

A Comprehensive Study of Ethereum Staking Cryptoeconomics:

Consensus Layer Rewards and Penalties

[Andy Yang](#)

TL;DR

1. A validator can get rewards from attestation, sync committee participation, block proposals, and whistleblow malicious activities.
2. Attestation rewards are about 84.375% of the optimal reward per epoch in the consensus layer.
3. A validator must have at least 42.86% uptime to break even.
4. Optimal Annualized Staking Rewards per Validator in the consensus layer is about 0.997 ETH, with 3.11% APR.

1 Introduction

On September 6, 2022, Ethereum underwent a significant transition from Proof-of-Work (PoW) to Proof-of-Stake (PoS) through the Paris upgrade, also known as "The Merge." This leap marked a significant shift from ETH 1 to ETH 2 (a.k.a. "Serenity"). Following this milestone shift, "Staking" stepped onto the stage, enabling validators to secure the network and earn rewards. At the same time, "Mining" faded into the ashes of history and no longer plays a significant role in the Ethereum consensus mechanism.

To ensure a seamless and well-guided transformation, the Ethereum Foundation has meticulously outlined multiple phases of the Serenity upgrades in EIP-2982, providing a roadmap for the network's continued evolution:

1. Phase 0: Launch of the Beacon Chain and PoS system for Decentralization and Security
2. Phase 1: Sharding for Scalability
3. Phase 1.5: Bridge the gap between Phase 0 and Phase 1
4. Phase 2: Provide smart contract execution for shard chains.

Due to ongoing discussions within the Ethereum community, the roadmap for Ethereum has also undergone some changes. During [EthCC Paris](#), Vitalik proposes a new roadmap: "The Merge," "The Surge," "The Verge," and "The Purge." Since the traditional "Sharding" model

is difficult and time-consuming to implement, "Proto-Danksharding" and "Danksharding" have taken its place for now.

Based on our five years of experience with PoS systems, we recognize that Ethereum's cryptoeconomics mechanisms could be more straightforward to understand under frequent changes in the roadmap and multiple system upgrades. As the cryptoeconomics of Ethereum have stabilized, we have written this report as a beginning of our three-part series:

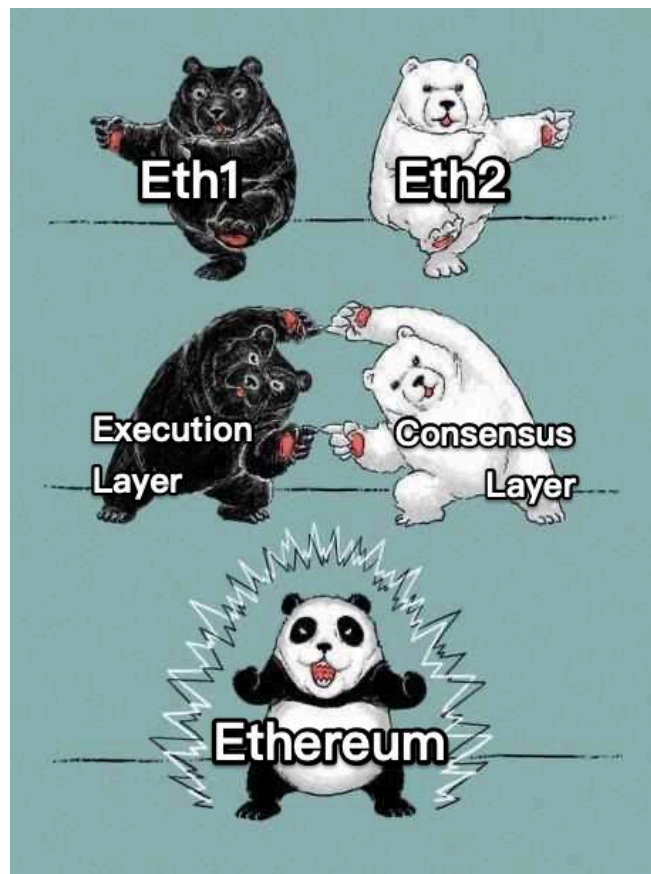
- Introduction & Consensus Layer Rewards and Penalties
- Execution Layer Rewards
- Inactivity leak and Slashing

2 Breakdown by Layers: EL & CL

In this section, we first give an introduction to two Post-Merge layers. Then, we break down the validator rewards allocated by the execution and consensus layers.

2.1 Post-Merge Layers

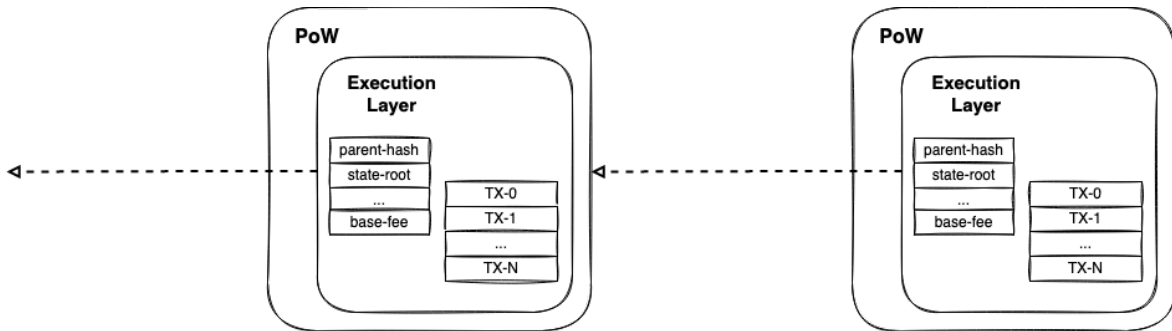
Generally, there are two layers in Ethereum: "ETH1" and "ETH2". As of late 2021, these terms are renamed "Execution Layer" and "Consensus Layer" in [The Great Naming Note](#) for better clarity and inclusivity. For simplicity, we'll call these two layers CL and EL in the following sections.



