Consensus Algorithms

Securing Blockchain Transactions

CoinBundle Team Follow
Sep 21, 2018 · 9 min read

You may already know that each and every transaction that you make on a blockchain network is completely verified and secured, but have you ever wondered why, or more importantly, how? Well, this is where blockchain consensus algorithms come into play, as there are several different ways in which various blockchain networks can both verify and secure a block of transactions on its network. Let's go over the different kinds of consensus algorithms which exist today and how they differ from one another.

This is not financial investment advice.
This article will describe various consensus algorithms that exist today.

In this article

Terminology

Consensus Algorithm: A consensus algorithm is a process in computer science used to achieve agreement on a single data value among distributed systems. Consensus algorithms are designed to achieve reliability in a network involving multiple nodes. Solving the issue — known as the consensus problem — is important in distributed computing and multi-agent systems.

Blockchain: The easiest way to understand blockchain is to think of it as a fully transparent and continuously updated record of the exchange of information through a network of personal computers, a system which nobody fully owns. This makes it decentralized and extremely difficult for anyone to single-handedly hack or corrupt the system, pretty much guaranteeing full validity and trust in each exchange of information.

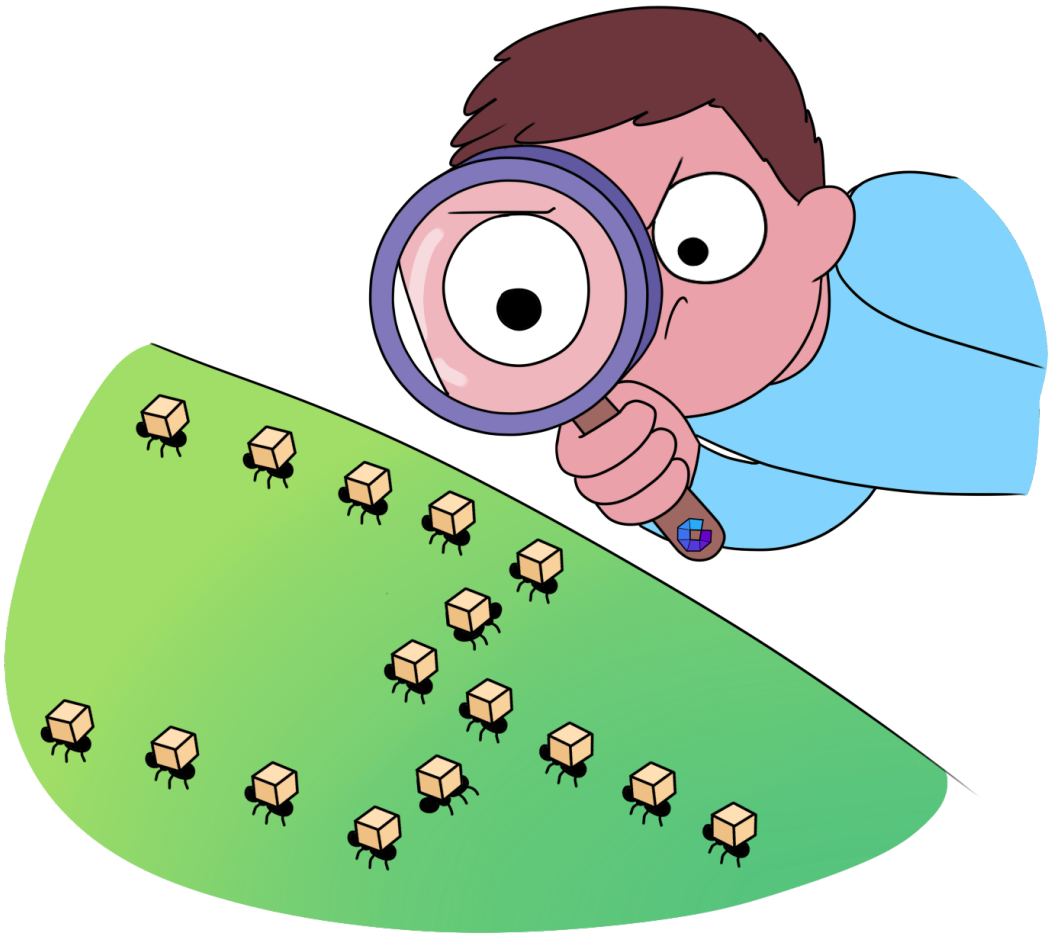
Node: A node is a point of intersection/connection within a network. In an environment where all devices are accessible through the network, these devices are all considered nodes.

