Blockchain and Distributed Ledger technologies

SAPIENZA
UNIVERSITÀ DI ROMA

Massimo La Morgia
massimo.lamorgia@uniroma1.it

Account Balance and Effective Balance

Each validator has 2 balance: actual balance and effective balance.

Actual balance: It is the sum of any deposits made for it via the deposit contract, plus accrued rewards, minus accrued penalties and withdrawals. The actual balance is updated at least once per epoch for all active validators, and every slot for sync committee participants.
The measurement units of the actual balance are the Gwei (10^{-9} ETH).

Effective balance: This balance is derived from the actual balance, using as units for the balance whole ETH and and changes to the effective balance are subject to **hysteresis**.
Its balance is capped at 32 ETH.

Even if the effective balance is capped at 32 ETH, a validator's actual balance could be much higher; for example, if a double deposit had been accidentally made, a validator would have an actual balance of 64 ETH but an effective balance of only 32 ETH.

Hysterisis

Hysterisis is the dependence of the state of a system on its history. It is used to make the effective balance vary much more slowly than actual balances.

The hysteresis levels are controlled by the hysteresis parameters:

Name	Value
HYSTERESIS QUOTIENT	4
HYSTERESIS DOWNWARD MULTIPLIER	1
HYSTERESIS UPWARD MULTIPLIER	5

These are applied at the **end of each epoch** during effective balance updates.
Every validator in the state (whether active or not) has its effective balance updated as follows:

- If actual balance is less than effective balance minus 0.25 then reduce the effective balance.

$$\frac{\text{HYSTERESIS DOWNWARD MULTIPLIER}}{\text{HYSTERESIS QUOTIENT}} = \frac{1}{4} = 0{,}25$$

- If actual balance is more than effective balance plus 1.25 then increase the effective balance by an increme

$$\frac{\text{HYSTERESIS UPWARD MULTIPLIER}}{\text{HYSTERESIS QUOTIENT}} = \frac{5}{4} = 1{,}25$$

Account Balance and Effective Balance

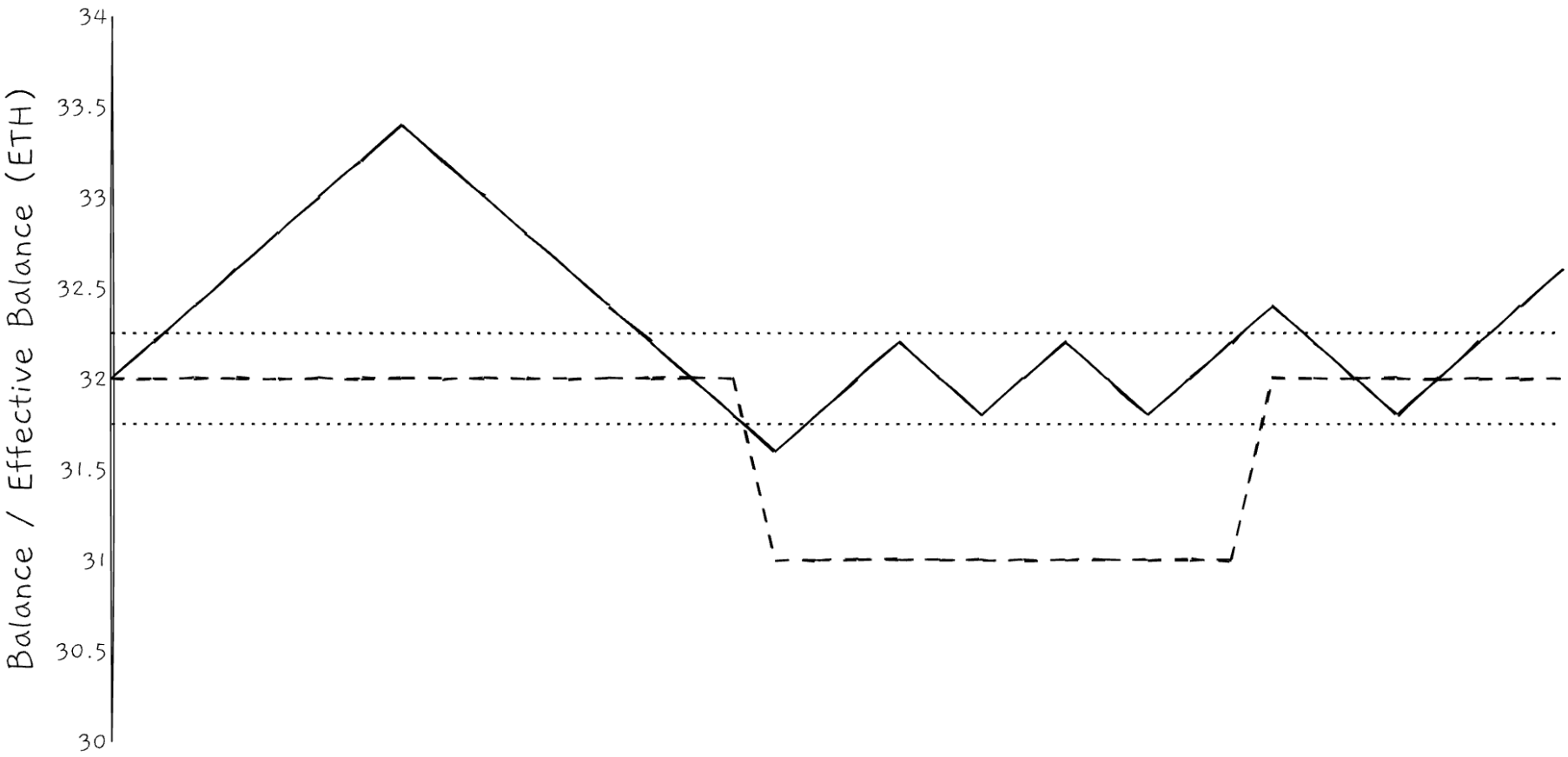


Illustration of the relationship between the actual balance (solid line) and the effective balance (dashed line) of a validator. The dotted lines are the thresholds at which the effective balance gets updated - the hysteresis

Issuance

Issuance is the amount of new Ether created by the protocol in order to incentivise its participants.