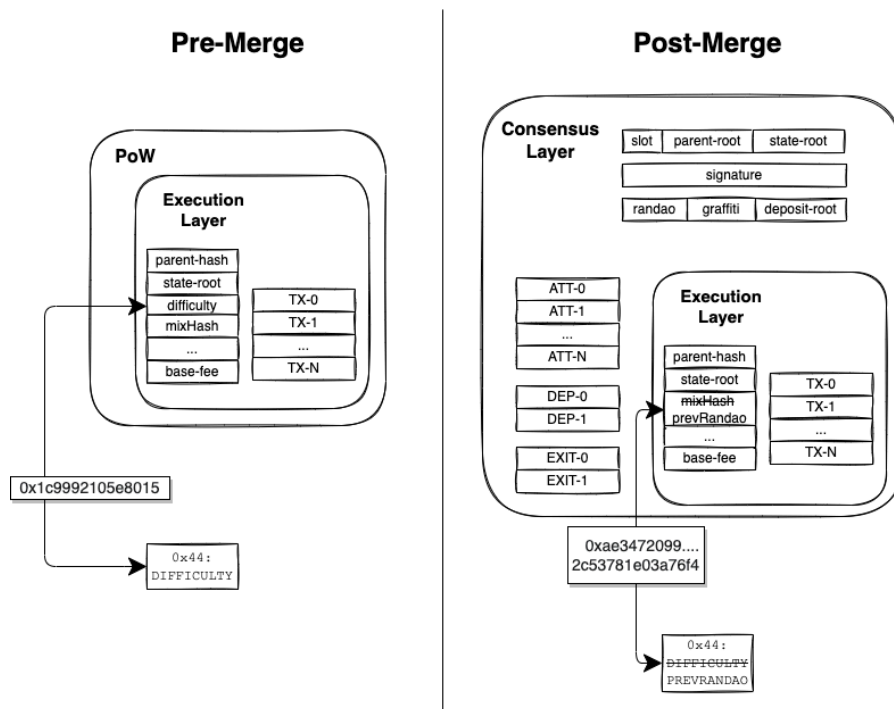
The main difference between the two layers is their responsibilities.

- Execution Layer: Responsible for all non-consensus duties: Listening and processing transactions, running smart contracts, and storing data.
- Consensus Layer: Responsible for consensus-related duties: Choosing fork, processing attestations, finalizing transactions, and producing beacon blocks.

A Pre-Merge PoW block looks like this:



After “The Merge”, a Beacon block wrapped the modified PoW block:



There are several changes:

- **Opcode Change:** **DIFFICULTY** opcode returns PoW difficulty in the block header at the Pre-Merge period. [EIP-4399](#) replaced **DIFFICULTY** opcode for PoW with a beacon chain random number opcode called **prevRandao**.

- **Block Structure Change:** After the opcode changes, PoW blocks become execution blocks assembled in consensus blocks (or beacon blocks). We can see a strange thing here: there are two **state-root** fields. These two duplicate fields store different data: The consensus layer is responsible for storing the state of the beacon chain, and the execution layer is responsible for storing the EVM account's state.

We won't dive into the specs, as a later report might cover how these two layers or chains work together.

2.2 Breakdown Rewards: EL and CL

You can receive both EL and CL rewards as a validator in Ethereum. But in the dark forest of Crypto, there is no free lunch. You have to play your role in the network accordingly, even being penalized if you don't do it well. Since rewards in these two layers depend on your roles and responsibilities in the network, we'll cover these terms first to ensure clarity.

2.2.1 Network Roles

- **Validator:** The validator is an entity that deposits 32 ETH into the network and receives rewards for maintaining the consensus of the ledger. Any of the roles mentioned below are played by validators. Network assigned different duties to them.
- **Proposer:** [Ethereum measures time in slots and epochs](#). Each slot is 12 (**SECONDS_PER_SLOT**) seconds, and 32 (**SLOTS_PER_EPOCH**) slots (6.4 minutes) make up an epoch. At each slot, a proposer is randomly chosen among all active validators. The probability of being selected is weighted by its effective balance. **Once chosen, the proposer must produce a block in that slot.**
- **Attester:** The attestation is the process where validators examine new blocks and provide their opinions on their validity. **Validators must process one attestation per epoch, and each attestation corresponds to a specific slot.**
- **Sync Committee Participant:** The sync committee is a new feature introduced in the Altair upgrade to reduce network overhead for light clients. This committee has 512 (**SYNC_COMMITTEE_SIZE**) validators randomly selected and refreshed every 256 epochs (**EPOCHS_PER_SYNC_COMMITTEE_PERIOD**). [The selected validators are responsible for continuously signing the block header that represents the new head of the chain at each slot.](#)

2.2.2 Network Duties and Rewards

Execution Layer

1. Priority fee/ tips

Following EIP-1559, Ethereum now burns the base fee. As a result, the block proposer now considers the priority fee/ tips as the primary economic incentive. Paying tips incentivize the block proposer to get transactions included in the block as soon as possible, thus fastening the transaction confirmation process.