Mining: Crypto mining is the process by which transactions are verified and added to the public ledger, also known as the blockchain, and also the means through which new Bitcoin are released.

Hash Algorithms (Functions): A cryptographic hash function is a mathematical function used in cryptography. Typical hash functions take inputs of varying length to return outputs of fixed length. A cryptographic hash function combines the message-passing capabilities of hash functions with security properties.

> *Familiarize yourself with the aforementioned terms to get a better understanding of how consensus algorithms work.*

What Are Consensus Algorithms?



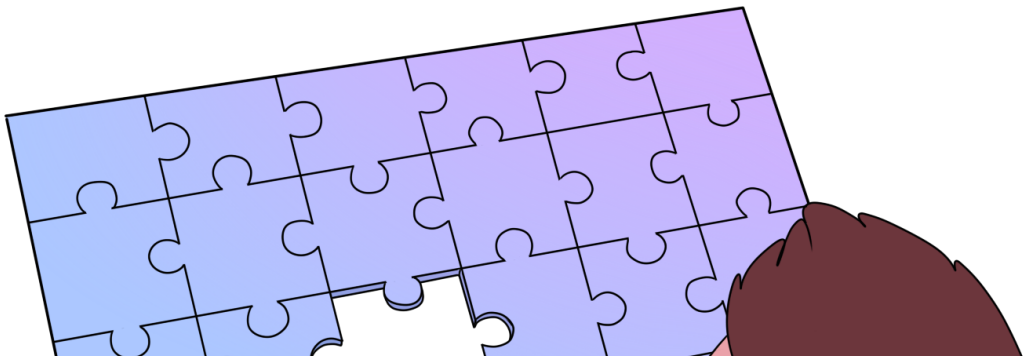
When it all comes down to it, consensus algorithms are what make the blockchain network so secure and decentralized. As we mentioned above, **consensus algorithms** are designed to achieve reliability in a network involving multiple nodes. Solving that issue — known as the consensus problem — is important in distributed computing and multi-agent systems.

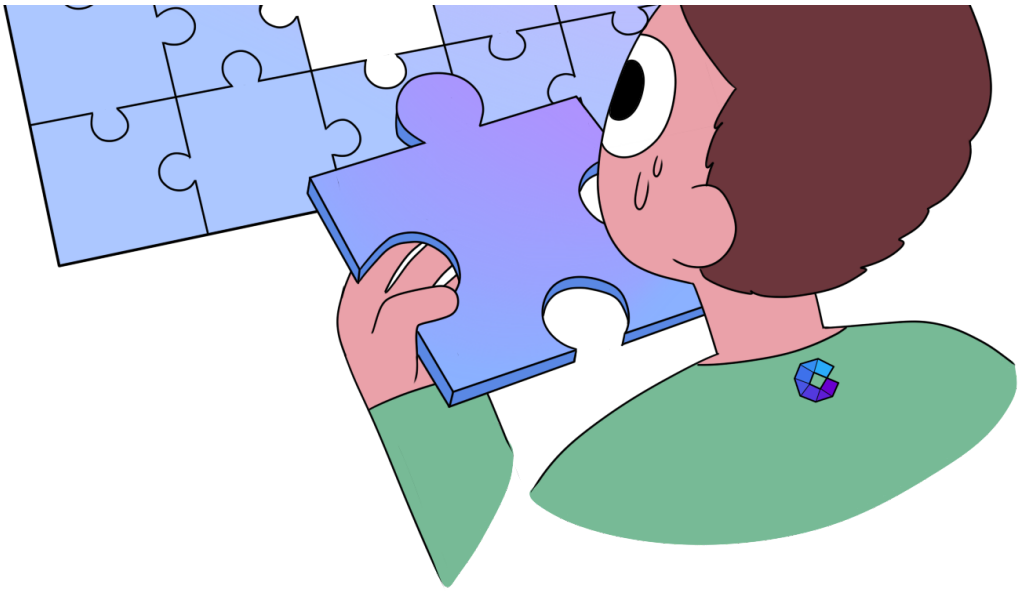
Essentially, consensus algorithms are capable of doing two things: ensuring that the next block in a blockchain is the one and only version of the truth, and keeping powerful adversaries from derailing the system and successfully forking the chain. However, there are also some problems with consensus algorithms that many people have pointed to.

