fee: A small fee is paid to the network to process the transaction. This fee is calculated based on the size of the transaction and the congestion of the network.

Signature (Witness): The sender's digital signature is used to verify the validity of the transaction and ensure that only the owner of the corresponding private key can spend the input UTXOs.

Step 4. Confirm block and forward movement: The slot leader broadcasts the newly created block to the nodes in the network. Nodes in the network verify the validity of the block created by the slot leader.

A valid transaction in Cardano is one that meets the following conditions:

Validity of input: All input UTXOs must exist and have not been spent. The sender must prove his ownership of these UTXOs by providing a valid digital signature.

Validity of output: The total value of the outputs plus transaction fees must be less than or equal to the total value of the inputs. This ensures that no new ADA is created from the air.

Check Script (if any): If any output UTXO contains script data, that script must be executed successfully. This ensures that the conditions specified in the script are met.

Valid signature: All signatures provided must be valid and correspond to the sender's private keys.

Protocol Compliance Check: The node checks whether the block complies with the rules of the Ouroboros protocol, for example: maximum number of transactions in a block, maximum block size, block metadata information.

Consensus: If the block passes all the above checks, the node considers it valid and adds it to its local copy of the blockchain. This process is repeated by all nodes in the network, creating a consensus on the state of the blockchain.

Once validated by a majority of nodes, the new block is officially added to the blockchain, extending the chain and permanently recording the transactions in that block.

Step 5. Get rewards: Slot leaders and stake pools receive ADA rewards for block creation and confirmation. This reward is shared with those who have authorized staking to the pool.

4.4.4. Advantages and disadvantages of PoS

4.4.4.1. Advantages of PoS

- Energy saving: One of the biggest advantages of PoS is that it consumes less energy than PoW. PoW requires miners to solve complex computational problems that consume a lot of energy. In contrast, PoS does not require complex calculations but instead selects block creators (validators) based on the stake they hold. This makes PoS a sustainable and environmentally friendly option.

- High scalability: PoS can process transactions faster and more efficiently as the number of participants increases, as it does not require complex calculations. This improves the scalability of the blockchain, making PoS a good choice for blockchain networks with large transaction volumes.

- Low cost: Not needing powerful hardware like in PoW helps reduce network participation costs for validators. This makes PoS more accessible to many participants and can reduce concentration on a small group of miners or investors.

- Safety and security: PoS has a strong security mechanism, as validators need to stake their assets to participate in block generation and transaction verification. If validators act dishonestly, they will lose part or all of their staked assets. This helps minimize attacks on the network, as PoS network attacks are expensive and difficult to execute.

- 51% attack resistance: In PoS, to dominate the network and perform a 51% attack (i.e. control more than 50% of the consensus power), the attacker needs to own more than

50% of the total shares of the network. This is extremely difficult and expensive to do, especially in large PoS networks.

4.4.4.2. Disadvantages of PoS

- Centralization: One of the biggest problems with PoS is the concentration of people who own a lot of shares. People who own large amounts of tokens have more opportunities to become validators and create blocks, leading to the network becoming concentrated in a few individuals or organizations with large assets. This could reduce the decentralization of the blockchain network, a key goal of systems like Bitcoin.

- The rich get richer: In PoS, those who own more tokens have a higher chance to earn rewards from verifying transactions and creating blocks. This could lead to an increase in the wealth of large investors and reduce opportunities for retail participants. This is a potential problem in maintaining fairness in the blockchain network.

- Randomness and risk of choosing Validator: Although PoS has a random mechanism for selecting validators, stake-based selection of block generators can result in some participants not being selected for a long time, which can reduce motivation to participate. Furthermore, some validator selection methods can lead to unfair or biased decisions.

- Need a large amount of shares to participate: To have a chance to become a validator and create blocks, participants need to have a significant amount of stake. This can create a barrier for new entrants or those with smaller assets. Some PoS networks require participants to stake a large amount of tokens to participate in the consensus process, which can cause the network to have a limited number of participants.

- Fragmentation and attack risks: PoS can face problems with network fragmentation, especially if there is not enough equity in the stakes. If there are too few validators, or if too many stakes are concentrated in a small number of people, it can cause problems with consensus and network efficiency. Furthermore, PoS is also subject to Sybil attacks, where an attacker creates multiple fake nodes to increase his stake and possibly alter the consensus process.

