Zane Witherspoon Follow
AKA DJ Eth Ledger • CTO of blockchain startup @DispatchLabsIO • Organizer of SF Ethereum Meetup • Follow all of my socials @ZaneWithSpoon
Feb 13 · 8 min read

A Hitchhiker's Guide to Consensus Algorithms

A quick classification of cryptocurrency consensus types

Don't Panic. Behind every great cryptocurrency, there's a great consensus algorithm. No consensus algorithm is perfect, but they each have their strengths. In the world of crypto, consensus algorithms exist to prevent *double spending*. Here's a quick rundown on some of the most popular consensus algorithms to date, from Blockchains to DAGs and everything in-between.

. . .

Proof-of-Work (PoW) -The OG Consensus

Pull a Rihanna and work work work work work

Popular implementations: Bitcoin, Ethereum, Litecoin, Dogecoin, (Most of them)

Pros: We know it works

Cons: Slow throughput; killing the planet

Proof of Work was the first blockchain consensus algorithm. Devised by Satoshi Nakamoto for use in the Bitcoin blockchain, we have PoW to thank for the massive mining operations and power consumption we see around the world. We know it works (which is a lot more that we can say for many other consensus algorithms), but at this stage in the game it's starting to be considered a legacy technology. Even Ethereum is migrating away from PoW for more energy and economically efficient PoS. With so many new alternatives, it's hard to see why a new blockchain would use PoW.

In PoW, *miners* solve hard, useless problems to create blocks. PoW runs on a system of "the longest chain wins." So assuming *most* miners are working on the same chain, that one will grow fastest will be the longest and most trustworthy. Hence Bitcoin is safe as long as more than 50% of the work being put in by miners is honest.

. . .

Proof-of-Stake (PoS) — New kid on the block(chain)

Stakey's going to fork you up

Popular implementations: Decred, Ethereum (soon), Peercoin

Pros: Attacks more expensive; More decentralized; Energy efficient

Cons: Nothing at Stake

In PoS, the blocks aren't created by miners doing work, but by *minters staking* their tokens to "bet" on which blocks are valid. In the case of a fork, minters spend their tokens voting on which fork to support. Assuming most people vote on the correct fork, validators who voted on the wrong fork would "lose their stake" in the correct one.

The common argument against proof-of-stake is the ***Nothing at Stake*** problem. The concern is that since it costs validators almost no computational power to support a fork unlike PoW, validators could vote for both sides of every fork that happens. Forks in PoS could then be much more common than in PoW, which some people worry could harm the credibility of the currency.

. . .

Delegated Proof-of-Stake (DPoS) — Elect your Validators

Just avoid the lobbyists and we'll be fine

Popular Implementations: Steemit, EOS, BitShares

Pros: Cheap transactions; scalable; energy efficient

Cons: Partially centralized

DPoS is the brain-child of Daniel Larimer, and is actually very different from PoS. In DPoS, token hodlers don't vote on the validity of the blocks themselves, but vote to elect delegates to do the validation on their behalf. There are generally between 21–100 elected delegates in a DPoS system. The delegates are shuffled periodically and given an order to deliver their blocks in. Having few delegates allows them to organize themselves efficiently and create designated time slots for each delegate to publish their block. If delegates continually miss their blocks or publish invalid transactions, the stakers vote them out and replace them with a better delegate.

In DPoS, miners can collaborate to make blocks instead of competing like in PoW and PoS. By partially centralizing the creation of blocks, DPoS is able to run orders of magnitude faster than most other consensus algorithms. EOS is set to be the first blockchain with block times < 1 second! A little quicker than bitcoin's 10 minute block times.

• • •

Proof-of-Authority (PoA) — Trust the know it all

You will respect my authority!

Popular Implementations: POA.Network, Ethereum Kovan testnet

Pros: High throughput; scalable

Cons: Centralized system

Proof-of-Authority is a consensus algorithm where transactions are validated by approved accounts, kind of like the "admins" of the system. These accounts are the authority that other nodes receive their truth from. PoA has high throughput, and is optimized for private networks. You're unlikely to see PoA running on a public chain due to its centralized nature.

