Proof of History, Proof of Stake, Proof of Work - Explained

What's this Article About?

Blockchains are distributed ledgers that record transactions across a network of computers. Consensus algorithms are crucial to blockchains as they are used to reach agreements on the ledger's state. They facilitate collaboration among mutually distrusting participants, eliminating the need for a centralized entity to validate data before it is added to the blockchain. Without consensus algorithms, there would be no way to ensure that all nodes agree on the blockchain's state - we'd be vulnerable to numerous attack vectors, run into issues of double-spending, compromise immutability, and wouldn't have a nice way to resolve disagreements or forks.

In this article, we will delve into consensus algorithms, looking at their importance, and the different types used in popular blockchains. The goal of this article is to give you a comprehensive understanding of what consensus algorithms are, why it is important to understand how they work, and go over how a few popular consensus algorithms work.

Why You Should Learn About Consensus Algorithms

It is important to understand consensus algorithms to effectively build atop of your blockchain of choice for the following reasons:

- Knowing how your blockchain reaches consensus affects architectural choices in the development of your decentralized application (dApp). You'll be asking questions like what are the costs of deployment? How many transactions should a user be sending in order to effectively use this dApp? How much does sending a transaction cost? The most important question you'll be asking is: what is its time to finality? Or, how long does it take for a transaction to be confirmed and added to the blockchain?
- Knowing the throughput and latency characteristics of a consensus algorithm can help you choose the right blockchain to deploy your dApp. Asking questions about your dApp's architecture will make you aware of areas you can optimize for performance
- Knowing security concerns of each algorithm will help you design more secure applications. What are the attack vectors for your blockchain? What are the design implications for smart contract development?
- Knowing the inner workings of your blockchain's consensus algorithm empowers you to grasp the nuances of its voting mechanisms, allowing you to actively participate in governance
- Knowing the specific economic incentives for each consensus algorithm helps encourage network participation. How can I participate in the network and earn rewards? What behavior can I avoid so I am not penalized?
- Knowing the fundamentals of consensus algorithms will help you better understand new or updated algorithms. How can you understand Delegated Proof of Stake or Leased Proof of Stake if you don't understand Proof of Stake?

What's a Consensus Algorithm?

One of the key challenges in distributed computing is achieving reliable systems performance even when some of its components fail. The issue is described in the Byzantine Generals' Problem. The problem describes a thought experiment where all of the system's participants must agree on a strategy to avoid failure

. It highlights the difficulties of reaching agreement in a network where some participants may act unpredictably or maliciously. To mitigate this, resilient coordination processes are needed to establish a single source of truth. This ensures that all participants act reliably across the network. We refer to the processes that help the system agree on one source of truth as consensus algorithms

.

Imagine a busy city intersection with no traffic lights. It would be absolute chaos with cars, trucks, bikes, and pedestrians all trying to go their own way, vying for their turn to cross. There would be accidents, misunderstandings, and distrust between participants. Thankfully, we have traffic lights. Traffic lights bring order; telling who to go and who to stop, adapting to real-time conditions. Most importantly, everyone agrees on traffic lights

. We all agree to the rules, and that the rules are applied uniformly.

Consensus algorithms are the traffic lights of blockchains. They set the rules for how transactions are added to a blockchain. They create a safe and efficient flow of data across the network by giving out "green lights" and "red lights" for valid and invalid transactions and blocks. These rules are applied uniformly in a secure, transparent manner. Consensus algorithms adapt to changing network conditions in order to maintain optimal performance all while operating within the

confines of these rules.

Consensus algorithms are crucial for blockchains. Without them, we wouldn't have a uniformed way to validate data in an adversarial environment. Instead, we'd have pandemonium. So much so that we would most likely revert back to relying on a central authority for validation because we'd be so tired of Sybil attacks and double-spending. We need consensus algorithms so our blockchains stay secure, immutable, and decentralized.

What's Proof of Work?

Proof of Work (PoW) is a form of cryptographic proof where one party (the prover) demonstrates that they have spent a specific amount of computational power to another party (the verifier). The verifier can easily verify this expenditure. It was invented by Moni Naor and Cynthia Dwork in 1993 to deter DoS attacks and spam on a network, and was later formalized in a 1999 paper by Markus Jakobsson and Ari Juels.