2. MEV (Maximum Extractable Value)

Maximal extractable value (MEV) refers to the maximum value that can be extracted from block production in excess of the standard block reward and gas fees by including, excluding, and changing the order of transactions in a block.

Consensus Layer

To dive into CL rewards, we must first understand the basic units of rewards accounting. Many sources still used deprecated rewards accounting methods before the Altair upgrade. That's why we are focusing on this issue at the beginning of this section.

In Phase 0, the base reward is the optimal reward per epoch:

$$\text{base reward} = \text{validator effective balance} * \text{base reward factor} / (\text{base reward per epoch} * \sqrt{\sum \text{active balance}})$$

The base reward equals that amount divided by the applied times of the base reward. (**BASE_REWARDS_PER_EPOCH**). Where **base_reward_factor** is 64, **base_rewards_per_epoch** is 4, and **sum**(active balance) is the total staked ETH across all active validators.

Proposer reward is $\frac{1}{8}$ of base reward, where **proposer reward quotient** is 8:

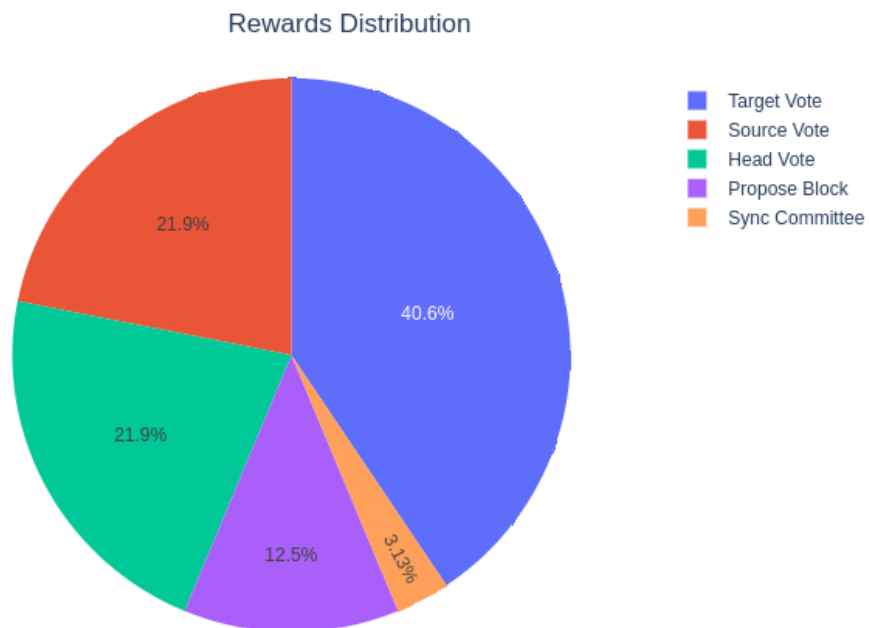
$$\text{proposer reward} = \text{base reward} / \text{proposer reward quotient}$$

Altair upgrade reformed the rewards accounting method by giving different duties different weights and base reward calculations. The incentive of the shift is that slashing penalties will overflow out of the maximum of uint64 mentioned in [issue #1284 of the Ethereum consensus specification](#). This change will result in a lower accuracy of the slashing penalty, and sometimes, the slashing penalty will be 0. However, this does not affect the protocol security,

as it is mentioned in [issue #1322](#) that the slashed validator has to pay a base penalty of 1 ETH, even if the calculated slashing penalty is 0. Vitalik also says that changing the **EFFECTIVE_BALANCE_INCREMENT** from 1 ETH to 0.01 ETH can increase accuracy for future needs.

Duty	Parameter	Type and Value
Source Vote	TIMELY_SOURCE_WEIGHT	uint64(14)
Target Vote	TIMELY_TARGET_WEIGHT	uint64(26)
Head Vote	TIMELY_HEAD_WEIGHT	uint64(14)
Participate Sync Committee	SYNC_REWARD_WEIGHT	uint64(2)
Propose Block	PROPOSER_WEIGHT	uint64(8)
N/A	WEIGHT_DENOMINATOR	uint64(64)

The **WEIGHT_DENOMINATOR** is 64, and the reward weight for accomplishing each part of the duty is $X / \text{WEIGHT_DENOMINATOR}$. In the case of a timely head vote, for example, the weight of the reward for a correct and timely head vote is **14/64**.



We perform some calculations for the base reward to quantify the rewards validators can get.

The exact balances change every epoch (due to rewards and penalties), so Ethereum stores them in a compact array that requires rehashing only <32 MB to update, while the effective balances (which are used for all other computations that require validator balances) are updated using a hysteresis formula: if the effective balance is n ETH, and if the exact balance goes below $n-0.25$ ETH, then the effective balance goes to $n-1$ ETH. If the exact balance goes above $n+1.25$ ETH, the effective balance goes to $n+1$ ETH.

Since the exact balance must change by at least a full 0.5 ETH to trigger an effective balance update, this ensures an attacker can't make effective balances update every epoch and thus cause processing the chain to become very slow by repeatedly nudging the exact balances above, and then below some threshold.

