Phil Glazer Follow
Investing @ MGV, previously @KKR_Co & @UCBerkeley
Mar 14 · 6 min read

An Overview of Cryptocurrency Consensus Algorithms

Source

One of the most important aspects of a decentralized cryptocurrency project is the consensus algorithm it employs. A consensus algorithm is crucial to the implementation of a digital currency because it prevents the double spending problem, a challenge that has historically limited the development of digital currencies until the recent development and adoption of the blockchain ledger method. Because cryptocurrencies are implemented as public, decentralized ledgers that are append-only, they must employ a **consensus algorithm** to verify that there "is one version of the truth" and that the network cannot be overwhelmed by bad actors.

As explained by TechTarget, "A consensus algorithm is a process in computer science used to achieve agreement on a single data value among distributed processes or systems. Consensus algorithms are designed to achieve reliability in a network involving multiple unreliable nodes. Solving that issue—known as the consensus problem—is important in distributed computing and multi-agent systems. To accommodate this reality, consensus algorithms necessarily assume that some processes and systems will be unavailable and that some communications will be lost. As a result, consensus algorithms must be

fault-tolerant. They typically assume, for example, that only a portion of nodes will respond but require a response from that portion, such as 51%, at a minimum."

In the context of cryptocurrencies, consensus algorithms are designed to ensure that transactions are valid and distributed across many participants to verify accuracy and resiliency through redundancy.

Across current projects, there are four leading implementations, each with its own unique set of benefits and trade-offs: Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-of-Stake (DPos), and Byzantine Fault Tolerance (BFT). **Note: this is an evolving space—other approaches exist and new approaches are likely to emerge.** I will also be updating this document as new information emerges about the consensus algorithms profiled and I discover better explanation.

Proof-of-Work (PoW)

Proof-of-Work (PoW) was the first successful decentralized blockchain consensus algorithm. PoW is still used by Bitcoin and other cryptocurrencies, like Ethereum (Ethereum plans to move to Proof-of-Stake, more details in next section), Litecoin, ZCash, Monero, and many more. PoW requires participants to perform work that is computationally intensive but easy to verify by others in the network. In the case of Bitcoin, "miners" compete to add a collection of transactions, known as a block, to the global blockchain maintained by the network. To do this, a miner must be the first to correctly figure out the "nonce", a number appended to the end of a string to create a hash that starts with a required number of zeroes (this is an abstraction of details; this piece provides a more detailed overview).

The most significant positive attribute of PoW is that is has been proven to work over a period of a few years, which is more than can be said for many other consensus algorithms. PoW is not without its shortcomings, however, which include high power consumption for the mining process and low transaction throughput.

Proof-of-Stake (PoS)

There are a variety of proposed implementations of Proof-of-Stake (PoS). In all implementations, PoS requires participants to "stake" a

portion of the coins that they hold in the network to verify transactions. Rather than "mining" by completing computationally difficult problems to verify transactions, "minters" stake their coins on transactions by locking up coins. The minter selected to complete the block is often selected in proportion to the value they have staked in the network compared to the total value of the network, how long coins held have been locked up, or some other measure to ensure that the minter is aligned with the long-term interest of the network. While Proof-of-Work deters bad behavior by making it computationally exhausting and uneconomical, Proof-of-Stake deters bad behavior by shifting verification to those who have the most value bundled up in the network and, therefore, have the greatest interest in seeing it succeed. Minters that stake their coins to chains that posses fraudulent transactions will have their stakes slashed. Like Proof-of-Work (PoW), the details of Proof-of-Stake are more nuanced than presented and this piece provides more information. Proof-of-Stake is currently implemented by Peercoin, Decred, and soon Ethereum, which lists a planned shift to PoS in its development timeline. That advantages of PoS are that it is more energy efficient and possibly better at preventing attacks than PoW, but has not yet been proven effective or implemented in a major project.

Delegated Proof-of-Stake (DPoS)

While Delegated Proof-of-Stake (DPoS) is similar in name to Proof-of-Stake, the implementation details are meaningfully different. In DPoS, instead of staking coins to validate transactions, token holders vote for a select group to serve the role of validating transactions. DPoS remains "decentralized" in the sense that all in the network participate in the selection of which nodes validate transactions, but centralized in the sense that a smaller group makes decisions which increases transaction speed and verification. DPoS implementations maintain a reputation, ongoing voting process, and shuffling system that keeps elected validators accountable and honest. The advantages of DPoS are that it is scalable and provides fast transaction verification, but the disadvantage is that it partially centralized and the governance model has not been proven effective in a large project. DPoS is employed by Steemit, EOS, and BitShares.

Byzantine Fault Tolerance (BFT)

Byzantine Fault Tolerance (BFT) is highly technical in nature (like the other consensus algorithms presented); a good explanation can be found here. In general, the BFT consensus algorithms employed by cryptocurrency projects allow generals (validators) to each manage the state of a chain and share messages between each other to arrive at the correct transaction record and to ensure honesty. Again, this topic is fairly nuanced and further additional information regarding the problem and implementation details can be found here. BFT is notably implemented by Ripple (where validators are pre-selected by the Ripple foundation) and Stellar (where anyone can be a validator and trust is established by the community). BFT is advantageous because it presents scalability and low cost transactions, but like DPoS introduces a component of centralization.

Emerging Consensus Algorithms

As mentioned in the introduction, the problem of consensus and transaction verification is difficult and highly nuanced. More consensus algorithms that make different sets of trade-offs are likely to be presented going forward and may replace the current set used.

Decentralized acyclic graphs (DAGs) are currently receiving a lot of attention and present a promising potential solution for scalability (more to come in a future piece). Hashgraph, Tangle, and Block-lattice are three implementations that have received significant recent attention (again, more on this soon—not all attention has been positive).

Conclusion

For the time being, consensus algorithms must make trade-offs between scalability and the degree to which they are centralized (though second-layer networks may change the scalability portion of the equation). It will be fascinating to see which mechanisms are best able to incentivize large-scale participation and stable governance as well as how protocols and communities adapt to incorporate technological developments.

. . .