Proof of Work was popularized by Bitcoin as a foundation for consensus in a permissionless decentralized network. Satoshi Nakamoto explains in the Bitcoin whitepaper how Proof of Work can be used to create a purely peer-to-peer version of electronic cash without requiring any intermediaries. Other popular blockchains that use a PoW-based consensus algorithm include Litecoin, Kadena, Monero, and Ethereum Classic. So, how does it work?

How It Works

Proof of Work blockchains require network participants to solve a complex mathematical problem, using a significant amount of computational power. The goal is to guess a 64-digit hexadecimal number, known as a hash. Finding out this hash sounds easy, but is not when you consider the hash is the result of hashing all the transaction information contained in a block with a random nonce (”number used once) using the SHA256 algorithm. The first participant to solve the problem gets to add the next block of transactions to the blockchain and is rewarded with a predetermined amount of cryptocurrency. This process of validating transactions and adding them to the blockchain is known as mining, and network participants are referred to as miners.

Benefits and Drawbacks

The nice thing about Proof of Work blockchains is that anyone

can participate in mining, promoting a decentralized distributed network. The computing power required to attack a PoW network is a great deal. This makes it prohibitively difficult, although still theoretically possible, for a single entity to execute a 51% attack. A 51% attack is when a malicious entity controls a majority of the network's hashing power so they can manipulate the transaction history. Proof of Work is a relatively simple to understand consensus algorithm that has been implemented and tested at scale via Bitcoin.

While this sounds advantageous, there are a number of drawbacks with Proof of Work. The high costs associated with mining hardware and electricity can, and has, led to mining centralization in areas with low energy costs. These high barriers to entry for profitable mining means that rewards are distributed unevenly, favoring those who can afford powerful mining rigs. This has led to the creation of massive Bitcoin mining farms. Because of its highly energy-intensive nature, a number of environmental concerns have been raised. This is one of the main reasons Ethereum transitioned to Proof of Stake with its update entitled [The Merge

](https://ethereum.org/en/roadmap/merge/). So, what is Proof of Stake?

What's Proof of Stake?

Proof of Stake (PoS) aims to remedy the computational and energy-intensive concerns associated with Proof of Work. Instead of relying on computational power to secure the network, Proof of Stake selects validators based on the number of tokens they hold as a stake in the network. Peercoin was the first cryptocurrency to use Proof of Stake in 2012, although it was used alongside a Proof of Work system.

How It Works

In Proof of Stake, miners are replaced with validators who propose and vote on blocks. These validators are required to lock up a certain amount of tokens as their stake in the network. The network selects a validator to validate the next block of transactions based on a number of factors such as the size of their stake or the time they've held their stake. The proposed block is then verified and attested by the other validators. If the block is attested as valid, then the block is added to the blockchain. For this, validators earn transaction fees and, sometimes, newly minted tokens for their validation efforts. If the block is attested as invalid, the block is not added to the blockchain and the validator is penalized. These validators are “slashed” meaning that they lose a portion of their stake. These slashing penalties are in place to discourage bad actors from proposing fraudulent blocks or creating discrepancies on the ledger.

Benefits and Drawbacks

The issues of high energy consumption are directly addressed by Proof of Stake. Ethereum's switch from Proof of Work to Proof of Stake resulted in a 99.84% reduction in the network's energy consumption.Proof of Stake algorithms are faster and better suited for scalability since their algorithms are designed for higher throughput. These algorithms are designed for quicker finality, meaning transactions are confirmed and added to the blockchain more quickly. Validators are also financially motivated to maintain exceptional infrastructure for validation, resulting in quicker validation times. Proof of Stake algorithms are also better suited for parallel transaction processing and sharding. Sharding is when the network is broken up into smaller pieces, or "shards", that process transactions independently and in parallel.

Proof of Stake, however, has its own set of drawbacks. While being more energy-efficient, validator rewards could be lower than their Proof of Work counterparts. This could potentially attract fewer participants and decrease the network's security. The initial distribution of the token can also affect fairness and decentralization in the network if it is not properly managed. Here, those with larger stakes have disproportionate influence over the network. Another potential issue is that validators could have nothing to lose for voting for multiple blockchain forks, as opposed to Proof of Work where this would require splitting computational power. Certain slashing conditions need to be put in place to avoid this kind of behavior.

