Final Project Report: Validators for Proof-of-Stake Consensus

Diana Fernandes (dianaimf), Longyu Yang (longyuy), Noora Alfayez (nalfayez)

Abstract

This work examines the transition from Proof of Work (PoW) to Proof of Stake (PoS) on Ethereum, referred to as "The Merge," and its impact on validator behaviour. It provides a comprehensive evaluation of how to configure and manage an Ethereum 2 validator node and explores the implications of the new consensus mechanism on the incentives and governance framework of the blockchain. The paper also examines the challenges associated with the decentralized settlement process and analyse the governance structure. Finally, it presents a chain structure and behaviour analysis and results from slashable offences, censorship, and MEV relays.

1 Introduction

The transition from Proof of Work (PoW) to Proof of Stake (PoS) on Ethereum has altered the incentives and governance framework for validators on the blockchain. This change, referred to as "The Merge," has had an impact on performance and modified the actions of validators. To date, there have been limited studies of validator behaviour on the Beacon Chain, and there is not a comprehensive evaluation of how to configure and manage an Ethereum 2 validator node.

2 Background

The *"Merge"* or, the original Ethereum Mainnet merge with a separate proof-of-stake blockchain called (Beacon Chain), occurred on 15 September 2022, where the shift from Proof of Work (PoW) to Proof of Stake (PoS) on the Ethereum Mainnet materialized[1].

The previous consensus mechanics, PoW, faced several criticisms, namely its energy consumption, where this shift aims to reduce Ethereum's energy consumption by 99.95 %, *"making Ethereum a green blockchain."* [2, 3].

The new consensus mechanism changes drastically the incentives for the Ethereum blockchain. More than computing power and energy, costs now rely on validators to propose and agree on new blocks. How incentives (rewards and penalties) are defined, will play an important role. As a poor incentive to perform validation will fail to attract validators and a system that does not penalise infractors will lead to mistrust of the whole network.

This decentralized settlement process, carries several challenges, namely the ones related to moral hazard, adverse selection, exploitation of asymmetric information and other opportunistic behaviours.

The term *"cryptoeconomics"*, is often applied when analysing the governance structure (normative and factual) of crypto-economic systems. Whereas has been defined as *"on individual decision-making and strategic interaction between different participants in a digital ecosystem (e.g. users, providers of key resources, application developers etc.), and uses methodologies from the field of economics - such as game theory, mechanism design and causal inference - to understand how to fund, design, develop, facilitate the operations and encourage the adoption of decentralized marketplaces and related services and digital assets."* [4]

The importance of analysing how these validators will perform is thus crucial to all nodes, to attest to the benefits of the new consensus mechanism, so as its resilience to dishonest and opportunistic behaviours.

3 Structure and Methodology

The present work is organised as follows: Section 1 does a short introduction and Section 2 provides the overall background.

Section 4 is the literature review, section 5 is the "Cognitive Walkthrough on Validator Setup", Section 6 is the "Beacon Chain Structure & Behavior Analysis Methodology".

Results are presented on Section 7 (Slashable Offenses, Censorship and MEV Relays).

Section 8 is the overall discussion and finally, the conclusion (section 9) and limitations and future work (Section 10).

4 Literature Review

4.1 Proof of Work vs Proof of Stake

The original blockchain consensus approach defined by Nakamoto, known as Proof of Work (PoW), raised many challenges with the increase in throughput demands [5]. The industry quickly introduced the alternative Proof-of-Stake mechanism to meet demands and achieve simplicity, robustness, and performance. The key difference between PoW and the emerging PoS approach is the "open enrollment" for miners versus the "permissioned enrollment" through stakes put by validators [6]. There are different deployments of the PoS consensus with various implementations and rules. Some of the notable variations are Delegated Proof of Stake (DPoS), Bonded Proof of Stake (BPoS), and Pure Proof of Stake (PPoS) [5]. Projects that are based on PoS consensus such as Algorand and recently Ethereum have shown how the design can impact performance and affect validators' behavior. In this project, we will analyze validators behavior on the Beacon Chain for Ethereum.