All rewards are calculated as a "base reward per increment." An increment is one unit of effective balance. An active validator has up to 32 increments. The **EFFECTIVE_BALANCE_INCREMENT** is 1 ETH, **BASE_REWARD_FACTOR** is 64, and the **total_active_balance** is the amount of all staked effective ETH.

$$base_reward_per_increment = \frac{EFFECTIVE_BALANCE_INCREMENT * BASE_REWARD_FACTOR}{\sqrt{total_active_balance}}$$

"Increments" seems difficult to understand. Simply put, if a validator has a 32 ETH effective balance, it has 32 increments.

$$increments = \frac{validator_effective_balance}{EFFECTIVE_BALANCE_INCREMENT}$$

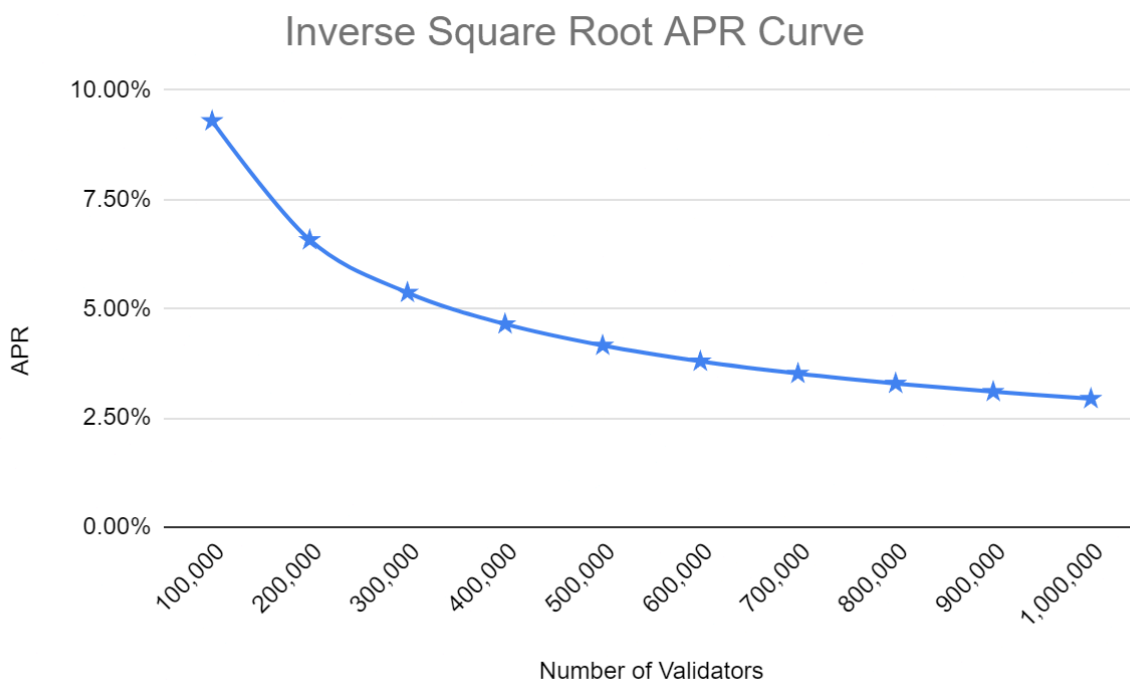
All rewards are proportional to the effective balance a validator has. For attestations, base reward calculation proves it explicitly. Assume a validator has n **increments** and b **base_reward_per_increment**. Its expected long-term return per epoch is nb .

$$base_reward = base_reward_per_increment * increments$$

Meanwhile, the effective balance influences the chosen probability for sync committee participants and block proposers. If the **total increments of all active validators** are T, the chosen probability is scaled with n/T.

$$\text{chosen probability} \propto \frac{\text{increments}}{\text{total increments}}$$

Assume we have a validator with an effective balance of 32 ETH. Rated Network shows there are 888,406 active validators with 28,428,992 ETH effective balances on 2023.12.6. According to [our calculation on the spreadsheet](#), a validator can earn 12128 Gwei per epoch on a long-term average. That means a validator can earn about 0.996012 ETH on average per year (82125 epochs) with 3.11% APR. Recall the calculation of **base_reward_per_increment**. The following figure shows the base reward is inverse root scaling by the total active staked balance. As the number of validators increases, the base reward decreases. Since attestation rewards are proportional to base_reward, it will also decrease.



On the contrary, the proposal reward might increase. Since every attester needs to finish an attestation per epoch, attestations per slot will also increase when the total number increases. A proposer can receive rewards from every attestation included in the block.

Thus, the proposer may get more rewards due to increasing attestations. But money doesn't grow on trees. Since the selected probability is proportional to the ratio of your active balance divided by the total active balance, the chances of being a validator are also reduced.

1. Attestation Rewards and Penalties

Before digging into the rabbit hole, let's see some basics.