Proof of Stake Variants

Prominent blockchains that use Proof of Stake include:

Ethereum

- Ethereum uses a LMD-GHOST algorithm with Casper-FFG, referred to as Gasper. LMD-GHOST is used to accumulate votes and ensure that nodes easily select the correct fork when one arises. Casper-FFG (Casper the Friendly Finality Gadget) upgrades certain blocks to "finalized" so new entrants into the network always sync to the canonical chain

Cardano

- Cardano uses a variant of Proof of Stake called Ourobros, the first provably secure Proof of Stake protocol. It's based on peer-reviewed research and is designed with scalability and security in mind. You can learn more about it here

Near

- Near uses Thresholded Proof of Stake, a deterministic way to have a large number of participants that maintain network maintenance by making decisions during specific time intervals. You can learn more about it here

Algorand

- Algorand uses Pure Proof of Stake, a more egalitarian approach to Proof of Stake built on Byzantine consensus. You can learn more about it here

As you can see, there are many variants of Proof of Stake. Most Proof of Stake blockchains use some variant of the original design but have modified it to suit their needs and optimize certain use cases. One of the most well known and widely used variants is Delegated Proof of Stake.

What's Delegated Proof of Stake?

Delegated Proof of Stake is an evolution of Proof of Stake, designed to enhance the efficiency and democratic nature of blockchain validation processes. It was developed by Daniel Larimer in 2014 and has been implemented in a number of prominent blockchains since then, namely: BitShares, EOS, TRON, and SUI.

How It Works

In Delegated Proof of Stake token holders vote for a group of delegates to validate and create new blocks on their behalf. Delegates are elected by token holders, where voting power is correlated to the amount of tokens held. Here, users vote by pooling their tokens into a staking pool and link them to a particular delegate. Delegates are incentivized to act honestly since they can be voted out due to malicious activity or failing to maintain sufficient uptime. When delegates validate a block, they receive the corresponding transaction fees as a reward. Delegates then distribute these rewards to users who supported them based on each user's stake. It is important to note that these delegates validate blocks deterministically, according to a public schedule. There is a limit on the number of delegates for each block meaning that delegates are shuffled periodically.

Benefits and Drawbacks

Delegated Proof of Stake carries many of the benefits of Proof of Stake: anyone can become a delegate, its low barrier to entry makes it more accessible and decentralized, it has improved performance since it requires only a limited number of delegates, and it doesn't require much power to run the network.

Delegated Proof of Stake, however, isn't perfect. Delegated Proof of Stake only requires a limited number of delegates for every new block. This raises concerns regarding giving a small group disproportionate influence over transaction verification

and governance decisions. This limit introduces the possibility of these delegates conspiring to act maliciously, greatly lowering the threshold of a 51% attack. Token holders could potentially bribe delegates to act maliciously on their behalf. More importantly, users are not mandated to participate in delegate elections - voter apathy could exacerbate these aforementioned centralization risks.

What's Proof of History?

Proof of History is Not A Consensus Algorithm

Proof of History is not

a consensus algorithm. More accurately, it is a component that aids in achieving consensus. The confusion likely arises due to its terminology - the term "Proof of X" likely implies a consensus algorithm for those familiar with Proof of Work and Proof of Stake. Proof of History is fundamental to Solana's architecture, deeply integrated in transaction ordering and program execution. It is easy to mistake it as Solana's consensus algorithm due to its prominence within the network.

So, why are we talking about it if it isn't a consensus algorithm? Proof of History addresses a fundamental problem in distributed systems - the agreement of time, or the sequencing of events. Solana uses Proof of History as a sort of "pre-consensus" algorithm to streamline consensus so transactions are processed efficiently. Because of this, validators can process transactions in parallel improving throughput and reducing latency. Thus, Proof of History is a component that aids in achieving consensus. It is better to think of Proof of History as a decentralized clock for the network - it provides a way to prove time and the order in which events occurred without having to rely on a third party.

Shortcomings of Traditional Approaches

Traditionally, blockchains synchronize on blocks, which are large chunks of transactions. This means that a transaction cannot be processed until a specific duration has passed. This is known as block time. In Proof of Work, block times need to be large (Bitcoin produces a block roughly every 10 minutes) to reduce the likelihood of multiple validators producing a new block at the same time. In Proof of Stake, there isn't a constraint but validators need timestamps to determine the order of incoming blocks. The popular workaround is to put a wallclock timestamp on each block. This timestamp, however, is only valid if it is greater than the median timestamp of the previous 11 blocks, and is less than the "network-adjusted time" plus two hours. Network-adjusted time refers to the median of timestamps returned by all nodes connected to you. This isn't the greatest solution because of clock drift and network latency. So now what?