. . .

Proof-of-Weight (PoWeight) — Bigger is better

Proof-of-Anything

Popular Implementations: Algorand, Filecoin, Chia

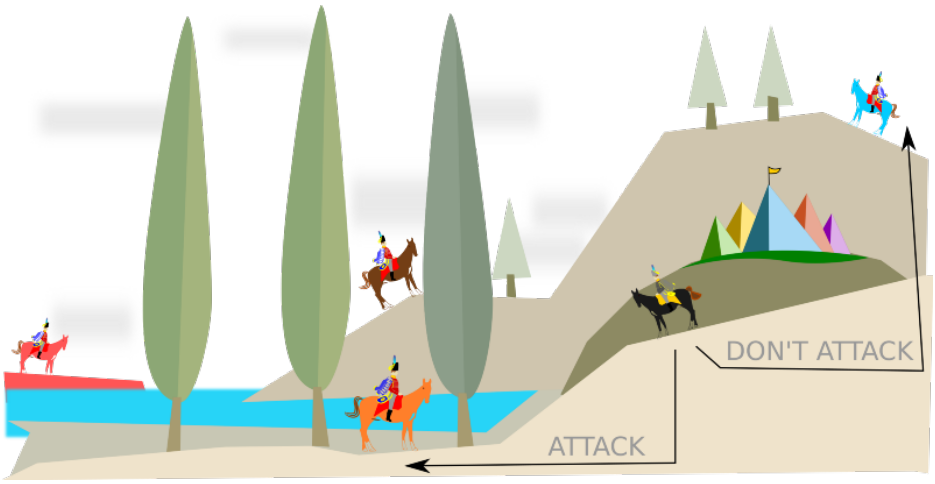
Pros: Customizable; scalable

Cons: Incentivization can be a challenge

Proof-of-Weight is a broad classification of consensus algorithms based around the Algorand consensus model. The general idea is that where in PoS, your percentage of tokens owned in the network represents your probability of "discovering" the next block, in a PoWeight system, some other relatively weighted value is used. Concrete example: Filecoin's Proof-of-Spacetime is weighted on how much IPFS data you're storing. Other systems could include weights for things like Proof-of-Reputation.

. . .

Byzantine Fault Tolerance (BFT) — Siege the blockchain!



Those generals really love sieging cities

Popular Implementations: Hyperledger, Stellar, Dispatch, and Ripple

Pros: High throughput; low cost; scalable

Cons: Semi-trusted

There's this classic problem is distributed computing that's usually explained with Byzantine generals. The problem is that several Byzantine generals and their respective portions of the Byzantine army and have surrounded a city. They must decide in unison whether or not to attack. If some generals attack without the others, their siege will end in tragedy. The generals are usually separated by distance and have to pass messages to communicate. Several cryptocurrency protocols use some version of BFT to come to consensus, each with their own pros and cons:

Practical Byzantine Fault Tolerance (PBFT): One of the first solutions to this problem was coined *Practical Byzantine Fault Tolerance*. Currently in use by Hyperledger Fabric, with few (< 20, after that things get a little) pre-selected generals PBFT runs incredibly efficiently. Pros: High transaction throughput Cons: Centralized/permissioned

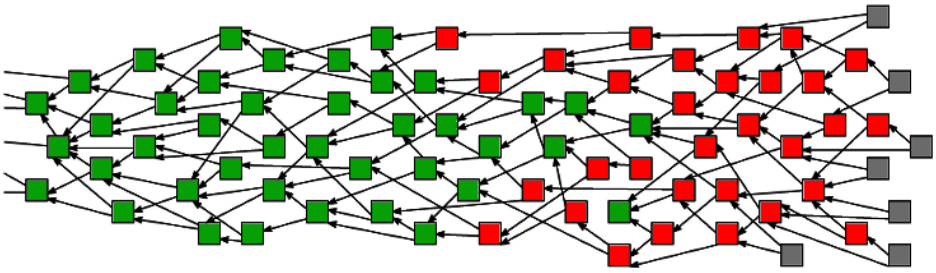
Federated Byzantine Agreement (FBA): FBA is another class of solutions to the Byzantine generals problem used by currencies like Stellar and Ripple. The *general* idea (heh), is that every Byzantine general, responsible for their own chain, sorts messages as they come in

to establish truth. In Ripple the generals (validators) are pre-selected by the Ripple foundation. In Stellar, anyone can be a validator so you choose which validators to trust.