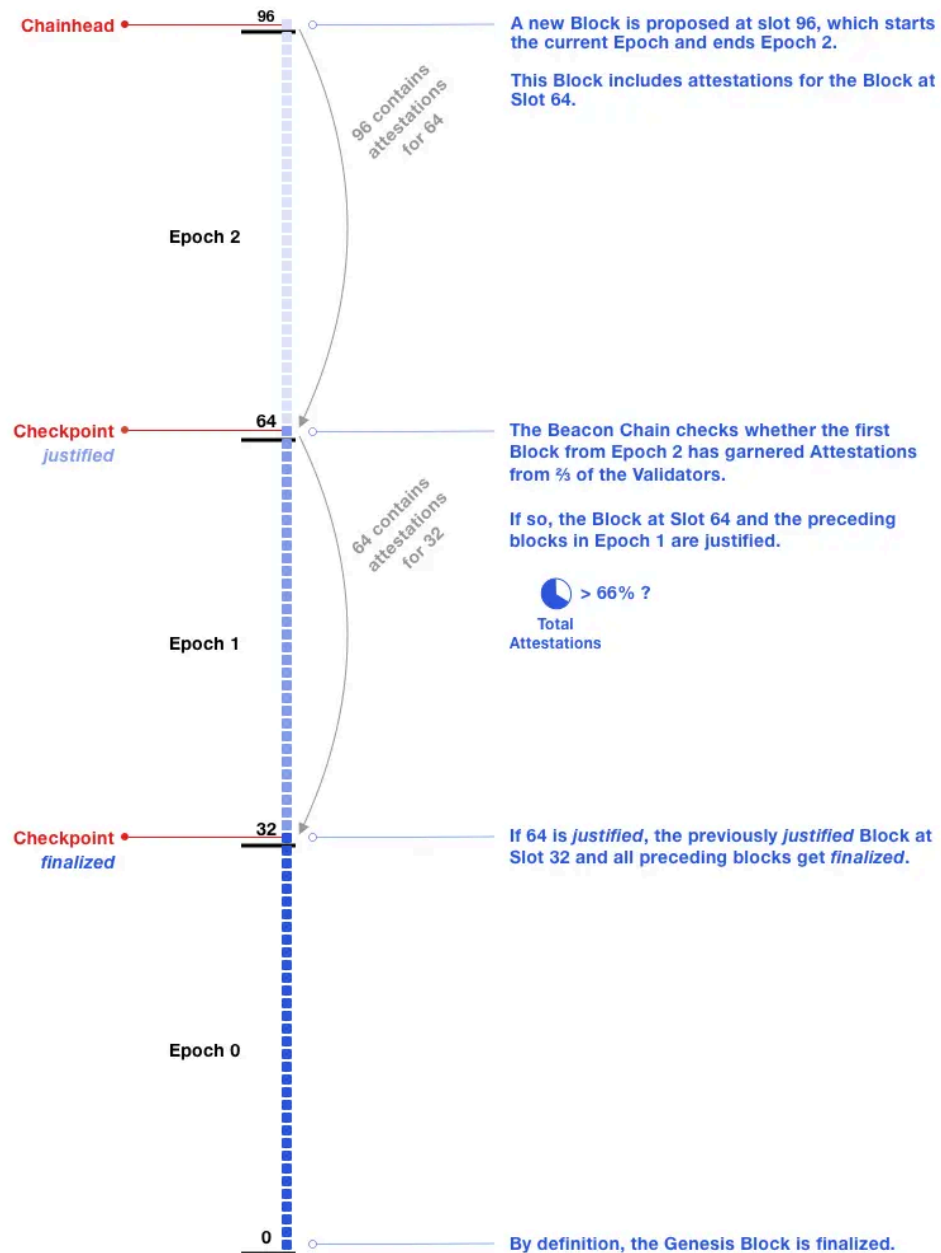
- **Supermajority:** A vote is a supermajority vote if over 2/3 staked balance on it.
- **Checkpoint:** A checkpoint is a block in the first slot of an epoch or the preceding most recent blocks (if the block at the first slot is missed).
- **Justified:** When an epoch ends, if its checkpoint gained a 2/3 of supermajority, this checkpoint is justified.
- **Finalized:** If checkpoint A is justified and checkpoint B becomes justified in the next epoch, then A becomes finalized.
- **Casper FFG:** Casper the Friendly Finality Gadget (Casper FFG) is an overlap atop a block proposal mechanism for finality.
- **LMD GHOST:** Latest Message Driven Greedy Heaviest-Observed Sub-Treeis (LMD-Ghost) is a fork choice rule used by nodes to determine the best chain.

An attestation contains three votes:

Source Vote: Submit an attestation that correctly identifies the source (recent justified block) of Casper FFG, and gets included within $\text{sqrt}(\text{SLOTS_PER_EPOCH})=5$ slots.

Target Vote: Submit an attestation that correctly identifies the target (next block to be justified) of Casper FFG, and gets included within $\text{SLOTS_PER_EPOCH} = 32$ slots.

Head Vote: submit an attestation that correctly identifies the head of the chain, and gets included within the next ($\text{MIN_ATTESTATION_INCLUSION_DELAY} = 1$) slot.



Attesters will only be rewarded if their votes are completed **correctly** and **timely**. Since the attestation could be up to 32 slots long, it could have been generated in the current epoch or the previous epoch, so the first thing Ethereum does is to check the epoch of the attestation's target vote to see which epoch in the beacon state we should be looking at. If the target vote is in the current epoch, then the justified checkpoint is the current justified checkpoint. Otherwise, it is the previous justified checkpoint.

Next, Ethereum determines the **correctness** of three votes, noting that only when the previous conditions are met will the following judgment be made. This implies that these votes are interlocked: **source vote** → **target vote** → **head vote**. If a vote is incorrect, the subsequent votes will also be incorrect. Consequently, a validator must first complete the correct source vote to receive the reward for the target vote. Similarly, to obtain the reward for the head vote, a validator must first complete both the valid source and target vote.