Some of these common criticisms include that it requires enormous amounts of computational energy, that it does not scale well (transaction confirmation takes about 10–60 minutes) and that the majority of mining is centralized in areas of the world where electricity is cheap. Yet, despite all of these problems, consensus algorithms are absolutely necessary for blockchain technology to fully execute its capabilities. With that being said, you should be aware of the fact that there are several different kinds of consensus algorithms which exist today, so let's go over them and discuss how they are related to each other.

> ***Consensus algorithms*** *ensure that the next block in a blockchain is fully validated and secured. There are multiple kinds of consensus algorithms which currently exist, each with different fundamental processes.*

Proof of Work





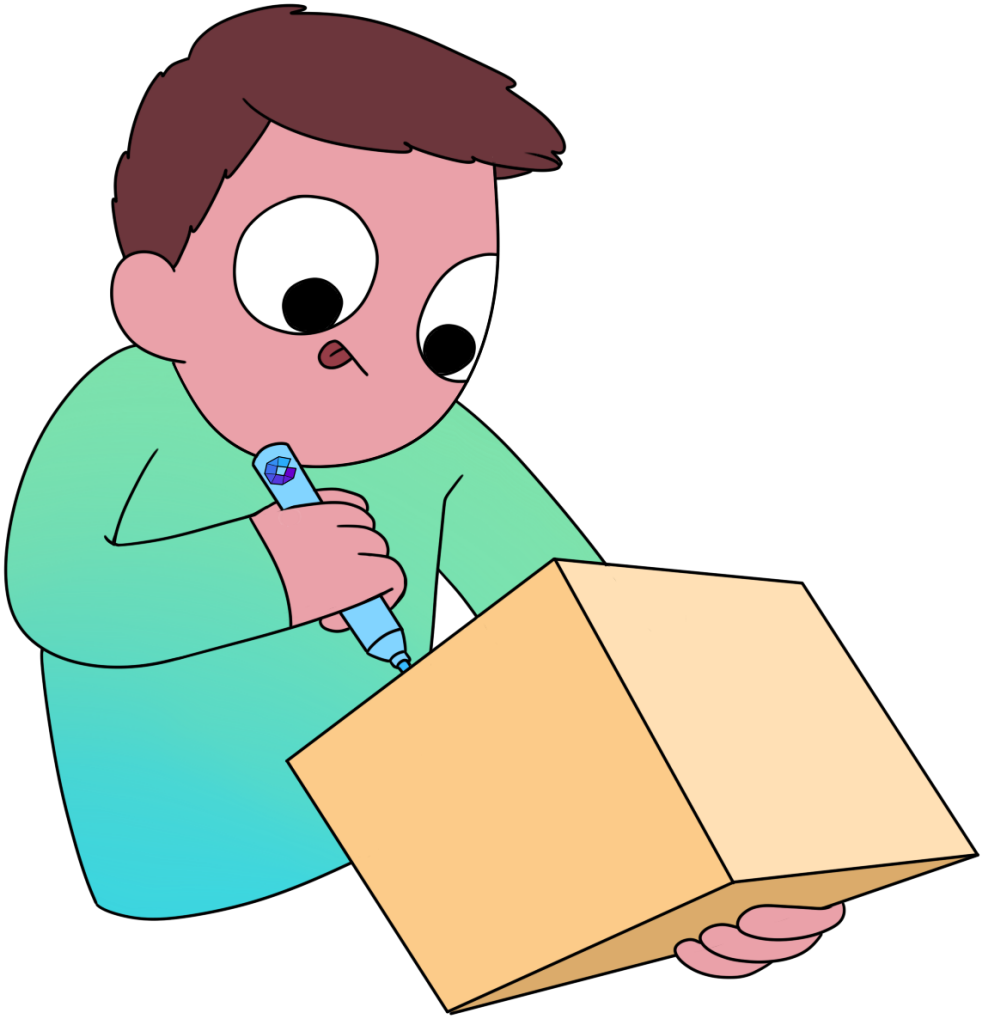
The first kind of consensus algorithm that we'll be discussing is the Proof of Work (PoW) algorithm. Proof of work in Bitcoin (and other proof of work cryptocurrencies) function as a tool that is used to process blocks of transactions and add them to the blockchain. In other words, proof of work is utilized for block generation.

