Increase the MAX_EFFECTIVE_BALANCE

– a modest proposal*

NOTE: This proposal **does not** increase the minimum of 32 ETH to become a validator.

by mike neuder, francesco d'amato, aditya asgaonkar, and justin drake – june 6, 2023

~ accompanying artifacts ~

[a] the diff view of a minimal consensus specpull request

[b] security considerations + annotated specdoc

[c] the full consensus pyspec/spec testspull request

Proposal

– Increase the MAX_EFFECTIVE_BALANCE

to encourage validator set contraction, which

1. unblocks single-slot finality and enshrined PBS, and
2. reduces unnecessary strain on the p2p layer.

Critically, we do not propose

1. increasing the 32 ETH minimum

required to become a validator, or

1. requiring any sort of validator consolidation (this process would be purely opt-in).

tl;dr; [MAX_EFFECTIVE_BALANCE

](https://github.com/ethereum/consensus-specs/blob/9c35b7384e78da643f51f9936c578da7d04db698/specs/phase0/beacon-chain.md#gwei-values) (abbr. MaxEB

) caps the effective balance of Ethereum validators at 32 ETH

. This cap results in a very large validator set; as of June 6, 2023, there are over600,000 active validators with an additional 90,000 in the activation queue. While having many validators signals decentralization, the MaxEB

artificially inflates the validator set size by forcing large staking operations to run thousands of validators. We argue that increasing the MaxEB

(i) unblocks future consensus layer upgrades on theroadmap, (ii) improves the performance of the current consensus mechanism and p2p layer, and (iii) enhances operational efficiency for both small and large-scale validators.

Many thanks to Caspar, Chris, Terence, Dan Marzec, Anders, Tim, Danny, Jim, and Rajiv for comments on draft versions of this document.

Effective balances and MAX_EFFECTIVE_BALANCE

Effective balance

is a field in the validator struct calculated using the amount of ETH

staked by each validator. This value is used for a number of consensus layer operations, including

1. checking if a validator is eligible for the activation queue,
2. calculating the slashing penalties and whistleblower rewards,
3. evaluating the attestation weight used for the fork-choice rule and the justification & finalization of epochs,
4. determining if a validator is selected as a proposer,
5. deciding if a validator is part of the next sync committee, etc…

Effective balance is calculated in increments of 10^9

gwei

(1

ETH

– the [EFFECTIVE_BALANCE_INCREMENT

](https://github.com/ethereum/consensus-specs/blob/9c35b7384e78da643f51f9936c578da7d04db698/specs/phase0/beacon-chain.md#gwei-values)) and is updated in [process_effective_balance_updates

](https://www.attestant.io/posts/understanding-validator-effective-balance/). The update rule behaves like a modified floor function with hysteresis zones determining when a balance changes. See [“Understanding validator effective balance”

](https://www.attestant.io/posts/understanding-validator-effective-balance/) for more details.

The MAX_EFFECTIVE_BALANCE

is a spec-defined constant of 32 \times 10^9

gwei

(32

ETH

), which sets a hard cap on the effective balance of any individual validator. Post-capella, validator balances are automatically withdrawn. As defined in the spec, exited validators have their full balance withdrawn and active validators with a balance exceeding the MaxEB

are partially withdrawn.

Why we should

increase it

There are many inefficiencies resulting from the MaxEB

being low. We analyze the benefits of increasing it from the perspective of (i) future roadmap upgrades, (ii) the current consensus and p2p layers, and (iii) the validators.

Without a validator set contraction, single-slot finality is not feasible using the current designs. Without single-slot finality, we believe that enshrined PBS is also not viable. Additionally, the current p2p layer is heavily burdened by the artificially large and rapidly growing validator set (see this thread from Potuz outlining what happened during the May 12 non-finality event). ~~ We see a validator set contraction as a must-have for a sustainable and upgradable Ethereum consensus layer. ~~

The roadmap perspective

As outlined in Vitalik’s roadmap, there are still ambitious goals around improving the consensus layer. We present two upgrades that are infeasible given the size of the validator set, but are unblocked by increasing the MaxEB

.

1. single-slot finality

– SSF has long been researched and is a critical component of the end-game vision for Ethereum Proof-of-Stake. Horn is the state-of-the-art BLS signature aggregation proposal. From the post, a validator set with 1 million participants results in the worst-case signature aggregation taking 2.8s

on a top-tier 2021 CPU and 6.1s

on an older machine. While there may be more improvements in the aggregation scheme and the hardware, this performance is prohibitively slow in the near-term given a validator set of this size. By compressing the validator set, we can begin working towards single-slot finality immediately.