Lastly, Ethereum evaluates the **timeliness** of the three votes:

“**Inclusion delay**” is the measurement for timeliness. It refers to the interval between the current slot and the slot where the attestation was made. For the block containing the attestation to be valid, the inclusion_delay must lie in the open interval of **[MIN_ATTESTATION_INCLUSION_DELAY, SLOTS_PER_EPOCH] = [1, 32]**. In other words, a valid attestation must be included in the next block or any block after up to 32 slots.

The **inclusion delay** restriction is a new requirement introduced after the Altair upgrade. In Phase 0, all valid votes receive rewards, with additional rewards inversely proportional to the inclusion delay. We attached the formula here:

$$\textit{Inclusion Delay Reward} = \textit{base reward} / \textit{inclusion delay}$$

After the Altair upgrade, the timeliness restrictions mentioned before replaced inclusion delay rewards. The change aims to reward behaviors that contribute to chain development more accurately. For example, voting for the correct head is always more profitable in Phase 0, even if it means waiting longer and risking not being included in the next slot. However, rewarding the right head vote for 30 slots late makes little sense.

The entire timeliness judgment process is as follows:

1. Assuming the source vote is correct, we determine if the inclusion delay of the source vote is completed within **integer_sqrt(SLOTS_PER_EPOCH)= 5** slots (or less).
2. Assuming the target vote is correct, we check if the inclusion delay of the target vote is completed within **SLOTS_PER_EPOCH=32** slots (or less).

Target vote is always helpful, but if the timespan of the target vote is at most a few epochs, it is relatively straightforward. Therefore, 32 slots serve as a reasonable constraint. This check is redundant because a block's attestation must be at least 32 slots.

3. Assuming the head vote is correct, we verify if the inclusion delay of the head vote exactly equals **MIN_ATTESTATION_INCLUSION_DELAY=1**.

Since head votes have only one empty slot, the head vote will become invalid if the block at the next slot is missed or late. However, this is undoubtedly beyond the attester's control. Therefore, head votes only receive rewards and are not penalized after the Altair upgrade (Phase 0 does have a penalty for the wrong head vote). So well-performing validators can maintain a balanced income during the inactivity leak period, as we expect at least 1/3 of blocks to be lost: they won't receive rewards, but they won't be penalized either.

Here is an attestation reward matrix:

Source Vote	Target Vote	Head Vote	Reward	Penalty	Result	Value (Gwei)
✗	✗	✗	0	40/64 * base_reward	-40/64 * base_reward	-7580
✓	✗	✗	14/64 * base_reward	26/64 * base_reward	-12/64 * base_reward	-2274
✓	✓	✗	40/64 * base_reward	0	40/64 * base_reward	7580
✓	✓	✓	54/64 * base_reward	0	54/64 * base_reward	10233

The reward may deviate from the results in this table due to the following reasons:

1. **Other Validators complete tasks imperfectly:** e.g., The next slot cannot include the validator's attestation and sync committee signature if the proposer is offline or poorly connected.

2. **Low Participation Rate:** The reward for a Validator performing attestation duty is proportional to the percentage of all validators performing that duty. For example, if only 80% of the validators fulfill their duties, the reward for each validator who performs their duties is multiplied by 0.8.

There is an additional type of reward known as the **slashing whistleblower reward**. The treatment of these rewards differs because predicting the number of slashing events in advance is impossible. Moreover, they do not contribute to the issuance of Ether because the slashed validator receives a larger slashing penalty, which offsets the rewards earned by the informants of the slashing events. If a proposer includes “slashing” evidence in their proposal, he can get rewards as the formula says:

$$\text{whistleblower reward} = \text{validator_effective_balance} // \text{WHISTLEBLOWER_REWARD_QUOTIENT}$$

Where **WHISTLEBLOWER_REWARD_QUOTIENT = 512**. We can do a simple calculation for this scenario, assuming the slashed validator has a 32 ETH effective balance. There are $32 * 10^9 // 512 = 62500000$ Gwei = 0.0625 ETH rewards. The attester who includes the slashed evidence can get $7/8 * 0.0625 = 0.054678$ ETH.

2. Sync Committee Rewards and Penalties

Sync committee rewards emerged in the Atair upgrade. Only one sync committee is active at any one time, and it contains a randomly selected subset of the total set of validators whose size is **SYNC_COMMITTEE_SIZE=512**. A sync committee performs its duties for only **EPOCHS_PER_SYNC_COMMITTEE_PERIOD=256** epochs until the following committee takes over.

Since Ethereum introduced this component to serve light clients, we'll start with some introductions first. Light nodes are nodes that run light client software. They can interact with the blockchain and effectively verify the state without storing a local copy of the blockchain data. With the sync committee, light clients can process only block headers to update their knowledge of the latest Ethereum blocks. This allows Ethereum nodes to run on mobile phones and IoT devices where computing resources are scarce, thus creating **ubiquitous Ethereum**. As an [Ethereum.org article](#) says: “Imagine a bicycle rental service(opens in a new tab) that uses an app with an embedded light client to verify that you own the rental service's NFT quickly and if so, unlocks a bicycle for you to ride away on!”