In Ethereum rewards are calculated in terms of a **base reward per increment** for brevity b.

$$b = \frac{\text{EFFECTIVE\_BALANCE\_INCREMENT} \times \text{BASE\_REWARD\_FACTOR}}{\sqrt{\text{TOTAL ACTIVE BALANCE}}}$$

b value are reported in Gwei.

An increment is one unit of effective balance, which is 1 ETH (EFFECTIVE_BALANCE_INCREMENT), so active validators have up to 32 increments.

BASE_REWARD_FACTOR is parameter that is used to tweak the issuance rate. It is set at 64.

TOTAL ACTIVE BALANCE is the total staked ether across all active validators.

Considering a network made of 500'000 validators (N), the total issuance for a block (slot) is:

$$b = \frac{64 \times 1000000000}{\sqrt{32 \times 10^9 \times N}} = 505\ Gwei$$

Issuance

$$b = \frac{\text{EFFECTIVE\_BALANCE\_INCREMENT x BASE\_REWARD\_FACTOR}}{\sqrt{\text{TOTAL ACTIVE BALANCE}}}$$

The **total issuance** is given by Tb. Where T is the total amount of the effective balances in Ether. Assuming that there are N validators, all with an effective balance of 32ETH, we have that the total issuance is given by $32Nb$.

Considering again a network made of 500'000 validators (N), the total issuance for a block (slot) is:

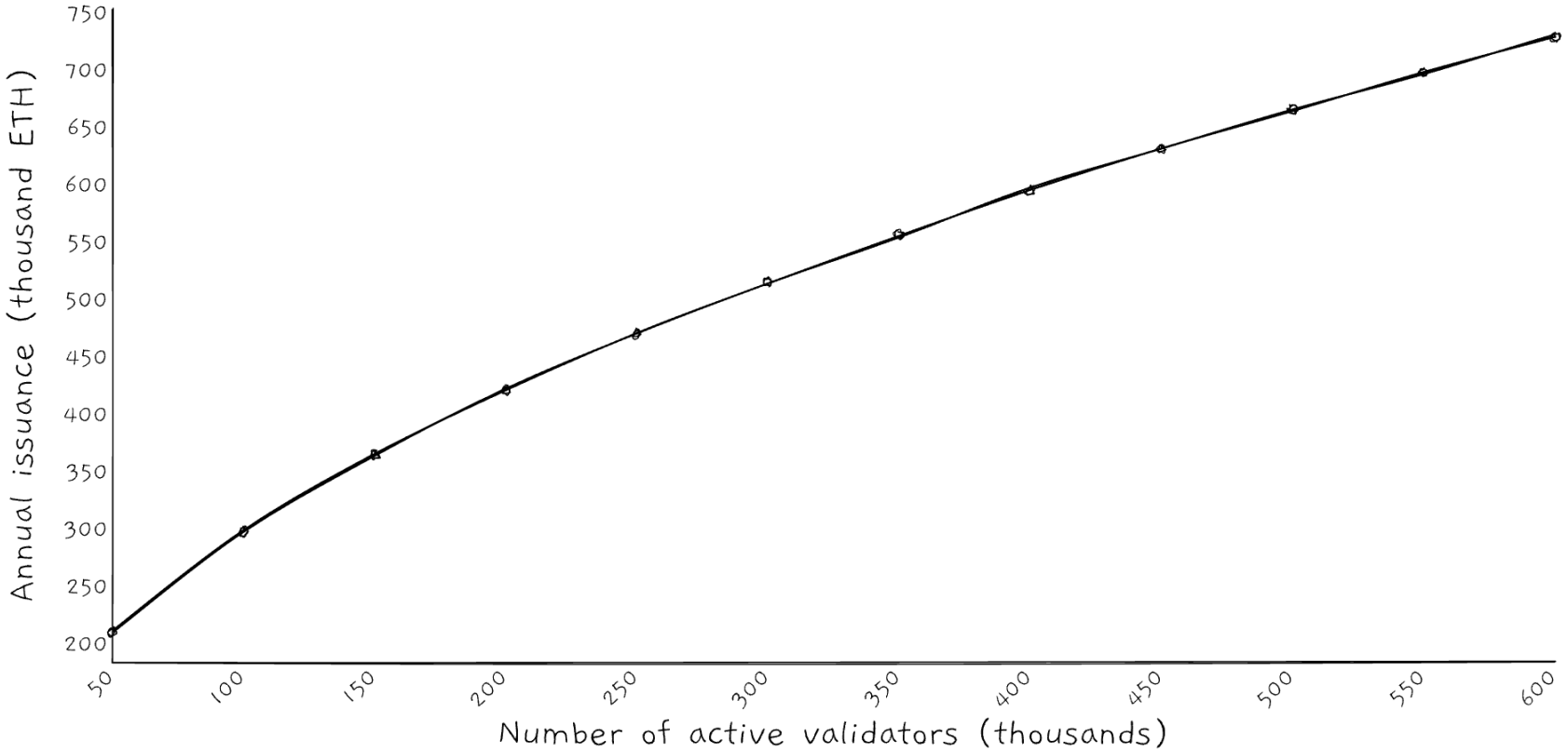
$$\frac{\text{EFFECTIVE\_BALANCE\_INCREMENT x BASE\_REWARD\_FACTOR x } N}{\sqrt{\text{TOTAL ACTIVE BALANCE}}}$$

$$= \frac{32 \text{ x } 64 \text{ x } N}{\sqrt{32\text{x } 10^{-9} \text{ x } N}} = \frac{32 \text{ x } 64 \text{ x } 500000}{\sqrt{32\text{x } 10^{9}\text{x } 500000}} = 8{,}095 \; \boldsymbol{ETH}$$