Proof of History

Solana takes a radical approach to this issue, known as Proof of History. Simply put, Proof of History is a way to prove time in an adversarial network. Proof of History acts as a cryptographic time-stamping function, allowing for nodes to agree on an order of events without having to talk to one another. This is achieved by using a sequential preimage resistant hash function (the function is hard to invert) to create a chain of hashes where each hash depends on the previous hash. Leader nodes apply timestamps to blocks using these cryptographic proofs to prove that some duration of time has passed since the last proof. Since all the hashes are chained, a historical record that proves data existed at a certain point in time is created.

This unique approach relies on Verifiable Delay Functions (VDFs), functions that take incredibly long to compute but whose output can be verified quickly. Verifiable Delay Functions are used in the creation of hashes that not only rely on the previous hash, but also rely on the time that has elapsed. This allows for the creation of a verifiable timeline of events. The use of Verifiable Delay Functions ensures this, as tampering with one hash would require the recalculation of all previous hashes. This adds an extra layer of security and integrity to Solana since there is only one verifiable timeline of events.

A Simple Analogy

Imagine a medieval town, bustling with trades, announcements, debates, and disputes. This town relies on a town crier as its central source of information, shouting important news and making sure everyone is on the same page. One day, the town crier falls ill and can no longer carry out their duties. The town is engulfed in chaos - no one can agree on what happened when, who said what, and what sequence of events everything occurred.

Enter a very meticulous scribe. This scribe sits in the middle of the town square with a unique ink and quill that is used to record every event to their journal. This ink is special - it changes color based on the last entry in the journal. Everyone can simply refer to the scribe's journal and confirm both the order and the timing of the events without having to go around and ask everyone. This journal becomes the undisputed source of truth for this town, removing the need for a town crier as people can make important announcements without them. This changing ink ensures a permanent, immutable source of truth that validates all previous entries. If you replace the town with a network, the journal with a ledger, and the changing ink with a cryptographic hashing function, you have one of the most reliable methods of synchronization. This is the power of Proof of History.

Advantages and Drawbacks

Proof of History allows for shorter blocktimes, the handling of a large number of transactions per second, and a single, verifiable source of time before consensus. This also allows for resource optimization - nodes can process transactions without waiting for consensus, allowing for better parallelism and efficient use of computational power. The use of Verifiable Delay Functions adds an extra layer of security as altering transactions would require recalculating the sequential hash. This would be very expensive and easy to detect. Moreover, anyone can verify the order and time of transactions because of the cryptographic timestamps

. We know with certainty when a transaction occurred without having to worry about the validity of wallclock timestamps. This promotes transparency and accountability, central tenets to the ethos of crypto.

Mind you, Proof of History isn't perfect. This model adds a great deal of complexity to Solana's network architecture, making it harder to understand and potentially increases the risk of bugs or vulnerabilities. Because of the resource intensive computation needed for the Verifiable Delay Functions, Solana nodes require more powerful hardware. This increases the cost of network participation in the short term. Thanks to Moore's Law, this hardware barrier should

lower over time due to the accessibility and affordability of once expensive powerful hardware. Moore's Law is the observation that the number of transistors on a microchip doubles roughly every two years, leading to an increase in computing power.

Conclusion

Congratulations! In this tutorial, we delved into the intricacies of consensus algorithms, what they are, why you should learn them, and looked at popular implementations. By now, you should have a well-rounded understanding of consensus algorithms. Understanding these algorithms is not just an academic endeavor - it is a practical necessity. This endeavor provides you with a fundamental understanding of the networks you're building on top of, directly influencing your ability to create robust and efficient applications. Here, you are able to contribute to the blockchain communities you are a part of in a meaningful way. Within the rapidly evolving landscape of blockchain technology, this knowledge is indispensable.

Looking to the future, we can expect more innovative consensus algorithms to emerge. We can even expect current models to be updated and tweaked to suit the latest obstacles to scalability, security, and efficiency. Whether you're an investor, developer, or blockchain enthusiast, this is an exciting time to be involved with this kind of proprietary technology. Your understanding of consensus algorithms puts you at the forefront of innovation, enabling you to navigate the space as it continues its upwards trajectory.

If you've read this far anon, thank you!

Additional Resources / Further Reading

- Bitcoin Whitepaper
- Ethereum Whitepaper
- Coinbase's PoW vs PoS Article
- Proof of Stake vs Delegated Proof of Stake
- Solana Documentation on Synchronization
- Video on VDFs by a16z crypto