Proof-of-Stake Primer

Author	josh
Date	2022-01-28

Overview

Every decentralized blockchain needs a consensus mechanism – that is, an algorithm for determining the state of the chain. How can a network of independent nodes, with no central authority, determine which transactions are a valid part of the chain's history? There are several popular consensus algorithms used by cryptocurrencies – primarily *Proof-of-Work* and *Proof-of-Stake.* Let's dive into how proof-of-stake consensus works at a basic level, and why it is useful.

Proof-of-Stake, Why and How

First, Proof-of-Work

Proof-of-Work is, currently, the most common consensus mechanism used by cryptocurrencies. Bitcoin, Litecoin, and others use PoW to secure their chains. Proof-of-work requires "miners" to solve computationally difficult guessing problems – using real-world electricity and, most often, specialized mining hardware (depending on the specific algorithm chosen).

Proof-of-work takes advantage of the strong pseudorandomness of cryptographic hash functions (such as SHA-256). In order to find a block, miners append a random number called a *nonce* to a candidate block header.

This gives some pseudorandom number out the other end. The miner that finds the nonce that gives an output with a certain number of leading zeros wins the race and gets their block added to the blockchain. The other nodes on the network validate this block is correct using the same block header with that nonce. This is because SHA-256 is deterministic. A particular input gives the same output every time.

An example (let's say the difficulty requires 4 leading zeros):

block + 0 -> 1aef0...
block + 1 -> ed032...
block + 2 -> 123ac8... .
.....
block + 4000 -> 0000a2

The miner that finds 4000 as a correct nonce first broadcasts its block to the rest of the nodes, which validate it and add it to the blockchain. The race starts all over again with a new block.

The interesting property here is that the output of SHA-256 is pseudorandom (a property of all strong cryptographic hash functions). So there's no way for miners to predict which nonce will "win" the race, they literally have to guess using as much computing power as possible.

Real blockchains such as Bitcoin operate with a much higher difficulty than, say, 4 leading zeros. The more leading 0-bits required, the more guessing miners have to do. This ties the difficulty of finding a block to the consumption of energy and available hashing power.

Proof-of-Stake, Explained

Enter Proof-of-Stake, an alternative consensus mechanism built on a different scarce resource – the coins of the network itself. In proof-of-stake, validators put up their coins as collateral in exchange for a chance to validate new blocks. Validators are chosen at random, and the chosen validator ensures that all the transactions in their proposed block follow the rules of the network.

If the validator violates the rules, *some or all of their coins are seized as punishment*, or their ability to participate as a validator is removed. The loss of valuable coins serve as incentive to play by the rules, and serves as a cost-barrier to attacking the network.

The algorithm used to select validators varies by the network, just as proof-of-work implementations vary. The general idea, however, is fairly straightforward. Proof-of-stake validator selection needs a source of entropy or pseudo-randomness, just as proof-of-work does. One interesting mechanism for this is a *commit and reveal* scheme. Each validator uses its own CS-PRNG source to generate a random number (usually 256 bits). This number is hashed and published on the blockchain, so no other nodes can know what the actual number is at first (since cryptographic hashes are irreversible). This is the *commit* stage. Then, a *reveal* takes place. Each node reveals the random number – and other nodes validate that the hash of this number matches the committed hash, thus proving they haven’t tried to change/game the committed random value.

For example, let’s say a validator randomly generates the number *0*. The validator publishes the SHA-256 hash of 0, 5feceb66… to the blockchain. Later in the reveal phase, the node publishes that 0 is the value and other nodes verify that 0 indeed gives the hash 5feceb66… (so the value has not been tampered with).

Next, the validators use some mechanism such as bitwise XOR to combine the validators’ random values and choose a number. This number, using a weighted algorithm, will be used to indicate the chosen validator for this block. A key part of this mechanism is that a validator’s probability of being chosen will be *weighted by the amount they stake*, so the more you stake to help the network, the more likely you are to be the chosen validator. This mechanism, however, must also ensure a high enough amount of randomness that the highest value stakers are not chosen every single time, thus centralizing the validation process. A delicate balance must be achieved.

Proof-of-Stake, a Lightweight Alternative

Proof-of-work has some advantages. It is generally considered quite secure, especially as the amount of overall hashpower on the network increases. The more honest hashpower there is on

the network, the harder it is for an attacker to outcompete the honest nodes (thus favoring their own malicious transactions).

However, there are some disadvantages to the proof-of-work system. One of them is simply energy consumption. Bitcoin especially uses quite a large amount of electricity compared to the overall transaction throughput and users, and has garnered some legitimate environmental concerns.

Another disadvantage is security and barrier to entry – at this stage, Bitcoin requires an inordinate amount of hardware and ongoing energy costs to participate, which favors those with large amounts of capital over most everyday users. This, in its own way, serves as a centralizing force.

With proof-of-stake, any user can participate in a staking pool with any amount of token the pool will allow – thus directly contributing to the network's security with their valuable coins, rather than directing investment into a quickly-obsolete mining rig and ongoing electricity costs.

As well, the cost to attack the network is based on the external cost of electricity, rather than any value of the network itself. Justin Bons postulates that it is more expensive to orchestrate a 51% attack on a valuable proof-of-stake network, as an attacker would have to accumulate 51% of the available coin supply to attack the chain, rather than 51% of of the hashpower costs (which could be conceivably much lower than the value of half the network's monetary value).

I find these arguments compelling in favor of proof-of-stake's security and decentralization, but this consensus mechanism is not trouble-free either. There are known problems with PoS that must be solved, such as the "nothing at stake" problem. This issue occurs when there is an arbitrary (or malicious) fork in the blockchain. In this cases, there's no particular monetary incentive for validating the "wrong" chain, since building on either chain can result in the reward for validation. Whereas with proof-of-work, splitting your hashpower to both chains decreases the probability of mining a block (and getting the reward) on either side of the fork. Protocol's such as Ethereum 2.0's PoS implementation help to solve this problem with the same penalties detailed above, where coins can be seized from the validator for behaving dishonestly.

Decentralized Consensus: The Stakes are Important

Whether Proof-of-Work, Proof-of-Stake, or some other algorithm, every decentralized cryptocurrency needs some way for all parties to agree on a valid chain. It's critical for every blockchain to take this task seriously, as the security of the chain is a core part of the cryptocurrency value proposition. Instead of relying on central authorities, we can now build systems that agree on valid transactions via open source software, cryptography, and economics. It's a fascinating system for any computer-science curious person to explore!