4.2 ETH Validator

A validator is a software client in the Ethereum network participating in receiving new blocks, validating transactions in those new blocks, and reaching a consensus of adding them to the blockchain through voting [7, 8]. To become a validator, one typically needs to stake 32 ETH to the deposit contract which is a gateway for staking on Ethereum. Aside from the amount of ETH, the deposit contract additionally takes as input two public keys (the validator public key and the withdrawal public key) signed by the validator private key so that later the validator can be identified and also approved by the network [9, 10].

There are four ways to become a validator: solo staking, staking-as-a-service, pooled-staking, and centralized exchanges [11]. For solo staking, a user invests their own hardware, setting up an execution client and a consensus client, and directly participating in the consensus network [12]. Staking-as-a-service, of which the concept is very similar to cloud computing vendors providing services on the cloud, allows a third-party provider to run validators on behalf of users [13]. Although it saves users from setting up and maintaining validators on their own, staking-as-a-service typically involves a monthly fee. Pooled-staking, on the other hand, allows users who don't have 32 ETH to stake any amount so that they become part of a joint effort to act as a validator and earn rewards [14]. And finally, centralized exchanges are third-party providers maintaining large pools of ETH and executing many validators, which is preferred by users who don't want to hold their ETH and manage their keys [11].

While validators are chosen at random, a validator is more likely to get chosen to validate if it stakes more coins. If the selected validator fails to validate, its coins are "slashed" [15]. Slashing occurs when the blockchain takes away some of the staked coins put forward by a validator after in the event that the validator fails to validate [16]. The validators must have expected levels of connectivity and hardware when they are being called upon. Therefore, running a validator means that a node has issued a commitment to contribute to the overall security and veracity on the network.

4.3 Ethereum & Algorand: Validator Setup Complexities

Several considerations need to be made to set up an Ethereum 2.0 validator. Preliminary research includes hardware requirements, setup of an Ethereum 1 node that validates the initial 32 ETH deposit, client installation, and key generation and management.

The process of setting up an Algorand "node" is prescribed in their documentation. It also addresses some of the requirements posed at the start of this section [17]. Algorand node setup may have differing complexities from unanticipated problems. These problems may be hard to solve through the smaller community size compared to that of Ethereum. Presently, there is no incentive to run an Algorand node, other than to further decentralize the Algorand network.

5 Cognitive Walkthrough on Validator Setup

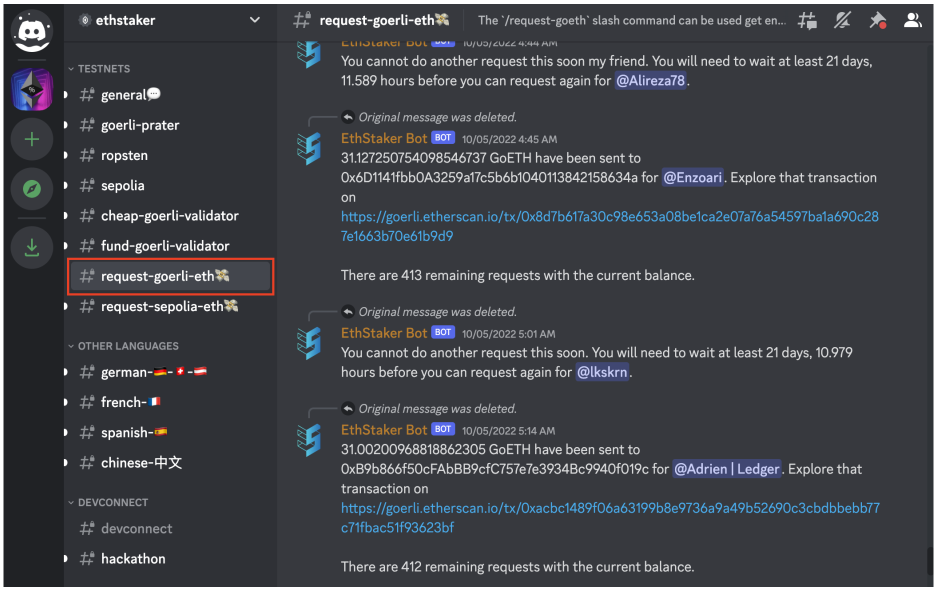
Eth Validator node setup became more complex post the Merge, PoS consensus. In this section, we evaluate this claim through usability analysis using the cognitive walkthrough method. Our goal is to assess how easy it is for a potential validator with moderate computer literacy to participate in the Ethereum network compared to other PoS consensus projects, namely setting up an Algorand relay node. We completed the setup of an ETH validator on the Ethereum test network, Goerli, and stopped at each step to identify possible mistakes or complexities faced by the user. The following subsections present our feedback on key elements of the validator setup process.

