title: Single slot finality description: Explanation of single slot finality lang: en

Single slot finality {#single-slot-finality}

It takes about 15 minutes for an Ethereum block to finalize. However, we can make Ethereum's consensus mechanism validate blocks more efficiently and decrease time-to-finality dramatically. Instead of waiting for fifteen minutes, blocks could get proposed and finalized in the same slot. This concept is known as **single slot finality (SSF)**.

What is finality? {#what-is-finality}

In Ethereum's proof-of-stake based consensus mechanism, finality refers to the guarantee that a block cannot be altered or removed from the blockchain without burning at least 33% of the total staked ETH. This is 'crypto-economic' security because confidence comes from the extremely high cost associated with changing the order or content of the chain that would prevent any rational economic actor from trying it.

Why aim for quicker finality? {#why-aim-for-quicker-finality}

The current time to finality has turned out to be too long. Most users do not want to wait 15 minutes for finality, and it is inconvenient for apps and exchanges that might want high transaction throughput to have to wait that long to be certain their transactions are permanent. Having a delay between a block's proposal and finalization also creates an opportunity for short reorgs that an attacker could use to censor certain blocks or extract MEV. The mechanism that deals with upgrading blocks in stages is also quite complex and has been patched several times to close security vulnerabilities, making it one of the parts of the Ethereum codebase where subtle bugs are more likely to arise. These issues could all be eliminated by reducing the time to finality to a single slot.

The decentralization / time / overhead tradeoff {#the-decentralization-time-overhead-tradeoff}

The finality guarantee is not an immediate property of a new block; it takes time for a new block to finalize. The reason for this is that validators representing at least 2/3 of the total staked ETH on the network has to vote for the block ("attest") in order for it to be considered finalized. Each validating node on the network has to process attestations from other nodes in order to know that a block has, or has not, achieved that 2/3 threshold.

The shorter the time allowed to reach finalization, the more computing power is required at each node because the attestation processing has to be done faster. Also, the more validating nodes exist on the network, the more attestations have to be processed for each block, also adding to the processing power required. The more processing power required, the fewer people can participate because more expensive hardware is needed to run each validating node. Increasing the time between blocks lessens the computing power required at each node but also lengthens the time to finality, because attestations are processed more slowly.

Therefore, there is a trade-off between the overhead (computing power), decentralization (number of nodes that can participate in validating the chain) and time to finality. The ideal system balances minimum computing power, maximum decentralization and minimum time to finality.

Ethereum's current consensus mechanism balanced these three parameters by:

- **Setting the minimum stake to 32 ETH** This sets an upper limit on the number of validators' attestations that have to be processed by individual nodes, and therefore an upper limit on computational requirements for each node.
- **Setting the time to finality at ~15 minutes** This gives sufficient time for validators run on normal home computers to safely process attestations for each block.

With the current mechanism design, in order to reduce the time to finality, it is necessary to reduce the number of validators on the network or increase the hardware requirements for each node. However, there are improvements that can be made

to the way attestations are processed that can allow more attestations to be counted without adding to the overhead at each node. The more efficient processing will allow finality to be determined within a single slot, rather than across two epochs.

Routes to SSF {#routes-to-ssf}

The current consensus mechanism combines attestations from multiple validators, known as committees, to reduce the number of messages each validator has to process to validate a block. Every validator has an opportunity to attest in each epoch (32 slots) but in each slot, only a subset of validators, known as a 'committee' attest. They do so by dividing up into subnets in which a few validators are selected to be 'aggregators'. Those aggregators each combine all the signatures they see from other validators in their subnet into a single aggregate signature. The aggregator that includes the greatest number of individual contributions passes their aggregate signature to the block proposer, who includes it in the block along with the aggregate signature from the other committees.

This process provides sufficient capacity for every validator to vote in each epoch, because `32 slots * 64 committees * 256 validators per committee = 524,288 validators per epoch`. At the time of writing (February 2023) there are ~513,000 active validators.

In this scheme, it is only possible for every validator to vote on a block by distributing their attestations across the whole epoch. However, there are potentially ways to improve the mechanism so that *every validator has the chance to attest in every slot*.