4.5. VARIATIONS OF PROOF OF Stake Algorithms

4.5.1. Delegated Proof of Stake (DPoS) Algorithm

- Delegated Proof-of-Stake (DPoS) is a variation of PoS in which users delegate their validation authority to a group of representatives (Delegates). Representatives are selected through a voting mechanism, with voting rights proportional to the amount of coins held by the user. Some networks use this algorithm such as EOS, TRON, Tezos, ...

4.5.1.1. Mechanism of action of DPoS

Vote: Token holders use their shares to vote for delegates they trust. The number of votes each person has is proportional to the number of tokens they hold.

Select Delegates: A certain number of delegates are elected based on the number of votes they receive. This amount is determined by the blockchain's protocol.

Validation and block generation: Elected delegates take turns validating transactions and creating new blocks. This order is usually determined by a predetermined schedule or a randomization algorithm.

Award: Delegates receive rewards for creating new blocks. This reward can be shared with those who voted for them, depending on how the protocol is set up.

4.5.1.2. Advantages of DPoS

Scalability: DPoS is capable of processing a larger number of transactions than traditional PoS due to the limited number of validators.

Fast transaction speed: Block generation times are typically faster than PoS, resulting in faster transaction speeds.

Energy efficiency: Similar to PoS, DPoS is more energy efficient than Proof of Work (PoW) like Bitcoin.

4.5.1.3. Disadvantages of DPoS

Centralization: Due to the limited number of delegates, DPoS can lead to the concentration of power in a small group of people, reducing the decentralization of the blockchain.

Possibility of manipulation: If a group of people holding a large amount of tokens can control the voting process and elect representatives close to them, they can manipulate the network.

4.5.2. Proof of Weight Algorithm (PoWeight)

Proof of Weight (PoWeight) is a consensus mechanism used in blockchain technology to secure the network and authenticate transactions. Unlike traditional PoW, which relies on solving complex cryptographic puzzles, PoWeight bases its consensus on the amount of "weight" or stake that participants hold in the network. This weight is often correlated to the number of tokens or assets controlled by participants, affecting their ability to create new blocks and validate transactions.

PoWeight was launched in 2017 as a consensus algorithm on the Filecoin blockchain platform and is a major upgrade of the PoS mechanism that aims to eliminate the biased nature of PoS. PoWeight is not a single consensus algorithm. Instead, it is a generic term for a whole series of consensus algorithms largely based on the Algorand consensus model developed by researchers at the MIT Computer Science and Artificial Intelligence Laboratory. , where Algorand is a very fast transaction confirmation protocol.

4.5.2.1. PoWeight's mechanism of action

Determine weight: Participants are assigned weights based on the amount of tokens or assets they hold and are willing to lock up as collateral. The more weight a participant has, the more influence they have in the network.

Block creation and validation: Participants with significant weight can propose new blocks to be added to the blockchain. Their chance of being selected to create a block is proportional to their weight. Other network participants validate proposed blocks. Since this process is based on weights and not computational work, it focuses on verifying the legitimacy of participants' stakes.

Achieving consensus: Once a block is proposed and validated, it is added to the blockchain if it meets the network's consensus rules. The system reaches consensus based on the weights of the participants in the block creation and validation process.

Rewards and incentives: Participants who successfully create and validate blocks will be rewarded with tokens or fees. Rewards are often proportional to the weight they hold, encouraging them to act honestly and participate actively.

4.5.2.2. Advantages of PoWeight

Scalability: The main advantage of the weighted proof consensus mechanism is that it is highly customizable and scalable to many users. The Weighted Proof system allows for the creation of committees consisting of random network users assigned 'weights' according to the consensus protocol.

Security: PoWeight also provides some degree of centralization that helps maintain a fully decentralized and secure network.

Branching risks: The weighted proof mechanism attempts to achieve consensus without any risk of new ramifications because it considers a relative weighted value based on any weighting factor and not just on the amount of money the node holds.

Reduce energy consumption: Unlike PoW, which requires significant computing power and energy, PoWeight relies on staking instead of extensive calculations, resulting in lower energy consumption.