This means that the total issuance is proportional to the validator's effective balance and inversely proportional to the number of validators on the network.

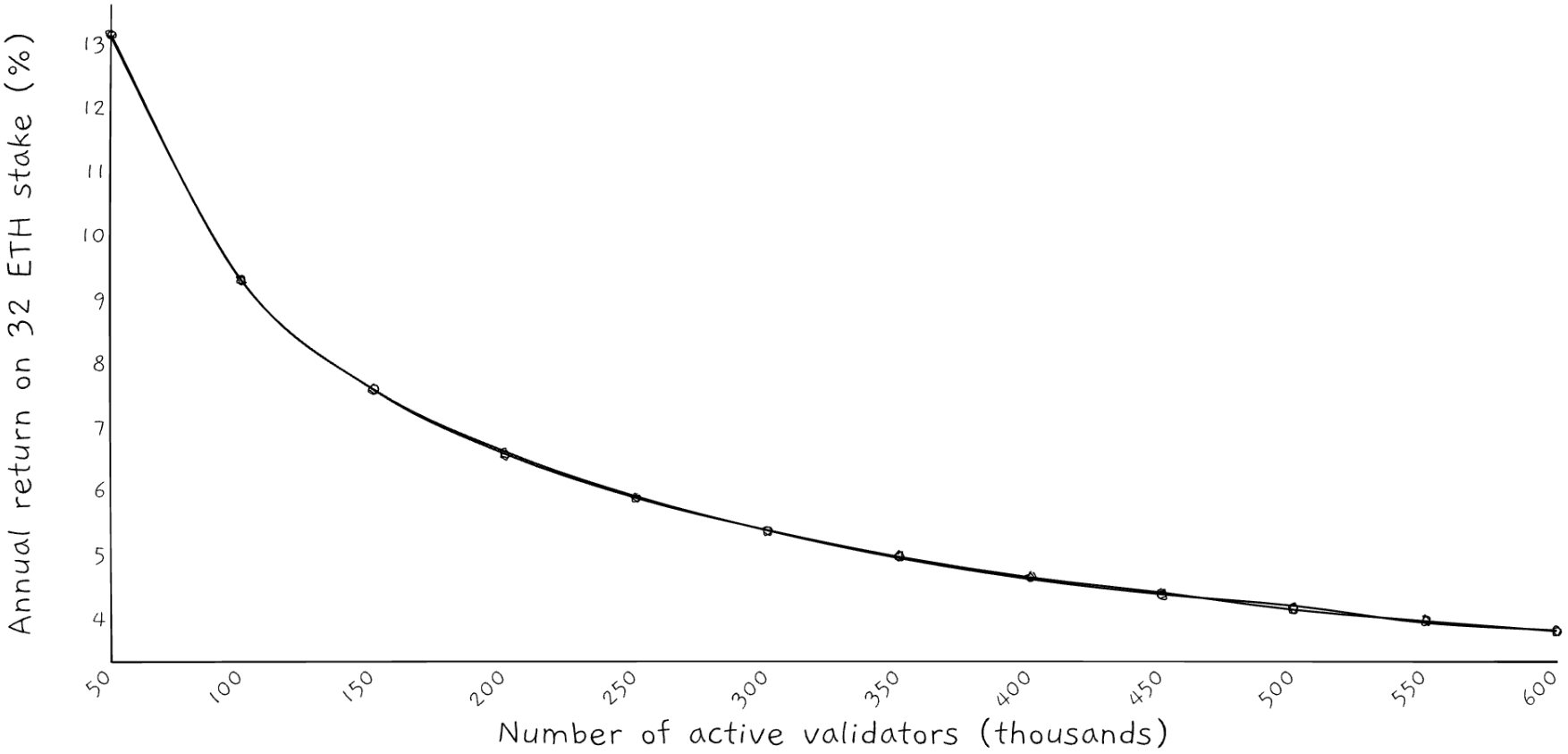
The more validators, the greater the overall issuance but the smaller the base reward per increment of the validator (as 1/sqrt(N)). These factors influence the APR (Annual percentage rate) for a staking node.

Issuance



Maximum annual protocol issuance on the beacon chain as a function of the number of active validators.

Issuance



The expected annual percentage rewards for stakers as a function of the number of active validators.

Attestatation Rewards

The protocol incentivizes each validator to behave well by providing rewards for its activities.

The total reward is 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. voting for a source checkpoint for Casper FFG;
2. voting for a target checkpoint for Casper FFG; and
3. voting for a chain head block for LMD-GHOST.
4. Proposing blocks.
5. Signing off on blocks in the sync committees that support light clients.