1. ePBS

– Enshrined Proposer-Builder Separation has also been discussed for multiple years. Due to security concerns around ex-ante/ex-post reorgs and balancing attacks, proposer boost was implemented as a stop-gap measure to protect HLMD-GHOST (Hybrid Latest Message Driven-GHOST). If we were to implement ePBS today, the security benefits from proposer

boost (or even the superior view-merge) are reduced. The high-level reasoning here is that the security properties of HLMD-GHOST rely heavily on honest proposers. With every other block being a “builder block”, the action space of a malicious proposer increases significantly (e.g., they may execute length-2k

ex-post reorgs with the probability equal to the length-k

ex-post reorgs under today’s mechanism). We plan on writing further on this topic in the coming weeks. With a smaller validator set, we can implement new mechanisms like SSF, which have stronger security properties. With a stronger consensus layer we can proceed to the implementation of ePBS (and even mev-burn) with much improved confidence around the security of the overall protocol.

The current consensus layer perspective

The consensus nodes are under a large load to handle the scale of the current validator set. On May 11th & 12th, 2023, the beacon chain experienced two multi-epoch delays in finalization. Part of the suspected root cause is high-resource utilization on the beacon nodes caused by the increase in the validator set and significant deposit inflows during each epcoh. We present two areas of today’s protocol that benefit from increasing the MaxEB

.

1. p2p layer

– To support Gasper, the full validator set is partitioned into 32 attesting committees (each attesting committee is split into 64 sub-committees used for attestation aggregation, but these sub-committees do not need to be majority honest for the security of the protocol and are mostly a historical artifact of the original sharding design which has been abandoned); each committee attests during one of the slots in an epoch. Each attestation requires a signature verification and must be aggregated. Many of these checks are redundant as they come from different validators running on the same machine and controlled by the same node operator. Any reduction of the validator set directly reduces the computational cost of processing the attestations over the duration of an epoch. See Aditya’s [“Removing unnecessary stress from Ethereum’s P2P layer”

](https://ethresear.ch/t/removing-unnecessary-stress-from-ethereums-p2p-network/15547) for more context.

1. processing of auto-withdrawals

– Since withdrawals of all balances over the MaxEB

are done automatically, there is a large withdrawal load incurred every epoch. By increasing the MaxEB

, validators can choose to leave their stake in the protocol to earn compounding rewards. Based on data from mated.network the average withdrawal queue length is about 6 days (this may quickly increase to ~10 days as more withdrawal credentials are updated and the validator set continues to grow). The vast majority of this queue is partial withdrawals from the active validator [sweep

](https://github.com/ethereum/consensus-specs/blob/468b5be7b8a516d6fbc60ad8abec8830a5181bf4/specs/capella/beacon-chain.md#introduction) (see [get_expected_withdrawals

](https://github.com/ethereum/consensus-specs/blob/468b5be7b8a516d6fbc60ad8abec8830a5181bf4/specs/capella/beacon-chain.md#new-get_expected_withdrawals)). A validator set contraction would directly reduce the withdrawal queue length.

The validator perspective

We focus on two pros of increasing the MaxEB

:

1. Democratization of compounding stake

(benefits solo-stakers)

– Currently, any stake above the MaxEB

is not earning staking rewards. Staking pools can use withdrawals to compound their staking balance very quickly because they coalesce their rewards over many validators to create 32 ETH

chunks needed to instantiate a new validator. With the current APR of~6%, a single validator will earn roughly 0.016\%

daily. At this rate, it will take a solo-staker over 11 years to earn a full 32 ETH

for a fresh validator. Coinbase on the other hand, will earn 32 \cdot 0.00016 \cdot 60000 \approx 307

new ETH

every day, which is enough to spin up 9 additional validators. By increasing the MaxEB

, validator's of any size can opt-in to compounding rewards.

1. Reducing the operational overhead of managing many validators

(benefits large-scale stakers)

– With the MaxEB

being so low, staking pools are required to manage thousands of validators. Frommevboost.pics, the top 3 validators by block-share are responsible for over 225,000 validators (though Lido is a conglomerate of 29 operators, each still represents a significant portion of the validator set).

1. Lido 161,989 (28.46%)
2. Coinbase 59,246 (10.41%)
3. Kraken 27,229 (4.78%)
4. This means 225,000 unique key-pairs, managing the signatures for each validator's attestations and proposals, and running multiple beacon nodes each with many validators. While we can assume large pools would still distribute their stake over many validators for redundancy and safety, increasing the MaxEB

would allow them the flexibility consolidate their balances rather than being arbitrarily capped at 32 ETH

.

Note: Reducing staking pool operational overhead could also be seen as a negative externality of this proposal. However, we believe that the protocol and solo-staker improvements are more significant than the benefits to the large staking pools.

Why we shouldn't

increase it

While we think the benefits of increasing the MaxEB

far outweigh the costs, we present two counter-arguments.