It is important to note that the sync committee has no slashing, so its security relies on the honest trust assumption. However, as long as a significant portion of the entire set of Ethereum validators is honest, it is statistically unlikely that the sync committee will sign off on an invalid block header, as its percentage of honesty will reflect the honesty of the entire set of Ethereum validators.

Like being chosen to propose a block, the probability of serving as a sync committee participant is also proportional to a validator's effective balance. For any given validator, this will rarely happen.

Sync committee participants are rewarded for every slot in which they correctly perform their duties. With 512 members in the committee and 32 slots per epoch, **the reward per validator per slot** for correct participation is:

$$\text{sync committee rewards} = \frac{\text{SYNC_REWARD_WEIGHT}}{\text{SLOTS_PER_EPOCH} * \text{SYNC_COMMITTEE_SIZE} * \text{WEIGHT_DENOMINATOR} * T * b}$$

T refers to the total increments of the active validator set, so this is a large number.

We already mentioned that there are 888,406 active validators. By some calculations, we have $2 / (64 * 512 * 32) * 888406 * 12128 \text{ Gwei} \approx 20550 \text{ Gwei}$.

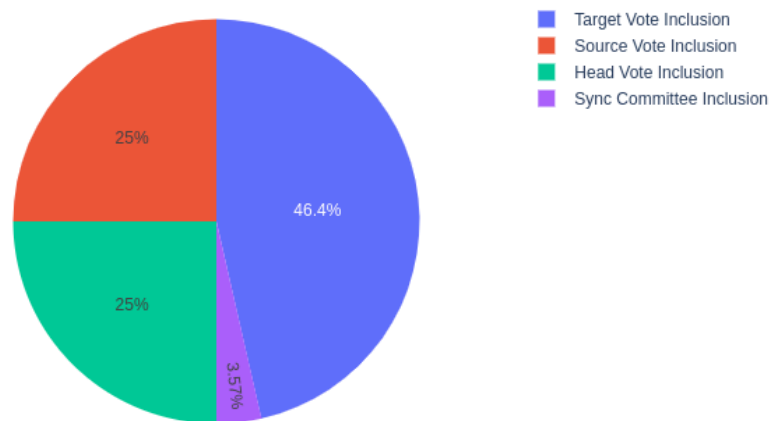
If a validator is chosen in the sync committee that doesn't participate (sign the wrong head block or doesn't show up), it receives a penalty equal to the reward it would have earned for being correct. The block proposer receives no rewards for the missing contribution.

Participant	Reward	Penalty	Result	Value (Gwei)
✘	0	$1/524288 * \text{base_reward}$	$-1/524288 * \text{base_reward}$	-20550
✔	$1/524288 * \text{base_reward}$	0	$1/524288 * \text{base_reward}$	20550

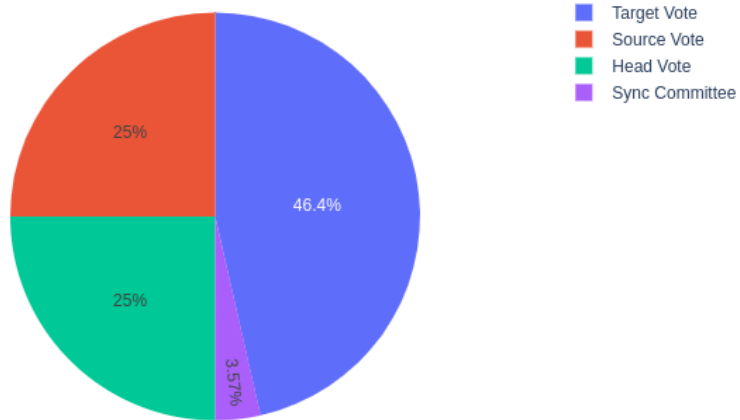
3. Proposal Rewards and Penalties

The proposal rewards for a validator are aligned with attestation rewards and sync committee rewards. That means including the corresponding vote could earn the same percentage of rewards as the attestation and sync committee counterparts. The proposal weight equals $8/64 = 1/8$, which means $7/8$ of rewards go to attesters and sync committee participants who perform their duties, and $1/8$ of rewards go to proposers who include the evidence in the blocks. If an attester creates a correct, timely source vote and is included by a proposer, he can get $14/54 * 8/64 * 8 = 1/27$. Suppose everyone (including the proposers) performs perfectly at everything. In that case, the attestation rewards add up to $54/64$ of the base reward, and the proposer rewards add up to $54/56$ of the $8/64$ proposer share (i.e., the remaining $8/64$ of the base reward). Note that the proposer's reward is determined by the attestation numbers included in the block. We created the following pie charts to make the reward distribution more straightforward.

Proposal Rewards Distribution



Non-Proposal Rewards Distribution



Given the formula as follows:

$$R = \frac{\text{total_active_validators}}{(\text{SLOTS_PER_EPOCH} * \text{attestation_rewards_per_slot})}$$

$$\text{attestation proposal rewards} = \frac{\text{PROPOSER_WEIGHT}}{(\text{WEIGHT_DENOMINATOR} - \text{PROPOSER_WEIGHT})} * R$$

Where R is the total attestation rewards in a block for attesters, since each validator must make an attestation per epoch, we have $888406 // 32 = 27762$ validators per slot. A block contains $27762 * 10233 \text{ Gwei} = 0.284088546 \text{ ETH}$ attestation rewards. A proposer got $1/7 * R = 0.040584078 \text{ ETH}$ rewards. According to the [ETH2 book](#): “The reward could be higher if the chain is not performing perfectly, as after a skip slot, the proposer can include high-value attestations from the missed slot.”