Lower operating costs: With reduced energy and computational needs, it is often more cost-effective to operate and maintain a blockchain network under PoWeight.

Share-based rewards: Participants are rewarded based on the number of weights or shares they hold. This financial incentive encourages active and honest participation, as individuals have a direct economic interest in maintaining the integrity of the network.

4.5.2.2. Some disadvantages of PoWeight

Setup and management: Implementing PoWeight requires careful design and management of stake distribution and weight calculations. This can be complex and may require sophisticated systems to handle stakes and weights correctly.

Token Distribution: The initial distribution of tokens or assets may affect the fairness of the consensus mechanism. If the initial distribution is uneven, it can lead to inequities in network control and influence.

Risks of Staking: The security of the network depends on the economic value of the stakes held by participants. In extreme cases, if a significant portion of the network's stake is compromised or stolen, this could compromise the security of the network.

Legal challenges: The regulatory landscape for PoWeight and similar mechanisms is still evolving. Legal uncertainty and compliance issues can pose challenges to adoption and operations.

Network download: Although PoWeight improves scalability compared to PoW, it can still encounter scalability issues when transaction volume or network scale is extremely large, depending on the implementation.

4.5.3 Some other algorithms.

4.5.3.1. PoH algorithm

Proof of History (PoH - Proof of History) is a consensus mechanism combined with other algorithms (usually PoS) to speed up transaction confirmation in a blockchain network. PoH creates an encrypted time series, allowing nodes to independently verify the order of events (transactions) without synchronizing time with each other, the time and order of recorded transactions again before entering the block. Some networks use this algorithm such as Solana which is a high performance, Pipelined blockchain platform.

PoH is a consensus mechanism developed by Anatoly Yakovenko, founder of Solana Labs. The core idea of PoH is that the order of events in a Blockchain network is as important

as the events themselves, and the ability to prove this order is essential to maintaining the integrity of the network.

To achieve this, PoH uses a Verifiable Delay Function (VDF) to generate a timestamp for each block in the Blockchain. VDF is designed to be difficult to manipulate, thanks to "delay resistance" and "memory resistance," making it difficult for attackers to manipulate timestamps. The timestamp generated by VDF is incorporated into each block on the Blockchain, providing an immutable and verifiable record of transaction order. Thanks to PoH, Solana achieves fast finality, meaning that when a block is added to the Blockchain, it is considered complete and cannot be changed.

PoH is mainly used in the Solana Blockchain network, with a design that can process thousands of transactions per second. PoH helps minimize storage and bandwidth requirements to maintain the Blockchain network, while improving system efficiency and speed, while ensuring security and transaction verification.

How PoH works

1. Cryptographic Timestamping

PoH uses a hash function that is sequential and pre-image resistant. This hash function takes an input (including the current state of the Blockchain and a random seed) and produces a unique, irreversible output, called a hash. This hash is a verifiable timestamp.

2. Generating a Hash Chain

Solana creates a hash chain by applying a hash function sequentially to the output of the previous hash. Each step in the chain represents a "tick" (time beat), with the number of hashes calculated representing the amount of time that has passed. The result is a continuous, verifiable time record that can be used to organize transactions.

3. Record transactions

When a transaction is made, it is tied to the closest hash in the PoH chain. Validators check the validity and timing of the transaction by ensuring that it references a hash in the current PoH chain.

4. Consensus

Transactions timestamped by PoH are then processed through a Proof of Stake (PoS) based consensus mechanism, specifically Tower BFT in the Solana network. Validators stake SOL tokens to participate, receive rewards for securing the network, and confirming transactions. Powered by PoH's timekeeping mechanism, Tower BFT achieves rapid consensus, allowing Solana to process thousands of transactions per second.

5. Verifiable delay function (VDF)

VDF ensures that block producers must go through it to gain the right to create blocks. Solana combines hashes of data from previously created states in the transaction chain, creating a verifiable timestamp that cannot be reproduced or altered.

Advantages of PoH

- High scalability: PoH allows Solana to process tens of thousands of transactions per second.
- Low latency: Transaction wait times are significantly reduced thanks to fast verification.

- Highly secure: The continuous hash chain ensures that the order of transactions cannot be manipulated.
- Energy Efficiency: Does not require complex calculations or energy consumption as in PoW.
- Eliminate Centralized Clocks: PoH integrates time directly into the Blockchain, eliminating the need for a centralized clock system.

