

Removing Unnecessary Stress from Ethereum's P2P Network

Ethereum is currently processing 2x the number of messages than are required. The root cause of this unnecessary stress on the network is the mismatch between the number of validators and the number of distinct participants (i.e., staking entities) in the protocol. The network is working overtime to aggregate messages from multiple validators of the same staking entity!

We should remove this unnecessary stress from Ethereum's p2p network by allowing large staking entities to consolidate their stake into fewer validators.

Author's Note:

There are other reasons to desire a reduction in the validator set size, such as [single-slot finality](#). I write this post with a singular objective - to reduce unnecessary p2p messages - because it's an important maintenance fix irrespective of other future protocol upgrades such as single-slot finality.

tl;dr

– steps to reduce unnecessary stress from Ethereum's network:

- Investigate the risks of having large variance in validator weights
- [consensus-specs

](<https://github.com/ethereum/consensus-specs>) changes: * Increase MAX_EFFECTIVE_BALANCE

- Provide one-step method for stake consolidation (i.e., validator exit & balance transfer into another validator)
- Update the withdrawal mechanism to support partial withdrawals when balance is below MAX_EFFECTIVE_BALANCE
- Increase MAX_EFFECTIVE_BALANCE
- Provide one-step method for stake consolidation (i.e., validator exit & balance transfer into another validator)
- Update the withdrawal mechanism to support partial withdrawals when balance is below MAX_EFFECTIVE_BALANCE
- Build [DVT](#) to provide resilient staking infrastructure

Problem

Let's better understand the problem with an example:

[

image

858×995 63.7 KB

](<https://ethresear.ch/uploads/default/original/2X/c/cbd3ad04bff02b4c40169724aef85508f962f0af.png>)

Each validator in the above figure is controlled by a distinct staker. The validators send their individual attestations into the Ethereum network for aggregation. Overall, the network processes 5 messages to account for the participation of 5 stakers in the protocol.

The problem appears when a large staker controls multiple validators:

[

image

882×983 57.9 KB

](<https://ethresear.ch/uploads/default/original/2X/d/d8465ea65ff6562766b9d2b77c1908aa9efdb611.png>)

The network is now processing 3 messages on behalf of the large staker. As compared to a staker with a single validator, the network bears a 3x cost to account for the large staker's participation in the protocol.

Now, let's look at the situation on mainnet Ethereum:

[

image

713×615 68.8 KB

](https://ethresear.ch/uploads/default/original/2X/7/7fd26820f2573b8a770d52f905bd2dcc9e02ea86.png)

About 50% of the current 560,000 validators are controlled by 10 entities [source: beaconcha.in]

. Half of all messages in the network are produced by just a few entities - meaning that we are processing 2x the number of messages than are required!

Another perspective on the unnecessary cost the network is bearing: the network spends half of its aggregation efforts towards attestations produced by just a few participants. If you run an Ethereum validator, half your bandwidth is consumed in aggregating the attestations produced by just a few participants.

The obvious next questions – Why do large stakers need to operate so many validators? Why can't they make do with fewer validators?

MAXIMUM_EFFECTIVE_BALANCE

The [effective balance

](https://github.com/ethereum/consensus-specs/blob/01b53691dcc36d37a5ad8994b3a32d8de69fb1aa/specs/phase0/beacon-chain.md#validator) of a validator is the amount of stake that counts towards the validator's weight in the PoS protocol.

[MAXIMUM_EFFECTIVE_BALANCE

](https://github.com/ethereum/consensus-specs/blob/01b53691dcc36d37a5ad8994b3a32d8de69fb1aa/specs/phase0/beacon-chain.md#gwei-values) is the maximum effective balance that a validator can have. This parameter is currently set at 32 ETH

. If a validator has a balance of more than MAXIMUM_EFFECTIVE_BALANCE

, the excess is not considered towards the validator's weight in the PoS protocol.

PoS protocol rewards are proportional to the validator's weight, which is capped at the MAXIMUM_EFFECTIVE_BALANCE

, so a staker with more than 32 ETH is forced to create multiple validators to gain the maximum possible rewards.

This [protocol design decision](#) was made in preparation for shard committees (a feature that is now obsolete) and assuming that we have an entire epoch for the hearing from the entire validator set. Since then, Ethereum has adopted a rollup-centric roadmap, which [does not require](#) this constraint!

Solution: Increase MAXIMUM_EFFECTIVE_BALANCE

[

image

785×969 53.1 KB

](https://ethresear.ch/uploads/default/original/2X/3/3da85b880ae0f37f7f17e937de51b2bb498d47c1.png)

Increasing MAX_EFFECTIVE_BALANCE

would allow these stakers to consolidate their capital into far fewer validators, thus reducing the validator set size & number of messages. Today, this would amount to a 50% reduction in the number of validators & messages!

Sampling Proposers & Committees

The beacon chain picks [proposers](#) & [committees](#) by random sampling of the validator set.

The sampling for proposers is weighted by the effective balance, so no change is required in this process.

However, the sampling for committees is not weighted by the effective balance. Increasing MAXIMUM_EFFECTIVE_BALANCE

would allow for large differences between the total weight of committees. An open research question is whether this presents any security risks, such as an increased possibility of reorgs. If so, we would need to change to a committee sampling mechanism that ensures roughly the same weight for each committee.

Validator Exit & Transfer of Stake

Currently, the only way to consolidate stake from multiple validators into a single one is to withdraw the stake to the Execution Layer (EL) & then top-up the balance of the single validator.

To streamline this process, it is useful to add a Consensus Layer (CL) feature for exiting a validator & transferring the entire stake to another validator. This would prevent the overhead of a CL-to-EL withdrawal and make it easier to convince large stakers to consolidate their stake in fewer validators.

Partial Withdrawal Mechanism

The current [partial withdrawal mechanism](#) allows validators to withdraw a part of their balance without exiting their validator. However, only the balance in excess of the MAX_EFFECTIVE_BALANCE

is available for partial withdrawal.

If the MAX_EFFECTIVE_BALANCE

is increased significantly, we need to support the use case of partial withdrawal when the validator's balance is lower than the MAX_EFFECTIVE_BALANCE

.

Resilience in Staking Infrastructure

A natural concern when suggesting that a large staker operate just a single validator is the reduction in the resilience of their staking setup. Currently, large stakers have their stake split into multiple validators running on independent machines (I hope!). By consolidating their stake into a single validator running on one machine, they would introduce a single point of failure in their staking infrastructure. An awesome solution to this problem is [Distributed Validator Technology \(DVT\)](#), which introduces resilience by allowing a single validator to be run from a cluster of machines.