The process of generating correct proofs in order to add a block to the blockchain is known as **"mining"** and the individuals that participate in the mining process are known as **"miners."** The very first implementation of a distributed and trustless consensus algorithm is Bitcoin's proof-of-work (PoW) algorithm. PoW requires miners to solve complex cryptographic puzzles before they can add a block to the blockchain.

In exchange for solving the puzzle, miners are rewarded with Bitcoin and is otherwise known as a **block reward**. It is important to note that each block that is added to the blockchain must follow a certain set of consensus rules. In order to get to the block in the first place, miners must compete with other miners to find a correct hash for each hash function. There is a network difficulty which dynamically adjusts itself for how hard it is for miners to find the correct hash values. As soon as a miner stumbles upon the correct hash value, all other nodes in the system verify that it is correct before executing the transaction of each new block.

> ***The Proof of Work (PoW) consensus algorithm*** *functions as a tool that is used to process blocks of transactions and add them to the blockchain. Nodes within the network verify each block before it's added to the blockchain.*

Proof of Stake



Next, we have the **Proof of Stake (PoS)** consensus algorithm. This is a relatively different and new way to generate blocks within a blockchain, differing from that of the Proof of Work algorithm. With proof of work, miners who find the correct hash are allowed to generate new blocks and are rewarded for doing so. However, with proof of stake systems, individuals that are chosen to generate a block, also known as **validators**, depend on a different set of criteria.

The specific criteria differs depending on the proof of stake system, but in most systems, a validator is chosen to generate a new block based on their economic stake in the network. So for example, validators are selected to generate a new block with a probability that is proportional to the amount of coins that the validator possesses. Thus,

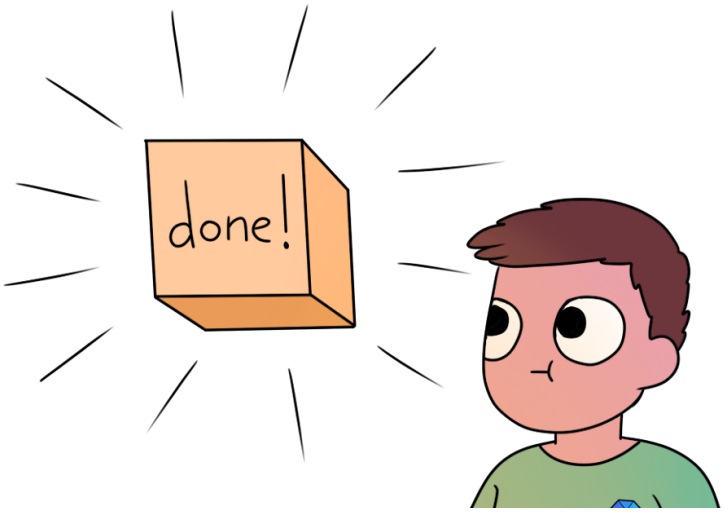
the more coins a validator houses in his wallet, the increased likelihood of being selected to generate a block.

Some proof of stake systems also take into account the length of time that a validator has held coins in their wallet. This criteria is usually referred to as “coin age.” **Coin age** is defined as the coin amount multiplied by the number of days that the coins have been held in a wallet. Therefore, a validator possessing a large holding of coins over a lengthy time-period is more likely to be selected to generate a new block. Proof of stake is seen as being a superior block generating mechanism to proof of work because of reasons primarily pertaining to energy consumption.

Operating proof of work systems such as Bitcoin requires a tremendous amount of energy. In fact, it’s estimated that roughly 6.5 million U.S. households could be powered by the energy that is consumed from operating Bitcoin. Instead of consuming electricity to produce countless hashes for the right to generate a block, as is required in proof of work systems, validators in proof of stake systems are selected for block generation based on their economic stake in the network, which is a system that requires considerably less computing resources to operate. As such, it’s seen as a much more reliable and energy-efficient system to use.