1. simplicity of the current implementation

– By ensuring the effective validator balances are constrained to the range [16,32] ETH

(16 ETH

is the ejection balance; effective balance could

drop slightly below 16 ETH

because of the exit queue duration, but 16 ETH

is the approximate lower bound), it is easy to reason about attestation weights, sync committee selection, and the random assignment of proposers. The protocol is already implemented, so the R&D costs incurred by changing the mechanism take away focus from other protocol efforts. * Response –

The spec change we are proposing is quite minimal. We analyze these changes in ["Security Considerations and Spec Changes for a MAX_EFFECTIVE_BALANCE Increase"

](https://notes.ethereum.org/@fradamt/meb-increase-security). We believe that the current size and growth of the validator set are unsustainable, justifying this change.

1. Response –

The spec change we are proposing is quite minimal. We analyze these changes in ["Security Considerations and Spec Changes for a MAX_EFFECTIVE_BALANCE Increase"

](https://notes.ethereum.org/@fradamt/meb-increase-security). We believe that the current size and growth of the validator set are unsustainable, justifying this change.

1. considerations around committees

– Preliminary note: In SSF, committees are no longer part of the fork-choice rule, and thus these concerns will be irrelevant.

Given validators have differing stake, partitioning them into committees may result in some validators having much larger impact over the committee than others. Additionally, by reducing the validator set size, we decrease the safety margin of random sampling. For sync committees, this is not an issue because sampling the participants is done with replacement and proportional to effective balance. Once a sync committee is selected each validator receives one vote. For attesting committees, some validators will have much more voting power by having a larger effective balance. Additionally, with less validators, there is a higher probability that an attacker could own the majority of the weight of an attesting committee. However, with a sufficiently large sample, we argue that the safety margins are not being reduced enough to warrent concern. For example, if the validator set is reduced by 4x, we need 55% honest validators to acheive safety (honest majority) of a single slot with probability 1-10^{11}

. With today's validator set size, we need 53% honest validators to achieve the same safety margin; a 4x reduction in the validator set size only increases honest validator threshold by 2%. * Response –

See committee analysis in ["Security of Committees"

](https://notes.ethereum.org/nHqON5l7SACkL_nPwz8Vqw?both#Security-of-committees). We believe this change adequetely addresses concerns around committees, and that even a validator set contraction is a safe and necessary step.

1. Response –

See committee analysis in ["Security of Committees"

](https://notes.ethereum.org/nHqON5l7SACkL_nPwz8Vqw?both#Security-of-committees). We believe this change adequetely addresses concerns around committees, and that even a validator set contraction is a safe and necessary step.

Mechanisms already in place

Attesting validator weight, proposer selection probability, and weight-based sampling with replacement for sync committees are already proportional to the effective balance of a validator. These three key components work without modification with a higher MaxEB

.

1. attesting validator weight

– validators with higher effective balances are already weighted higher in the fork-choice rule. See [get_attesting_balance

](https://github.com/ethereum/consensus-specs/blob/9c35b7384e78da643f51f9936c578da7d04db698/specs/phase0/beacon-chain.md#helper-functions-1). This will accurately weight higher-stake validators as having more influence over the canonical chain (as desired). We provide an analysis of the probability of a maliciously controlled attesting committee in ["Security of Committees"

](https://notes.ethereum.org/nHqON5l7SACkL_nPwz8Vqw?both#Security-of-committees).

1. proposer selection probability

– We already weight the probability of becoming a proposer by the effective balance of that validator. See [compute_proposer_index

](https://github.com/ethereum/consensus-specs/blob/9c35b7384e78da643f51f9936c578da7d04db698/specs/phase0/beacon-chain.md#compute_proposer_index). Currently, if a validator's effective balance (EB

) is below the MaxEB

, they are selected as the proposer given their validator index was randomly chosen only if,

$EB \cdot 255 \geq MaxEB \cdot r, \; \text{where} \; r \sim U(0, 255)$.

Thus if EB = MaxEB

, given the validator index was selected, the validator becomes the proposer with probability 1. Otherwise the probability is

$Pr(proposer \mid selected) = Pr\left(r \leq \frac{255 \cdot EB}{MaxEB}\right)$

This works as intended even with a higher MaxEB

, though it will slightly increase the time it takes to calculate the next proposer index (lower values of EB

will result in lower probability of selection, thus more iterations of the loop).

1. sync committee selection

– Sampling the validators to select the sync committee is already done with replacement (see [get_next_sync_committee_indices

](https://github.com/ethereum/consensus-specs/blob/468b5be7b8a516d6fbc60ad8abec8830a5181bf4/specs/altair/beacon-chain.md#get_next_sync_committee_indices)). Additionally, each validator selected for the committee has a single vote. This logic works as intended even with a large increase to the MaxEB

.