The weightings for each component are as follows:

Name	Value	Symbol
TIMELY_SOURCE_WEIGHT	14	W_s
TIMELY_TARGET_WEIGHT	26	W_t
TIMELY_HEAD_WEIGHT	14	W_h
SYNC_REWARD_WEIGHT	2	W_y
PROPOSER_WEIGHT	8	W_p
WEIGHT_DENOMINATOR	64	W_Σ

Moreover, rewards scale with the effective balance and are proportional to the type of activities.

Rewards

These weights sum to 64. The reward is calculated as the sum of the applicable weights divided by 64.

Thus, a validator that has made timely source, target and head votes, proposed a block and participated in a sync committee could receive:

$$\frac{64}{64} bn$$

where n is the effective balance.

Attestation Rewards

The largest part, 84.4%, of validators' rewards come from making attestations.

Although committee and slot assignments for attesting are randomised, every active validator will be selected to make exactly one attestation each epoch.

Attestations receive rewards ***only*** if they are included in beacon chain blocks.

Recall, that the attestation contains 3 votes.

Each vote is eligible for a reward as long as it is correct and meets the timeliness requirements.

Validity	Timeliness	Reward
Correct source	Within 5 slots	$\frac{W_s}{W_\Sigma} nb$
Correct source and target	Within 32 slots	$\frac{W_t}{W_\Sigma} nb$
Correct source, target and head	Within 1 slot	$\frac{W_h}{W_\Sigma} nb$

These are cumulative, so the maximum attestation reward per epoch (for getting all three votes correct and getting the attestation included the next block) is $\frac{W_s+W_t+W_h}{W_\Sigma} \boldsymbol{nb}$, or $0.84375\boldsymbol{nb}$.

Attestation rewards

Correct means that the **attestation agrees with the view of the** blockchain that the current block **proposer** has.

If the attesting validator votes for different checkpoints or head blocks then it is on a different fork and that vote is not useful.

Moreover, the validator will not get rewards if it votes for a head of a fork that will not become the canonical one. This is done in order to disincentives attacks.

Other than this, there are several reasons that could cause an honest validator, operating correctly, to lose rewards for reasons beyond its control. Examples include:

- The proposer can go offline, causing the next block to be skipped.
- The next proposer may be on a minority fork.

Proposer rewards for attestations

Block proposers receive

$$\frac{8}{64}\ bn = \frac{1}{8}\ bn$$

for each valid attestation included in the block, where n refers to the effective balance of the attester.
The actual value of the reward scales with the number of attesting validators.

If the attestations in a block are worth a total of R in rewards to the attesters, then the proposer that includes the attestations in a block receives a reward of:

$$R_{A_P} = \frac{W_p}{W_\Sigma - W_p} R = \frac{8}{56} R = \frac{1}{7} R$$

Thus, a proposer is strongly incentivized to include high value attestations, which basically means including them quickly, and including well-packed, as correct as possible aggregates.

Sync committee

A sync committee is a special group of 512 randomly selected validators whose job is to help light clients keep track of the latest state of the Ethereum blockchain without downloading full blocks.

A sync committee lasts 256 epochs ≈ 27.3 hours. After each period, a new set of 512 validators is selected at random.

Every slot, members of the sync committee:

- verify the header of the most recent block
- sign the header (not the full block)
- contribute to an aggregate signature and a bitfield indicating which of the 512 members participated

This produces a compact proof that can be checked very efficiently.

Light clients:

- do not download full blocks
- do not execute full state transitions
- rely on sync committee signatures to ensure the latest block header is authentic and canonical (resistant to adversarial attacks)

Sync committee rewards

Sync committee participants receive a reward for every slot that 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

$$R_Y = \frac{W_y}{32 \times 512 \times W_\Sigma} Tb$$

The T here is the total increments of the whole active validator set.

The per-epoch per-validator reward is 32 times this.

Thus, assuming that there are 500'000 validators and each of them has an effective balance of 32ETH

$$T = 32 * 500000$$

$$R_Y = \frac{2}{32 \times 512 \times 64} 32 * 500000 * b$$

Proposer rewards for sync committee

As with attestations, the block proposer that includes the sync committee's output receives a reward proportional to the reward of the whole committee:

$$R_{Y_P} = 512\frac{W_p}{W_\Sigma - W_p}R_Y = R_{Y_P} = 512\frac{8}{56}R_Y = R_{Y_P} = 512\frac{1}{7}R_Y$$

Computing the rewards

The following calculations are based on 500 thousand active validators, all performing perfectly and all with 32 ETH of effective balance.

Base reward per increment

$$b = \frac{\text{EFFECTIVE\_BALANCE\_INCREMENT x BASE\_REWARD\_FACTOR}}{\sqrt{\text{TOTAL ACTIVE BALANCE}}} = \frac{1{,}000{,}000{,}000 \times 64}{\sqrt{32{,}000{,}000{,}000 \times 500{,}000}} = \mathbf{505}\ \textit{Gwei}$$

Value of a single attestation

$$R_A = \frac{W_s + W_t + W_h}{W_\Sigma} nb = \frac{14+26+14}{64} 32b = \mathbf{13{,}635}\ \text{Gwei}$$

Value of a single sync committee contribution

$$R_Y = \frac{W_y}{32 \times 512 \times W_\Sigma} Tb = \frac{2}{32 \times 512 \times 64} 500{,}000 \times 32b = \mathbf{15{,}411}\ \text{Gwei}$$

Value of a block proposal due to attestations

$$R_{A_P} = \frac{W_p}{W_\Sigma - W_p} R = \frac{500{,}000}{32} \frac{8}{64-8} R_A = \mathbf{30{,}435{,}267}\ \text{Gwei}$$

We have to divide by 32 because we are computing for a single block proposal

Computing the rewards

Value of a block proposal due to sync committee contributions

$$R_{Y_P} = 512\frac{W_p}{W_\Sigma - W_p}R_Y = 512\frac{8}{64-8}R_Y = 1{,}127{,}204 \text{ Gwei}$$

Putting it all together, the total available reward per epoch across all validators is $500{,}000R_A + 32\left(512R_Y + R_{A_P} + R_{Y_P}\right) = 8{,}080{,}000{,}000$ Gwei

Finally, as a check-sum, $Tb = 500{,}000 \times 32b = 8{,}080{,}000{,}000\ Gwei = 8.080\ ETH$ issued per epoch.