5.1 The Stake

One potential complexity of the validator process is that you will need to have 32 Ethereum available in your Ethereum account in order to make the validator deposit. If you do not already have this amount of Ethereum, you will need to purchase it on an exchange or obtain it through another means. This can be a complex process, especially if you are new to cryptocurrency.

For testnet, new validator must have a MetaMask account connected to the Goerli Test Netowork. Most users request testnet ETH by joining the ethstaker discord and interacting with chatbot through the #request-goerli-eth channel. In order to receive funds from the faucet, the identity must be verified through BrightID, a social identity network to limit users from using multiple accounts. The BrightID verification process itself is cumbersome. It requires attending a connection party meeting via Zoom, finding legitimate connections as well as getting sponsorship via third-party apps. One example of how users can mistakenly navigate this step is attending the connection party meetup online. Although there is multiple slots with many languages for the parties, one must directly join the zoom session from the online calendar. The website gives an indication that you should have prior registration to join the meeting whereas one is expected to just click on the desired slot and join an active session the minute it starts (and wait to be admitted).

On the contrary, the staking step on Algorand is straightforward and simple. The Algorand organization provided a web faucet where users are expected to provide their Algorand address and test Algos will likely be dispensed to the account successfully on the first trial. The figures below shows the drastic difference between the Ethereum and Algorand faucets (Discord channel vs Algorand dispenser website).



EthStaker Bot BOT 10/05/2022 4:44 AM
You cannot do another request this soon my friend. You will need to wait at least 21 days, 11.589 hours before you can request again for @Alireza78.

Original message was deleted.
EthStaker Bot BOT 10/05/2022 4:45 AM
31.127250754098546737 GoETH have been sent to 0x6D1141fbb0A3259a17c5b6b1040113842158634a for @Enzoari. Explore that transaction on https://goerli.etherscan.io/tx/0x8d7b617a30c98e653a08be1ca2e07a76a54597ba1a690c287e1663b70e61b9d9

There are 413 remaining requests with the current balance.

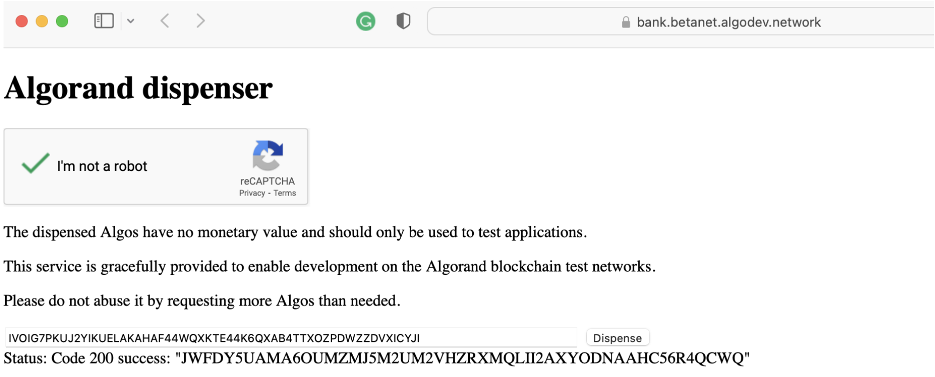
Original message was deleted.
EthStaker Bot BOT 10/05/2022 5:01 AM
You cannot do another request this soon. You will need to wait at least 21 days, 10.979 hours before you can request again for @lkskrn.

