

TL; DR

This post describes the current background behind validator exiting order design. We, Lido dev team, want to provoke public discussion and gather feedback and stakeholders' sentiment; especially looking forward to hearing from Node Operators representatives.

The current dev team preference is Approach 1: eject first the validator from a Node Operator with the largest amount of active validators.

Intro

When withdrawals in Ethereum become feasible Lido stakers will be able to withdraw ETH in exchange for their stETH via Lido protocol. To satisfy the withdrawal requests Lido will have to eject validators to get the users' funds back. In what order should Lido choose validators of which Node Operators to eject? The purpose of this post is to initiate a public discussion on the topic.

The question of how the already chosen validator gets ejected technically is an important and non-trivial one. Especially while withdrawal credentials initiated exits are not implemented in Ethereum and Node Operators are in charge of signing the exit messages. But this question is proposed to be left outside of the scope of this discussion.

There are multiple possible considerations regarding ejection ordering.

- must be aligned with Lido [goals](#) / [scorecard](#)
- provide Node Operators with equal opportunities to earn with Lido (don't favor or disadvantage any Node Operators in some unclear, "unfair" manner)
- take into account the current validator distribution
- be consistent with [the choice of the next Node Operator whose validator to activate](#)
- be technically feasible/reasonable from the point of view of the dev team

Let's consider them one by one, explore potential criteria for ejection ordering and consider a few approaches.

Lido goals

Lido has published [Decentralization Roadmap](#) where the preferred properties of the validator set are specified (also see the [scorecard](#)).

Validator ejection ordering algorithm might be considered as means of helping to achieve these properties. Here are some of them:

- no operators with >1% of the total stake
- good performance
- operators earn well enough to build a profitable, dependable business on staking
- an emphasis on improving client diversity in Ethereum
- operations are distributed geographically and jurisdictionally
- distributed variation of on-premise infra and different cloud providers

Provide Node Operators with equal opportunities to earn with Lido

When a Node Operator joins Lido, it has to invest in infrastructure, human resources, etc. And only after it gains the capability to earn. The more active validators it has, the more ROI it can get over time.

From this point of view, it is not preferable to eject validators of the Node Operator who had significantly less capability to earn with Lido.

Current validator distribution

As of Oct 7 2022, the distribution of active validators between the Node Operators looks like this:

index

name
active validators
network share
rewards earned
0

Staking Facilities

7391
1.68%
613.0
2

[P2P.ORG](#) - P2P Validator

7391
1.68%
697.6
4

stakefish

7391
1.68%
676.6
5

Blockscape

7391
1.68%
625.5
6

DSRV

7391
1.68%
490.7
8

SkillZ

7390
1.68%
583.2
9

RockX

7390

1.68%

383.5

11

Allnodes

7390

1.68%

388.4

15

ChainLayer

7390

1.68%

308.0

17

BridgeTower

7390

1.68%

294.0

3

Chorus One

7000

1.59%

643.9

10

Figment

7000

1.59%

346.1

16

Simply Staking

6500

1.48%

264.2

7

Everstake

6000

1.37%

435.2

19

InfStones

6000

1.37%

224.0

14

Stakin

5664

1.29%

204.2

18

Stakely

5000

1.14%

199.2

20

HashQuark

3768

0.86%

132.6

13

Blockdaemon*

3346

0.76%

193.2

12

Anyblock Analytics*

2300

0.52%

164.0

21

ConsenSys Codefi

1447

0.33%

50.7

1

Certus One

1000
0.23%
153.3
23
CryptoManufaktur
674
0.15%
4.2
24
Kukis Global
673
0.15%
4.3
26
ChainSafe
673
0.15%
3.5
28
Sigma Prime
673
0.15%
3.7
22
RockLogic GmbH
633
0.14%
4.0
25
Nethermind
633
0.14%
3.9
27
Prysmatic Labs
100
0.02%

- Blockdaemon and Anyblock Analytics are [the same entity](#).

Column index

denotes the internal Node Operator index in the Node Operators registry contract. It starts from 0 and gets incremented each time a new Node Operator is onboarded.

The data sources: [Node Operators registry contract](#), [Dune request](#).