Penalties

Attestations

Attestations are penalized for being **missing, late, or incorrect**.

Penalties are subtracted from validators' balances on the beacon chain and effectively burned, so they reduce the net issuance of the chain.

Attesters are penalized for missed Casper FFG votes, that is, missed source or target votes. But there is no penalty for a missed head vote.

Timeliness	1 slot	<= 5 slots	<= 32 slots	> 32 Slots (missing)
Wrong source	$-W_s - W_t$	$-W_s - W_t$	$-W_s - W_t$	$-W_s - W_t$
Correct source only	$W_s - W_t$	$W_s - W_t$	$-W_s - W_t$	$-W_s - W_t$
Correct source and target only	$W_s + W_t$	$W_s + W_t$	$-W_s + W_t$	$-W_s - W_t$
Correct source, target and head	$W_s + W_t + W_h$	$W_s + W_t$	$-W_s + W_t$	$-W_s - W_t$

we need then to multiply by $\frac{nb}{W_\Sigma}$ to get the actual reward.

Penalties

Block proposer

There are no explicit penalties related to block proposers.

Sync committee

Validators that don't participate (sign the wrong head block or don't show up at all) receive a penalty exactly equal to the reward they would have earned for being correct. And the block proposer receives nothing for the missing contribution.

Inactivity leak

The problem

If the chain hasn't finalized a checkpoint for longer than **4** epochs, then it enters "inactivity leak" mode.

The reason

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 solution

The inactivity leak is a kind of emergency state in which rewards and penalties are modified, with the ultimate aim to create the conditions required for the chain to recover finality.

The idea

When loss of finality is detected the inactivity leak gradually reduces the stakes of validators who are not making attestations until, eventually, the participating validators control 2/3 of the remaining stake. They can then begin to finalize checkpoints once again.

Inactivity leak

Modification to the rewards

- Attesters receive no attestation rewards while attestation penalties are unchanged.
- Any validators deemed inactive have their inactivity scores raised, leading to an additional inactivity penalty that potentially grows quadratically with time.
- Proposer and sync committee rewards are unchanged.

Why validator do not receive attestation rewards

The reason is due to the possibility of ***discouragement attacks***.

An attacker might deliberately drive the chain into an inactivity leak, perhaps by a combination of censorship and denial of service attack on other validators.

This would cause the non-participants to suffer the leak, while the attacker continues to attest normally. Thus the idea is to increase the cost to the attacker in this scenario by not rewarding attestations at all during an inactivity leak.

Inactivity leak

Inactivity score

It is an individual score of the validator, and it is updated each epoch as follows:

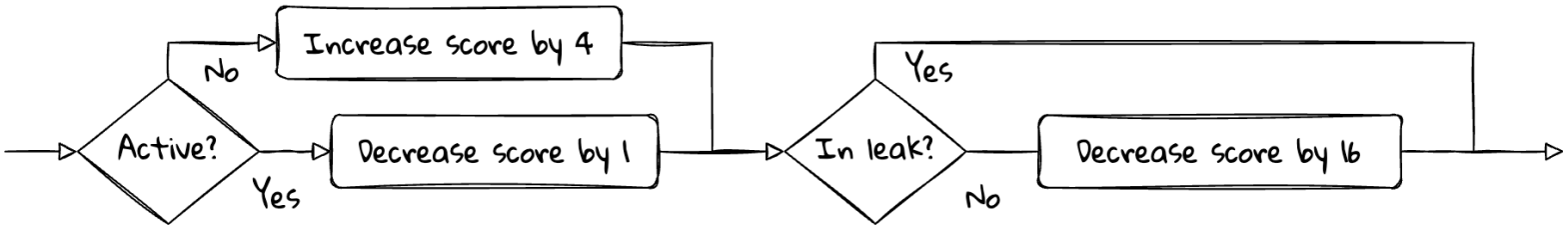
At the end of epoch N, irrespective of the inactivity leak,

- decrease a validator's score by one when it made a correct and timely target vote in epoch $N-1$, and
- increase the validator's score by INACTIVITY_SCORE_BIAS (4) otherwise.

When ***not*** in an inactivity leak,

- decrease every validator's score by INACTIVITY_SCORE_RECOVERY_RATE (16)

Validator score can not be lower than 0.



Inactivity leak

Inactivity penalties

The inactivity penalty is applied to all validators at every epoch based on their individual inactivity scores, ***irrespective of whether a leak is in progress or not***.