Proposers can also get rewards by including the sync committee’s output. For the whole committee, a block proposer can get:

$$\text{sync committee proposal rewards} = \frac{\text{SYNC_COMMITTEE_SIZE} * \text{PROPOSER_WEIGHT}}{\text{WEIGHT_DENOMINATOR} - \text{PROPOSER_WEIGHT}}$$

According to our previous calculation, the single sync committee contribution per validator per slot is 20550 Gwei. So we get $512 / 7 * 20550 \text{ Gwei} = 1,503,086 \text{ Gwei} \approx 0.0015 \text{ ETH}$.

Source Vote Included	Target Vote Included	Head Vote Included	Reward	Penalty	Result	Value (Gwei)
✗	✗	✗	0	0	0	0
✓	✗	✗	$14/56 * \text{base_reward} * 8/64-8$	0	$14/56 * \text{base_reward} * 8/64-8$	-2274
✓	✓	✗	$40/56 * \text{base_reward} * 8/64-8$	0	$40/56 * \text{base_reward} * 8/64-8$	7580
✓	✓	✓	$54/56 * \text{base_reward} * 8/64-8$	0	$54/56 * \text{base_reward} * 8/64-8$	10233

We mentioned whistleblower rewards in the attestation reward section before. A proposer who includes the slashed evidence can get $1/8 * \text{whistleblower reward} = 0.0078125 \text{ ETH}$.

The probability of being chosen to propose a block is proportional to effective balance. If you have a 32 ETH effective balance, and another validator has a 16 ETH effective balance, you are twice as likely to be selected.

Missing a proposal will not cause any penalty. However, the reward for proposing a block is relatively high. Missing a proposal means losing a lot of ETH, thus incentivizing proposers to perform their duties well.

2.2.3 Break-Even Uptime for Validators

Suppose there are two kinds of validators: fully functioning online and offline validators, where p is the proportion of fully functioning and online ($p = 0.9 \Rightarrow p = 90\%$). Then, the maximum reward (long-term average) for a fully functioning online validator is $\text{base_reward} * (50/64 * p + 14/64 * p^2)$. The derivation follows:

First, consider the attestation and sync committee rewards for 56/64 and ignore the proposer rewards for 8/64. All rewards will be based on the collective reward mechanism (considering the participation rate p) multiplied by p .

Assuming that X is the maximum reward for completing a given duty, and p is the proportion of verifiers who complete that duty, completing that duty will result in a $p * X$ reward. Failing to do so will result in a penalty of X .

The rewards and penalties are different because we need penalties to ensure that attestation is a net gain if you're online at least about 2/3 of the time. The inactivity leak period might be activated if validators can't meet the minimum online time. If so, a validator's optimal reward is reduced to 0, while bad-performing validators are penalized with increasing penalties. We'll introduce more details in the last part of our series: "Inactivity Leak and Slashing" if you are interested.

Thus, the maximum possible reward ratio for Non-Proposal Rewards is $42/64 * p + 14/64 * p^2$. The head vote is multiplied by p twice for:

- **The first p** is multiplied by the proportion of validators that perform attestation.

- **The second p** is multiplied by the proportion of validators that are proposing blocks. We use p here because the proposer is chosen pseudo-randomly among all active validators. (We don't consider the influence of effective balance here)

Now consider Proposal Rewards. Since included attestation derives it, we only need to worry about the proportion of duties the attester performs, so simply multiply all the rewards by the first p , which is $8/64 * p$.

When added together, the total percentage is $50/64 * p + 14/64 * p^2$.

If a validator is offline, the penalty percentage is $-42/64$. So we have $50/64 * p + 14/64 * p^2 - 42/64 (1-p)$. If the equation is greater than 0, a validator will make profits. Through some mathematics, we figure that if $p > 3/7$ ($\approx 42.86\%$), a validator can break even. So don't worry too much about occasional downtime.

2.2.4 Optimal Rewards Table for a Single Validator

Duty	Expected Time	Duration	Rewards (Ether)
Attestation	Once per epoch (6.4 minutes)	1 slot	0.000010233
Block Proposal	Per two months	1 slot	0.04208716371
Sync Committee Participation	Per five years	256 epochs (~ 1 day 3 hours)	0.1683456
Slashing Whistleblower	Unpredictable	1 slot	At most 0.0625 ETH (1/8 for proposer, 7/8 for attester)

Using `base_reward` as the accounting unit might not be easy to read, so we made a table to show the optimal rewards in each case.

Appendix

1. [Consensus Layer Rewards Calculation Spreadsheet](#)

Reference

1. [Upgrading Ethereum | 2.8 The Incentive Layer](#)
2. [Proof-of-stake rewards and penalties | ethereum.org](#)
3. [The Ethereum 2.0 Beacon Chain Explained | Consensys](#)
4. [Phase0 Consensus Specs](#)
5. [Capella Consensus Specs](#)
6. [Phase0 Annotated Specs](#)
7. [Altair Annotated Specs](#)
8. [Rewards and Penalties - beaconcha.in Knowledge Base](#)
9. [How does my validator earn ETH? - EthStaker KB](#)