Relation with the activation algorithm

The algorithm which chooses new validators for activation is implemented in function [assignNextSigningKeys

](<https://github.com/lidofinance/lido-dao/blob/master/contracts/0.4.24/nos/NodeOperatorsRegistry.sol#L330>) of the NodeOperatorsRegistry contract. The algorithm chooses the next validator from the Node Operator with the least stake (within its limit). Thus the activation algorithm flattens the distribution of validators between Node Operators.

The design of the ejection ordering algorithm should consider how both algorithms working together influence the validator distribution. For example, it might not be suitable to eject just the newly activated validator.

Technical considerations

There are two basic ways the ejection ordering algorithm could be implemented: on-chain and off-chain. The capability to implement it on-chain depends significantly on its simplicity and on-chain availability of the data it needs.

Also, even for the off-chain algorithm, it's preferred to be simple and straightforward for security and reliability. It would also help to make it trustless and ossify the more significant part of the Lido protocol when possible.

Potential criteria for ejection

Here we explore the potential metrics of a Node Operator for use in the ejection ordering algorithm.

Number of the active validators

This property of a Node Operator can be observed directly on-chain by a call to [getNodeOperator

](<https://etherscan.io/address/0x55032650b14df07b85bF18A3a3eC8E0Af2e028d5#readProxyContract#F14>) and calculation of usedSigningKeys - stoppedValidators

.

Amount of the stETH rewards earned so far

This metric can be considered as a measure of the opportunity for a Node Operator to earn with Lido.

This property of a Node Operator can be obtained from a blockchain indexer service. Here is an example of a Dune request: [steth distribution](#)

It seems there is no way to obtain the metric from the current chain state, without accessing archive data.

This metric does not reflect the instant state of the validator distribution. For example, if the ejection ordering algorithm chooses a validator from the Node Operator with the largest amount of rewards today, tomorrow, the amount of rewards won't change much, and the same Node Operator will get chosen. But this metric can potentially be updated to account only the rewards earned by the current active validators to reflect the changes instantly.

Total active validators age

This metric can also be considered as a measure of the opportunity of a Node Operator to earn with Lido. To calculate it for a Node Operator, sum ages of all its active validators (age = time passed since the activation till now). This metric gets updated once the validator stops being active.

Node Operator performance

Monitor penalties and slashings of a Node Operator on the beacon chain. Take it into account in the ejection ordering algorithm as means to disadvantage those who performed worse.

“Soft” properties of validators

Here are some examples of such properties:

- geographic location diversity
- client diversity
- diversity of on-premise infra and cloud providers

The ejection ordering algorithm might take this into account as means of shaping the validator set.

These properties of a validator cannot be trustlessly identified, and thus are not good as ejection criteria.

Potential approaches

Approach 1: eject from the Node Operator with the largest number of active validators

This one is a simple and easy-to-implement algorithm. To choose a validator for ejection, choose a Node Operator with the most active validators count. If there are multiple such NOs, choose the one with the highest validator index to reduce the probability of ejecting the newly added validator, because the activation algorithm chooses the one with the lowest.

To choose a validator of the Node Operator, choose any.

For example, the seven validators ejection Node Operators order looks like this:

DSRV, index 6: 7391 --> 7390 Blockscape, index 5: 7391 --> 7390 stakefish, index 4: 7391 --> 7390 P2P.ORG - P2P
Validator index 2: 7391 --> 7390 Staking Facilities, index 0: 7391 --> 7390 P2P.ORG - P2P Validator index 2: 7390 --> 7389
Staking Facilities, index 0: 7390 --> 7389

Approach 2: eject from the Node Operator with the largest total active validators age

To choose a validator for ejection, select the Node Operator with the largest total validators age.

To calculate the total validators age of a Node Operator, iterate over all its validators and sum time duration since the validator activation till now.

Combined approach

Use different strategies depending on the current state of the validator set. Here is a potential solution, which accelerates movement towards the “less than 1% stake” goal while trying to keep even opportunities to earn with Lido between the largest Node Operator.

Conditions:

1. if there are Node Operators with higher than 1% of the network stake, use strategy A
2. otherwise, strategy B

Strategy A

. Choose the validator for exit among Node Operators with a larger than 1% stake. Choose the Node Operator with the highest total age of the active validators.

Strategy B

. Choose the Node Operator with the largest number of active validators (see Solution 1).