---

title: Proof-of-stake rewards and penalties description: Learn about the in-protocol incentives in proof-of-stake Ethereum. lang: en

---

Ethereum is secured using its native cryptocurrency, ether (ETH). Node operators that wish to participate in validating blocks and identifying the head of the chain deposit ether into a smart contract on Ethereum. They are then paid in ether to run validator software that checks the validity of new blocks received over the peer-to-peer network and apply the fork-choice algorithm to identify the head of the chain.

There are two primary roles for a validator: 1) checking new blocks and "attesting" to them if they are valid, 2) proposing new blocks when selected at random from the total validator pool. If the validator fails to do either of these tasks when asked they miss out on an ether payout. Validators are also sometimes tasked with signature aggregation and participating in sync committees.

There are also some actions that are very difficult to do accidentally and signify some malicious intent, such as proposing multiple blocks for the same slot or attesting to multiple blocks for the same slot. These are "slashable" behaviors that result in the validator having some amount of ether (up to 1 ETH) burned before the validator is removed from the network, which takes 36 days. The slashed validator's ether slowly drains away across the exit period, but on Day 18 they receive a "correlation penalty" which is larger when more validators are slashed around the same time. The consensus mechanism's incentive structure therefore pays for honesty and punishes bad actors.

All rewards and penalties are applied once per epoch.

Read on for more details...

Rewards and penalties {#rewards}

Rewards {#rewards}

Validators receive rewards when they make votes that are consistent with the majority of other validators, when they propose blocks, and when they participate in sync committees. The value of the rewards in each epoch are calculated from a `base_reward`. This is the base unit that other rewards are calculated from. The `base_reward` represents the average reward received by a validator under optimal conditions per epoch. This is calculated from the validator's effective balance and the total number of active validators as follows:

```
base_reward = effective_balance * (base_reward_factor / (base_rewards_per_epoch * sqrt(sum(active_balance))))
```

where `base_reward_factor` is 64, `base_rewards_per_epoch` is 4 and `sum(active balance)` is the total staked ether across all active validators.

This means the base reward is proportional to the validator's effective balance and inversely proportional to the number of validators on the network. The more validators, the greater the overall issuance (as `sqrt(N)` but the smaller the `base_reward` per validator (as `1/sqrt(N)`). These factors influence the APR for a staking node. Read the rationale for this in Vitalik's notes.

The total reward is then calculated as the sum of five components that each have a weighting that determines how much each component adds to the total reward. The components are:

```
1. source vote: the validator has made a timely vote for the correct source checkpoint 2. target vote: the
validator has made a timely vote for the correct target checkpoint 3. head vote: the validator has made a timely
vote for the correct head block 4. sync committee reward: the validator has participated in a sync committee 5.
proposer reward: the validator has proposed a block in the correct slot
```

The weightings for each component are as follows:

```
TIMELY_SOURCE_WEIGHT uint64(14) TIMELY_TARGET_WEIGHT uint64(26) TIMELY_HEAD_WEIGHT uint64(14) SYNC_REWARD_WEIGHT
uint64(2) PROPOSER_WEIGHT uint64(8)
```

These weights sum to 64. The reward is calculated as the sum of the applicable weights divided by 64. A validator that has made timely source, target and head votes, proposed a block and participated in a sync committee could receive `64/64 * base_reward == base_reward`. However, a validator is not usually a block proposer, so their maximum reward is `64-8 /64 *`

`base_reward == 7/8 * base_reward`. Validators that are neither block proposers nor in a sync committee can receive `64-8-2 / 64 * base_reward == 6.75/8 * base_reward`.

An additional reward is added to incentivize rapid attestations. This is the `inclusion_delay_reward`. This has a value equal to the `base_reward` multiplied by `1/delay` where `delay` is the number of slots separating the block proposal and attestation. For example, if the attestation is submitted within one slot of the block proposal the attestor receives `base_reward * 1/1 == base_reward`. If the attestation arrives in the next slot, the attestor received `base_reward * 1/2` and so on.

Block proposers receive `8 / 64 * base_reward` for **each valid attestation** included in the block, so the actual value of the reward scales with the number of attesting validators. Block proposers can also increase their reward by including evidence of misbehavior by other validators in their proposed block. These rewards are the "carrots" that encourage validator honesty. A block proposer which includes slashing will be rewarded with the `slashed_validators_effective_balance / 512`.

Penalties {#penalties}

So far we have considered perfectly well-behaved validators, but what about validators that do not make timely head, source and target votes or do so slowly?

The penalties for missing the target and source votes are equal to the rewards the attestor would have received had they submitted them. This means that instead of having the reward added to their balance, they have an equal value removed from their balance. There is no penalty for missing the head vote (i.e. head votes are only rewarded, never penalized). There is no penalty associated with the `inclusion_delay` - the reward will simply not be added to the validator's balance. There is also no penalty for failing to propose a block.

Read more about rewards and penalties in the consensus specs. Rewards and penalties were adjusted in the Bellatrix upgrade - watch Danny Ryan and Vitalik discuss this in this Peep an EIP video.

Slashing {#slashing}

Slashing is a more severe action that results in the forceful removal of a validator from the network and an associated loss of their staked ether. There are three ways a validator can be slashed, all of which amount to the dishonest proposal or attestation of blocks:

- By proposing and signing two different blocks for the same slot
- By attesting to a block that "surrounds" another one (effectively changing history)
- By "double voting" by attesting to two candidates for the same block

If these actions are detected, the validator is slashed. This means that 1/32 of their staked ether (up to a maximum of 1 ether) is immediately burned, then a 36 day removal period begins. During this removal period the validator's stake gradually bleeds away. At the mid-point (Day 18) an additional penalty is applied whose magnitude scales with the total staked ether of all slashed validators in the 36 days prior to the slashing event. This means that when more validators are slashed, the magnitude of the slash increases. The maximum slash is the full effective balance of all slashed validators (i.e. if there are lots of validators being slashed they could lose their entire stake). On the other hand, a single, isolated slashing event only burns a small portion of the validator's stake. This midpoint penalty that scales with the number of slashed validators is called the "correlation penalty".

Inactivity leak {#inactivity-leak}

If the consensus layer has gone more than four epochs without finalizing, an emergency protocol called the "inactivity leak" is activated. The ultimate aim of the inactivity leak is to create the conditions required for the chain to recover finality. As explained above, finality requires a 2/3 majority of the total staked ether to agree on source and target checkpoints. If validators representing more than 1/3 of the total validators go offline or fail to submit correct attestations then it is not possible for a 2/3 supermajority to finalize checkpoints. The inactivity leak lets the stake belonging to the inactive validators gradually bleed away until they control less than 1/3 of the total stake, allowing the remaining active validators finalize the chain. However large the pool of inactive validators, the remaining active validators will eventually control >2/3 of the stake. The loss of stake is a strong incentive for inactive validators to reactivate as soon as possible! An inactivity leak scenario

was encountered on the Medalla testnet when < 66% of active validators were able to come to consensus on the current head of the blockchain. The inactivity leak was activated and finality was eventually regained!

The reward, penalty and slashing design of the consensus mechanism encourages individual validators to behave correctly. However, from these design choices emerges a system that strongly incentivizes equal distribution of validators across multiple clients, and should strongly disincentivize single-client dominance.

Further reading {#further-reading}

- [Upgrading Ethereum: The incentive layer](#)
- [Incentives in Ethereum's hybrid Casper protocol](#)
- [Vitalik's annotated spec](#)
- [Eth2 Slashing Prevention Tips](#)

Sources

- *https://benjaminion.xyz/eth2-annotated-spec/phase0/beacon-chain/*