Since the Ethereum consensus mechanism was designed, the signature aggregation scheme (BLS) has been found to be far more scalable than was initially thought, while the ability of clients to process and verify signatures has also improved. It turns out that processing attestations from a huge number of validators is actually possible within a single slot. For example, with one million validators each voting twice in each slot, and slot times adjusted to be 16 seconds, nodes would be required to verify signatures at a minimum rate of 125,000 aggregations per second in order to process all 1 million attestations within the slot. In reality, it takes a normal computer around 500 nanoseconds to do one signature verification, meaning 125,000 can be done in ~62.5 ms - far below the one second threshold.

Further efficiency gains could be made by creating supercommittees of e.g. 125,000 randomly selected validators per slot. Only these validators get to vote on a block and therefore only this subset of validators decide whether a block is finalized. Whether this is a good idea or not comes down to how expensive the community would prefer a successful attack on Ethereum to be. This is because instead of requiring 2/3 of the total staked ether, an attacker could finalize a dishonest block with 2/3 of the staked ether *in that supercommittee*. This is still an active area of research, but it seems plausible that for a validator set sufficiently large to require supercommittees in the first place, the cost of attacking one of those subcommittees will be extremely high (e.g. the ETH denominated cost of attack would be `2/3 * 125,000 * 32 = ~2.6 million ETH`). The cost of attack can be adjusted by increasing the size of the validator set (e.g. tune the validator size so the cost of attack is equal to 1 million ether, 4 million ether, 10 million ether, etc). Preliminary polls of the community seem to suggest that 1-2 million ether is an acceptable cost of attack, which implies ~65,536 - 97,152 validators per supercommittee.

However, verification is not the true bottleneck - it is signature aggregation that really challenges validator nodes. To scale signature aggregation will probably require increasing the number of validators in each subnet, increasing the number of subnets, or adding additional layers of aggregation (i.e. implement committees of committees). Part of the solution might be allowing specialized aggregators - similar to how block building and generating commitments for rollup data will be outsourced to specialized block builders under proposer-builder separation (PBS) and Danksharding.

What is the role of the fork-choice rule in SSF? {#role-of-the-fork-choice-rule}

Today's consensus mechanism relies on a tight coupling between the finality gadget (the algorithm that determines whether 2/3 of validators have attested to a certain chain) and the fork choice rule (the algorithm that decides which chain is the correct one when there are multiple options). The fork choice algorithm only considers blocks *since* the last finalized block. Under SSF there wouldn't be any blocks for the fork choice rule to consider, because finality occurs in the same slot as the block is proposed. This means that under SSF *either* the fork choice algorithm *or* the finality gadget would be active at any time. The finality gadget would finalize blocks where 2/3 of validators were online and attesting honestly. If a block is not

able to exceed the 2/3 threshold, the fork choice rule would kick in to determine which chain to follow. This also creates an opportunity to maintain the inactivity leak mechanism that recovers a chain where >1/3 validators go offline, albeit with some additional nuances.

Outstanding issues {#outstanding-issues}

The problem with scaling aggregation by growing the number of validators per subnet is that it leads to greater load on the peer-to-peer network. The problem with adding layers of aggregations is that it is quite complex to engineer and adds latency (i.e. it could take longer for the block proposer to hear from all the subnet aggregators). It is also not clear how to deal with the scenario that there are more active validators on the network than can feasibly be processed in each slot, even with BLS signature aggregation. One potential solution is that, because all validators attest in every slot and there are no committees under SSF, the 32 ETH cap on the effective balance could be removed entirely, meaning operators managing multiple validators could consolidate their stake and run fewer, reducing the number of messages that validating nodes have to process to account for the entire validator set. This relies on large stakers agreeing to consolidate their validators. It is also possible to impose a fixed cap on the number of validators or the amount of staked ETH at any time. However, this requires some mechanism for deciding which validators are allowed to participate and which are not, which is liable to create unwanted secondary effects.

Current progress {#current-progress}

SSF is in the research phase. It is not expected to ship for several years, likely after other substantial upgrades such as Verkle trees and Danksharding.

Further reading {#further-reading}

- Vitalik on SSF at EDCON 2022
- Vitalik's notes: Paths to single slot finality