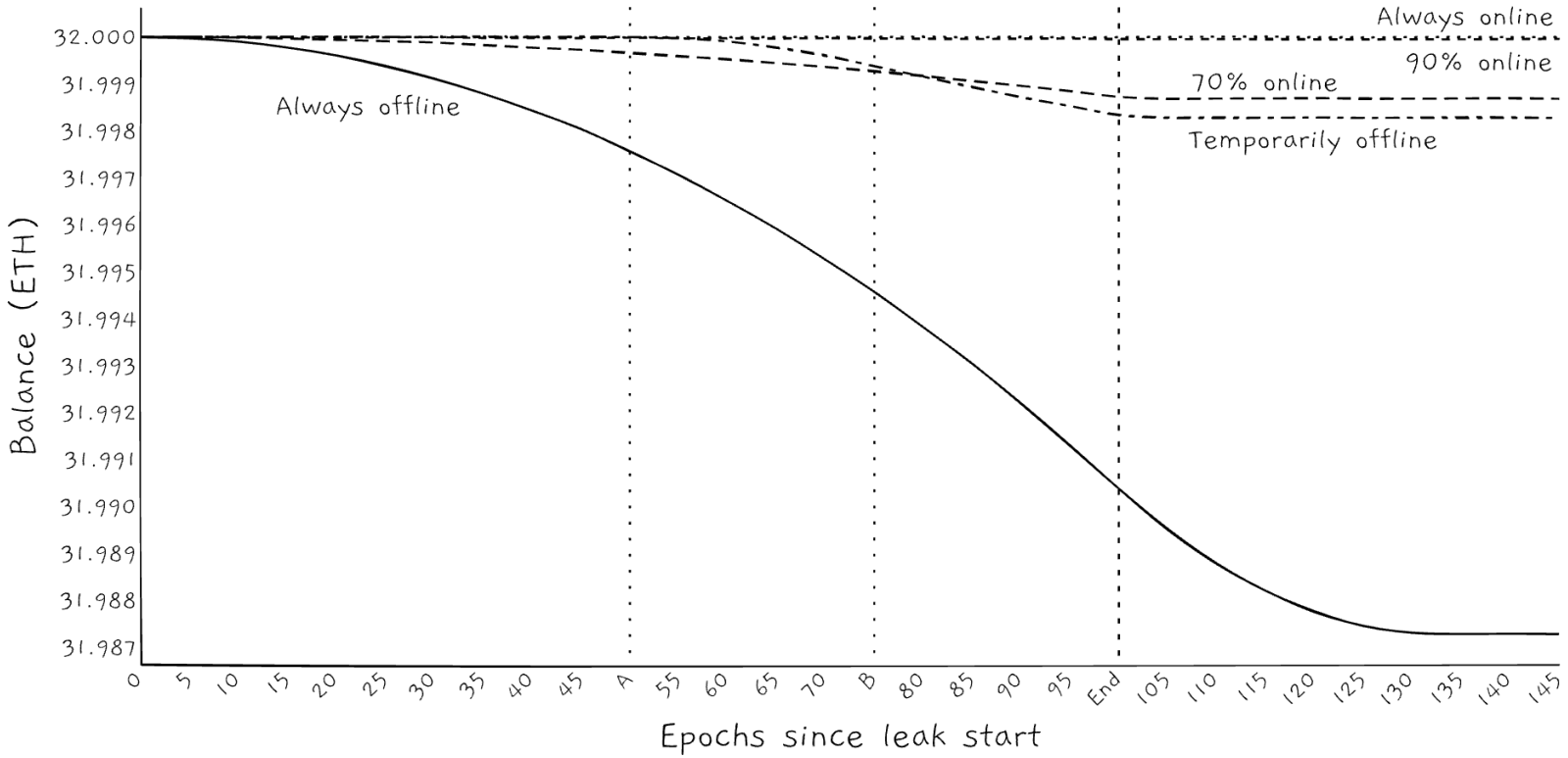
The penalty for validator i is calculated as

$$\frac{s_i B_i}{(\texttt{INACTIVITY\_SCORE\_BIAS} \times \texttt{INACTIVITY\_PENALTY\_QUOTIENT\_BELLATRIX})} = \frac{s_i B_i}{4 \times 16777216}$$

Where:

- s_i is the validator's inactivity score
- B_i is the validator's effective balance
- `INACTIVITY_SCORE_BIAS` is a constant valued as 4
- `INACTIVITY_PENALTY_QUOTIENT_BELLATRIX` is a constant valued as 16777216

Inactivity leak



Inactivity leak

Validators ejection

A validator will be exited when its effective balance drops to EJECTION_BALANCE (16 ETH).

If a validator's effective balance falls to 16 Ether or below then it is exited from the system.

Due to the way that effective balance is calculated, the ejection will happen when the actual balance drops below 16.75 ETH.

Note that: it is not necessary for non-participating validators to be ejected from the active validator set in order for the inactivity leak to be effective at regaining finality. Reducing the proportion of the total stake held by those non-participating validators is sufficient.

Slashing

Slashing occurs when validators make attestations or block proposals that break very specific protocol rules.

It applies to behavior that could **potentially** be part of an **attack** on the chain. So it is a punishment.

Getting slashed means losing a significant amount of stake and being ejected from the protocol.

The behaviors that lead to slashing are as follows:

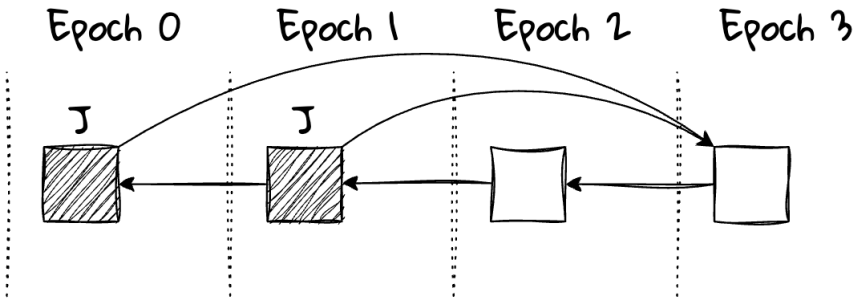
- **Double vote**: making two differing attestations for the same target checkpoint (Casper FFG)
- **Surrounding vote**: making an attestation whose source and target votes "surround" those in another attestation from the same validator (Casper FFG).
- **Double block**: proposing more than one distinct block at the same height (Casper FFG)
- **Double head attestation**: attesting to different head blocks, with the same source and target checkpoints (Casper FFG)

.