Disadvantages of PoH

- High hardware requirements: Due to the need to process large volumes of data, nodes must have powerful hardware, which can reduce decentralization.
- Centralization: Hardware performance requirements can lead to centralization among a few large nodes.
- Technical complexity: PoH implementation requires tight integration between hardware and software, increasing operational complexity.

4.5.3.2. Proof of Authority (PoA) Algorithm

Proof of authority **(**Proof of Authority (PoA) is a consensus algorithm based on reputation, providing a feasible and effective solution for blockchain networks, especially private blockchain networks. The term was proposed in 2017 by Gavin Wood, co-founder and former CTO of Ethereum.

The PoA consensus algorithm leverages the value of identity, instead of having validating nodes staking cryptocurrency (as in PoS), they stake their own reputation. Therefore, PoA blockchains are protected by carefully selected validating nodes that are considered trusted entities.

The PoA consensus algorithm is very flexible and is considered a valuable option for applications in the logistics sector. For supply chains, for example, PoA is seen as an effective and affordable solution. The PoA model allows businesses to maintain their privacy while still taking advantage of the benefits of blockchain technology.

Microsoft Azure is another example of a company applying PoA. The Azure platform provides solutions for private networks that do not require an internal currency, such as “gas” in ether, because mining is not necessary.

PoA is a highly scalable system because it relies on a small number of validators. The system is run by pre-approved participants who are responsible for verifying blocks and transactions. Some networks use this algorithm such as Binance Smart Chain, VeChain, Microsoft Azure Blockchain, ...

How PoA works

PoA is a consensus mechanism that relies on the validation of blockchain transactions by authorized entities. Compared to PoS, PoA is designed to provide a more scalable and efficient solution for building private blockchain networks.

Identity-based authentication: Unlike PoW and PoS that depend on computing power or the amount of stake, PoA uses the identity of validators as the main validation condition. This is an effective consensus mechanism that emphasizes trust and reputation recognition.

Select Validator: Validator or "authority" must have a clear identity and be verified by the entire network. Upon receiving a new transaction proposal, all validators will independently verify the transaction based on the network's rules. If the majority agrees, the transaction will be added to the new block.

Block validation: A consensus algorithm such as weighted randomization or sequential round is used to select validators to create blocks. Once the block is created, it will be broadcast to all nodes in the network. Nodes will check the block's validity, including transactions and links to previous blocks. Once the majority agrees, the block will be added to the blockchain.

Advantages of PoA

- Efficiency: Does not require energy-consuming mining like PoW or large capital deposits like PoS.
- Speed: Fast authentication process, low latency due to fixed list of validators.
- Security: Validator reputation and clear identity help protect against Sybil attacks and other malicious activities.
- Governance: Suitable for private or consortium blockchain networks where centralized governance is necessary.
- Scalability: Easy to expand due to not being limited by resources like PoW or PoS.

Disadvantages of PoA

- Centralization: Validators are pre-selected entities, leading to concerns about the possibility of collusion or centralization of power.
- Limitation of decentralization: PoA does not reach a high level of decentralization like PoW or PoS.
- Validator Attack Risk: If a malicious entity controls a majority of validators, the network can be compromised.

4.5.3.3. PoC algorithm

- Proof of Capacity (PoC) is a consensus algorithm in blockchain, based on hard drive storage capacity instead of computing power or staked assets. In PoC, participants use storage to mine and validate transactions. PoC is also known as Proof of Space or Proof of Storage. Some networks use this algorithm such as Burstcoin, Chia Network, Signum, ...

In PoC, miners use their hard drive capacity to solve mathematical problems and create new blocks. The larger the storage capacity, the higher the miner's chance of creating a new block. Unlike PoW, where miners have to solve complex mathematical problems, PoC uses a simpler hashing algorithm that requires less computing power. This makes PoC more accessible to small-scale miners who cannot afford expensive mining equipment.

Key points of PoC:

1. Storage capacity

In PoC, storage capacity is the most important resource. The more storage space miners have, the higher their chances of generating new blocks. However, this does not mean that

large capacity is required to join the network. Even with a small capacity, miners can still receive rewards.