Original message was deleted.
EthStaker Bot BOT 10/05/2022 5:14 AM
31.002009688188623O5 GoETH have been sent to 0xB9b866f50cFAbBB9cfC757e7e3934Bc9940f019c for @Adrien | Ledger. Explore that transaction on https://goerli.etherscan.io/tx/0xacbc1489f06a63199b8e9736a9a49b52690c3cbdbbebb77c71fbac51f93623bf

There are 412 remaining requests with the current balance.

Figure 1: Ethereum Discord - Test Eth Faucet

Overall, while obtaining a 32 Ethereum stake to set up a validator is not a particularly unachievable process. However, it is not straight forward and does require some knowledge Ethereum validator setup process, setting up and maintaining a wallet on MetaMask, as well as going through multiple



Algorand dispenser

The dispensed Algos have no monetary value and should only be used to test applications.

This service is gracefully provided to enable development on the Algorand blockchain test networks.

Please do not abuse it by requesting more Algos than needed.

IVOIG7PKUJ2YIKUELAKAHAF44WQXKTE44K6QXAB4TTXOZPDWZZDVXICYJI

Status: Code 200 success: "JWFDY5UAMA6OUMZMJ5M2UM2VHZRXMQLII2AXYODNAAHC56R4QCWQ"

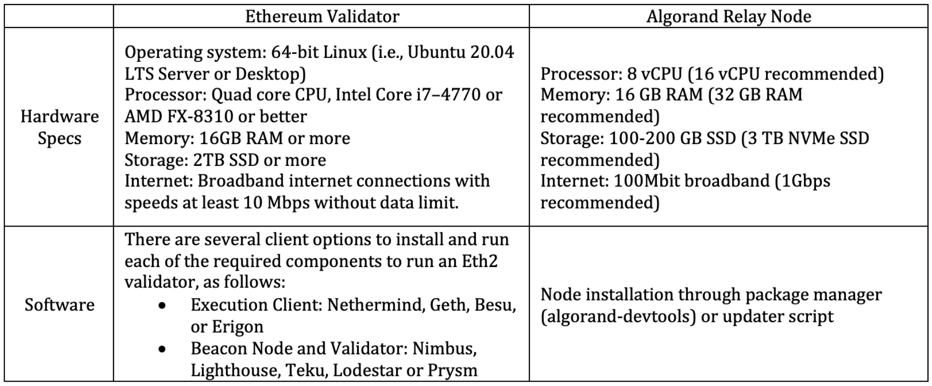
Figure 2: Algorand dispenser website - Test Algo Faucet

registrations and verification procedures (for testnet). It may take someone long time (weeks) without access to straightforward guidelines.

5.2 Environment Setup & Hardware Requirements

Setting up the Ethereum Validator environment requires intensive computing resources. To achieve this during our setup, we setup a VM instance on Google Cloud Computing Engine using the $300 free trial balance. Similar level is required to run an Algornad Relay Node. Generally, participating in blockchain cryptocurrency networks is hardware demanding compared to user daily average computing requirements. Nevertheless, it is similar for Ethereum validators and Algorand relay nodes.

Furthermore, there are several software options for setting up the environment to run the three components; Execution Client (formerly Eth1 node), Beacon node (consensus client), Validator. On the contrary, setting up an Algorand relay node is more straight forward and takes less time. Below table summarizes the environment setup requirements of the two PoS consensus networks.