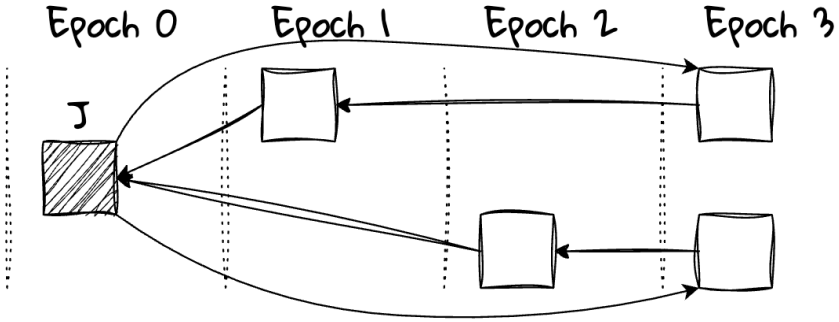
Double vote

Considering the following notation: if c is a checkpoint, then $h(c)$ is the height of that checkpoint.
A validator must not publish distinct votes $s_1 \rightarrow t_1$ and $s_2 \rightarrow t_2$ such that $h(t_1) = h(t_2)$.

i.e.: a validator must make at most one vote for any target epoch.



One way to violate the no double vote rule: voting for the same target checkpoint from different source checkpoints: $0 \rightarrow 3$ *and* $1 \rightarrow 3$.



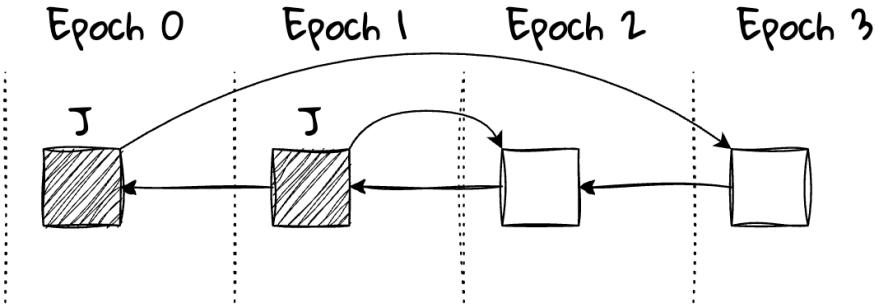
Another way to violate the no double vote rule: voting for different target checkpoints in the same epoch: $0 \rightarrow 3$ *and* $0 \rightarrow 3'$.

Surrounding vote

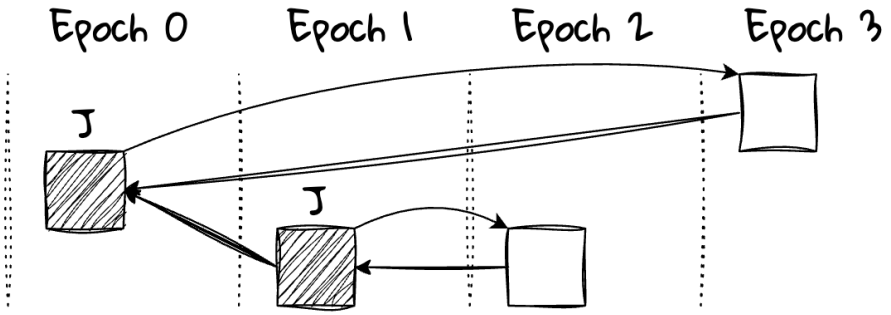
Considering the following notation: if c is a checkpoint, then $h(c)$ is the height of that checkpoint.

A validator must not publish distinct votes $s_1 \rightarrow t_1$ and $s_2 \rightarrow t_2$ such that $h(s_1) < h(s_2) < h(t_2) < h(t_1)$.

i.e.: a validator must not make a vote such that its link either surrounds, or is surrounded by, a previous link it voted for.



One way to violate the no surround vote rule: the link $0 \rightarrow 3$ *surrounds the link* $1 \rightarrow 2$.

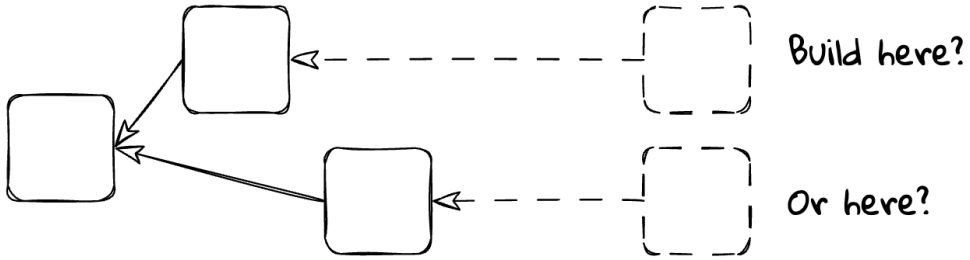


Another way to violate the no surround vote rule: again, the link $0 \rightarrow 3$ *surrounds the link* $1 \rightarrow 2$, *albeit on different branches.*

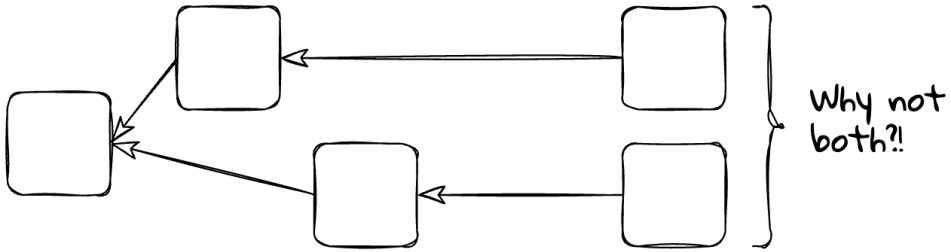
Double block and double attestation

it is almost costless for validators to produce blocks. However, when a proposer have to produce a block have to choice the right branch or fork on where to publish it.