> ***The Proof of Stake (PoS) algorithm*** *utilizes validators who are chosen based on criteria that includes coin age and economic stake. PoS also requires significantly less energy thanks to its validator selection criteria.*

Proof of Burn



Another consensus algorithm which is used by numerous blockchain networks is the **Proof of Burn (PoB)** algorithm which works in a fairly simple fashion. The miners of the PoB coins will send coins to an unspendable address — otherwise known as an “eater address” — thus taking them forever out of circulation or burning them. These transactions are recorded on the blockchain, ensuring that there’s a necessary proof that the coins cannot be spent again, and the user who burned the coins is issued a reward.

The entire idea behind proof-of-burn consensus is that the user burning the cryptocurrency is showing long-term commitment to the coin by burning it. This is because they are taking a short-term loss in exchange for a long-term gain. Additionally, burning coins is also viewed as less resource intensive by some since the main resource being used is the person’s willingness to delay their profits. As long as someone is okay with being patient for their gains, then this may be the consensus algorithm from him or her.

As time progresses, the user of a proof-of-burn coin continues receiving rewards, either increasing their stake of alternative coins or earning greater privileges for mining on the network. This way, if a user burns more coins, they’ll have a greater chance of successfully mining the next block and further increasing their overall rewards. Moreover, there are actually a few different ways to implement a proof-of-burn consensus mechanism. In some cases, an existing proof-of-work coin can be burned in

exchange for the PoB coin. In other cases, the actual PoB coin is burned in order to gain increased mining privileges.

In much the same way that the cost of mining Bitcoin increases over time in the form of hardware and electricity costs, the cost of mining a PoB coin also increases over time as more coins need to be burned to maintain the same odds of being selected to mine the next block.

> ***The Proof of Burn (PoB) algorithm*** *utilizes miners who send their coins to an unsendable address, effectively "burning" the coin forever. The entire idea behind proof-of-burn consensus is that the user burning the cryptocurrency shows long-term commitment to the coin by burning it, while receiving his or her gains much later.*

Byzantine Fault Tolerance

The last consensus algorithm which we'll be discussing is the Byzantine Fault Tolerance (BFT) algorithm. The first thing you may notice when looking at this is that it doesn't have the words "proof of" in its name. Well, this is because its derived from the Byzantine Generals' Problem, a dilemma that has been extensively researched and optimized with a diverse set of solutions in practice and actively being developed. So then, how does BFT work after all?

Well, **Byzantine Fault Tolerance** is the ability of a distributed computer network to function as desired and correctly reach a sufficient consensus despite malicious components (nodes) of the system failing or propagating incorrect information to other peers. The main objective is to defend against any catastrophic system failures by mitigating the influence that these malicious nodes may have on the correct function of the network. As a result, the right and correct consensus is reached by the honest nodes which are still in the system.

Circling back to the Byzantine Generals' Problem, which is what set this system in motion, there have been many specifications and improvements made to the overall algorithm since the overarching problem has been studied and analyzed for years now. **Practical Byzantine Fault Tolerance (pBFT)** is one of these optimizations and was introduced by Miguel Castro and Barbara Liskov in an academic paper in 1999 titled "Practical Byzantine Fault Tolerance." It aims to improve upon original BFT consensus mechanisms and has been implemented and enhanced in several modern distributed computer systems, including some popular blockchain platforms.

> ***The Byzantine Fault Tolerance (BFT) algorithm*** *seeks to correctly reach a sufficient consensus despite malicious components (nodes) of the system failing or propagating incorrect information to other peers. The main objective is to defend against any catastrophic system failures by mitigating the influence that these malicious nodes have.*

Conclusion

So now that you've seen several different kinds of consensus algorithms which exist today across the blockchain industry, it's important to be aware of why they exist in the

first place. Without consensus algorithms, many (if not all) blockchain networks would not be able to function properly and execute its full capabilities while still remaining decentralized and completely verified. Whether it's PoS, PoW, or PoB (or even BFT), there are now many ways to handle the verification process when it comes to securing each new block on the blockchain. As always, happy investing!

. . .

What's your favorite consensus algorithm?
Let us know why in the comments!

Blockchain Algorithms Cryptocurrency Beginnerscoinbundle Security