2. The "plotting" process

Plotting is the process of pre-calculating hash functions to increase mining speed. Miners use their storage space to create plot files containing all the hashes needed for mining. These files are then used to mine new blocks. The plotting process can take a lot of time and resources, but once completed, mining becomes faster and more efficient.

3. Mining

Mining in PoC involves reading the plot file and looking for the correct answer to the mathematical problem. When the correct answer is found, miners can create a new block and receive a reward. Mining in PoC consumes less energy than PoW, making it a more sustainable solution.

4. Rewards

Miners in PoC receive rewards for creating new blocks. Rewards are usually in the form of cryptocurrency. The amount of reward depends on the size of the block and the current market value of that cryptocurrency.

Advantages of PoC:

- Sustainable and environmentally friendly: PoC consumes less energy, contributing to reducing negative impacts on the environment.
- Ease of access: Does not require expensive mining equipment, allowing more people to participate.
- Supports small miners: Even those with limited storage can participate and receive rewards.

PoC is a sustainable and more accessible alternative to traditional consensus algorithms. By using storage capacity as the main resource, PoC contributes to building an energy-efficient and environmentally friendly blockchain network. As the blockchain industry continues to grow, we can expect more innovative consensus algorithms such as PoC to emerge.

4.6. Questions and exercises

1 What is the blockchain consensus algorithm, and what role does it play in maintaining the security and stability of the distributed system?
2 Outline the main steps in the working mechanism of a blockchain consensus algorithm.
3 Why is decentralization an important requirement for a consensus algorithm?
4 Describe the Byzantine generals problem and its relevance to Byzantine fault-tolerant systems (BFT).
5 Compare the advantages and disadvantages of Proof of Work (PoW) and Proof of Stake (PoS) algorithms.
6 Explain how the Proof of History (PoH) algorithm works in the Solana network and its role in increasing transaction processing speed.
7 How does the Proof of Authority (PoA) algorithm work, and why is it suitable for private blockchain networks?
8 Describe the Slot Leader selection mechanism in Cardano's Ouroboros algorithm.

9 How does the Proof of Capacity (PoC) algorithm help reduce energy consumption, and how is it different from PoW?
10 How does Delegated Proof of Stake (DPoS) solve the centralization problem compared to PoS?
11 Let's explain how the Byzantine fault tolerance (BFT) system ensures the stability of the blockchain network even when some nodes are dishonest.
12 One miner using the PoC algorithm has 5 TB of storage, while another has 10 TB. Calculate each person's relative probability of creating a new block.
13 Create a table comparing the advantages and disadvantages of PoW, PoS and PoC. Propose suitable application situations for each algorithm.
14 Simulate how to create and validate blocks using the PoW algorithm. Describes the steps from candidate block generation to block validation.
15 If you had to build a blockchain for a financial application with high requirements for speed and security, which algorithm would you choose among PoW, PoS, PoH or PoA? Explain why.

References

[1] Andreas M. Antonopoulos, “Mastering Bitcoin”, Second edition, Oreilly (2017).

[2] Andreas M. Antonopoulos, Dr. Gavin Wood, “MasteringEthereum, Building Smart Contracts and DApps”, Oreilly (2019).

[3] The Byzantine Generals Problem, Leslie Lamport, Robert Shostak, and Marshall Pease, SRI International (1982).

[4] Markus Jakobsson, Proofs of work and bread pudding protocols (extended abstract), Information Sciences Research Center, Bell Labs, Murray Hili, New Jersey 07974, www.bell-Iabs.com/user (1999).

[5] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld. Proof of activity: Extending bitcoin’s proof of work via proof of stake [extended abstract]y. SIGMETRICS Performance Evaluation Review, 42(3):34–37, 2014

[6] Aggelos Kiayias, Alexander Russell, Bernardo David, Roman Oliynykov, Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol, July 20, 2019

[7] https://www.lcx.com/proof-of-authority-explained/

[8] https://academy.cardanofoundation.org/cbca

[9] https://ocw.mit.edu/courses/15-s12-blockchain-and-money-fall-018/pages/lecture-slides/

[10] https://iohk.io/en/research/library/papers/ouroboros-a-provably-secure-proof-of-stake-blockchain-protocol/