For its incredible throughput, low transaction cost, and network scalability, I believe the FBA class of consensus algorithms are the best we've discovered for distributed consensus.

. . .

Directed Acyclic Graphs (DAGs) — aka the Blockchain Killers!



Spaghetti Consensus

Popular Implementations: Iota, Hashgraph, Raiblocks/Nano

Pros: Network Scalability; low cost

Cons: Depends on implementation

DAGs are hotter than Vitalik's Tinder profile right now. DAGs are a form of consensus that doesn't use the blockchain data structure and handles transactions mostly asynchronously. The big pro is theoretically infinite transactions per second, but DAGs have strengths and weaknesses like any other consensus.

Tangle: Tangle is the DAG consensus algorithm used by Iota. In order to send an Iota transaction, you need to validate two previous transactions you're received. The two-for-one, pay-it-forward consensus strengthens

the validity of transactions the more transactions are added to the Tangle. Because the consensus is established by the transactions, theoretically, if someone can generate 1/3 of the transactions they could convince the rest of the network their invalid transactions are valid. Until there's enough transaction volume that creating 1/3rd of the volume becomes unfeasible, Iota is sort-of "double-checking" all of the network's transactions on a centralized node called "The Coordinator". Iota says The Coordinator works like training wheels for the system, and will be removed once the Tangle is big enough.

Hashgraph: Hashgraph is a gossip-protocol consensus developed by Leemon Baird. Nodes share their known transactions with other nodes at random so eventually all the transactions are gossiped around to all of the nodes. Hashgraph is really fast (250,000+ transactions per second) but isn't resistant to Sybil attacks. So Hashgraph is a great option for private networks, but you're not going to see it implemented in a public network like Ethereum or Dispatch any time soon.

Block-lattice: Nano (formerly Raiblocks) runs with a twist on the blockchain called a *Block-lattice.* The Block-lattice is a structure where every user (address) gets their own chain that only they can write to, and everyone holds a copy of all of the chains. Every transaction is broken down into both a send block on the sender's chain and a receive block on the receiving party's chain. The Block-lattice seems almost too simple to work, but it's already out there running in the wild. The unique structure does leave the Block-lattice open to some unique attack vectors like the *Penny-spend* attack, where attackers inflate the number of chains node must keep track of by sending negligible amounts to a wide array of empty wallets.

SPECTRE: *Serialization of Proof-of-work Events: Confirming Transactions via*
Recursive Elections, better known as SPECTRE, is a proposed Bitcoin scaling solution that utilizes a combination of PoW and DAGs to reach scalable consensus. In SPECTRE, the blocks are mined pointing to multiple parents, not just one, so the network could potentially handle multiple blocks per second. Mining a block pointing to some parent blocks supports those blocks validity. Compared to PoW's "longest chain

wins", SPECTRE uses something like "blocks with the most childen win." SPECTRE hasn't been battle-tested in the wild yet, and new attack vectors are likely to emerge, but it feels like a very clever potential way to fix Bitcoin.

. . .

Did I miss your favorite algorithm? Love the post? Feedback is always appreciated! Hopefully you've found reading this guide as helpful as I found writing it to be. Big thanks to my team at Dispatch Labs for helping me edit and noodle around the best consensus ever! Shoutout Hal Finney🙏 And thanks to all the blockchain builders out there who've gotten us this far ❤️

> *If you found this article useful or entertaining please drop a 👏 or a share. Follow me on Twitter for more insights twitter.com/ZaneWithSpoon*