Making the wrong choice could lead to not be part of the canonical chain and thus not earn the rewards.



Therefore, a good strategy would seem to be to propose multiple blocks, one built on each possible head, then at least one of my blocks is guaranteed to become part of the eventual canonical chain.



This is undesirable because it prolongs a fork and prevents the network from converging on a single canonical history. For these reasons, such behavior is punished and lead to slashing.

A similar issue arises for attesters, who might vote for multiple heads to avoid choosing the wrong fork.

Slashing – The punishment

Slashing is triggered by the evidence of the offence being included in a beacon chain block. Once the evidence is confirmed by the network, the offending validator (or validators) is slashed.

The offender immediately has $\frac{1}{32}$ of its effective balance deducted from its actual balance. This is a maximum of 1 ETH due to the cap on effective balance. Then a 36 day removal period begins.

At the halfway point of this period (18 days after being slashed) the slashed validator is due to receive a second penalty.

This second penalty is based on the total amount of stake slashed during the 18 days before and after our validator was slashed.

The idea is to scale the punishment so that a one-off event posing little threat to the chain is only lightly punished, while a mass slashing event that might be the result of an attempt to finalise conflicting blocks is punished to the maximum extent possible.

A validator that is slashed continues to receive attestation penalties until its withdrawable epoch, which is set to 8192 epochs (36 days) after the slashing, and they are unable to receive any attestation rewards during this time. They are also subject for this entire period to any inactivity leak that might be in operation. Whatever the slashed validator does, it is penalised exactly as if it is failing to participate.

An interesting edge case is that slashed validators are eligible to be selected for sync committee duty and to receive the rewards for sync committee participation.

The whistleblower

In order for the chain to verify slashings and take action against the offender, the evidence needs to be included in a beacon block. To incentivize validators to make the effort there is a specific reward for the proposer of a block that includes slashings.

The idea is to incentivize nodes that search for and discover evidence of slashable behavior, which can be an intensive process.

The reward amount to $\frac{B}{512}$ where B is the effective balance of the validator being slashed.

The whistleblower receives $\frac{7}{8}$ of the above reward and the proposer $\frac{1}{8}$.

Validator lifecycle

A node staking 32 ETH into a deposit contract on Ethereum mainnet, will activate one validator.

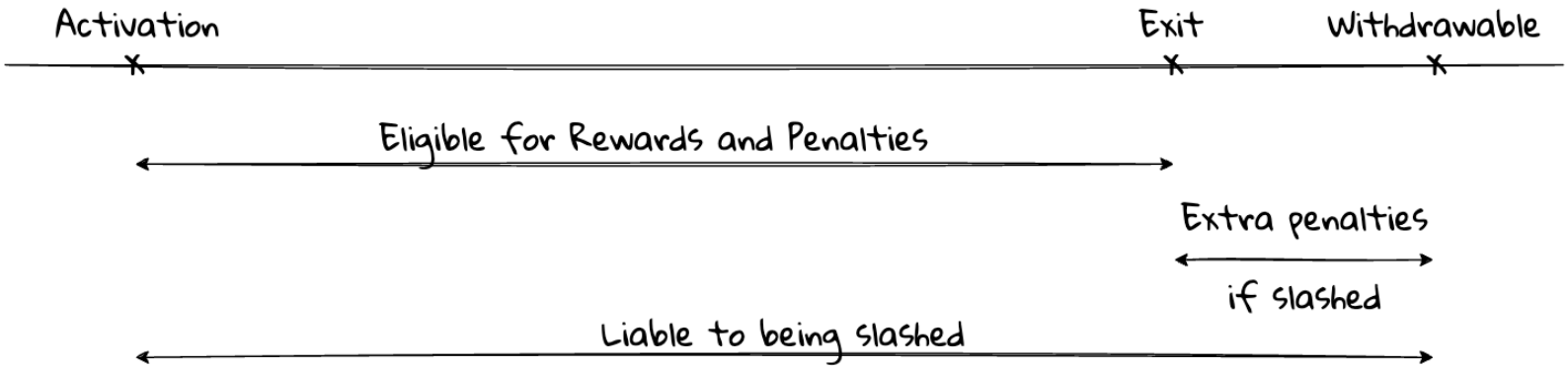
The Chain deactivates (“forced exit”) all validators whose effective balance reaches 16 ETH.

Validators can also “voluntary exit” after serving for at least 2,048 epochs, around 9 days.

In any voluntary or forced exit, there is a delay of 4 epochs before stakers can withdraw their stake.

Within the 4 epochs, a validator can still be caught and slashed.
An honest validator’s balance is withdrawable in around 27 hours.

But a slashed validator incurs a delay of 8,192 epochs (approximately 36 days).



Timeline of the eligibility of validators for rewards

Validator lifecycle