	Ethereum Validator	Algorand Relay Node
Hardware Specs	Operating system: 64-bit Linux (i.e., Ubuntu 20.04 LTS Server or Desktop) Processor: Quad core CPU, Intel Core i7–4770 or AMD FX-8310 or better Memory: 16GB RAM or more Storage: 2TB SSD or more Internet: Broadband internet connections with speeds at least 10 Mbps without data limit.	Processor: 8 vCPU (16 vCPU recommended) Memory: 16 GB RAM (32 GB RAM recommended) Storage: 100-200 GB SSD (3 TB NVMe SSD recommended) Internet: 100Mbit broadband (1Gbps recommended)
Software	There are several client options to install and run each of the required components to run an Eth2 validator, as follows: • Execution Client: Nethermind, Geth, Besu, or Erigon • Beacon Node and Validator: Nimbus, Lighthouse, Teku, Lodestar or Prysm	Node installation through package manager (algorand-devtools) or updater script

Figure 3: Environment Comparison

5.3 Policies & Penalties

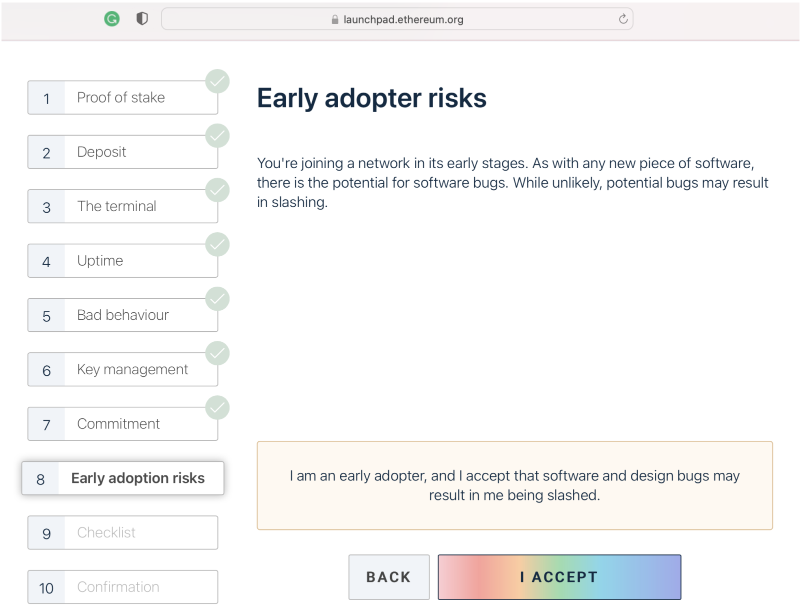
The verification process for Ethereum is much longer than for Algorand. It takes several days/weeks for the node to sync to the latest chain head. On the contrary, Algorand offers something known as ”Fast Catch-up”. This is a revolutionary feature that rapidly updates a node using catchpoint snapshots (a copy of the latest 10,000 transactions on the chain). This enables syncing of an entire node in minutes. [18]. As for penalties, Ethereum has several rules and slashable offenses that will be discussed extensively in later sections. What is worth noting is that these penalties are not clear throughout the validator setup process. Perhaps this due to the fact the setup guidelines are not developed and maintained by the Ethereum organization but through communities.

5.4 Key Management

Key management is a critical requirement to run a Ethereum validator network. There are several options to manage keys for Ethereum which can be quite confusing for basic users. During our setup, we were confused that we had the Wagyu Key Gen software. We later discovered that we can simply utilize the Staking Deposit Cli. This is a critical component for the validator to understand that we felt is perhaps made complex and not clear. As for Algorand, key management is achieved through the same packaged software. Both networks use mnemonic (seed phrase) to secure keys.

5.5 Limited Functionalities & Risks

The key idea with limitation within the Ethereum network post the Merge is the lack flexibility to withdraw or transfer deposited stake (32 Eth). For Algorand, there are no limitation plus the stake is minimal (0.01 Algo).



Early adopter risks

You're joining a network in its early stages. As with any new piece of software, there is the potential for software bugs. While unlikely, potential bugs may result in slashing.

I am an early adopter, and I accept that software and design bugs may result in me being slashed.

Figure 4: An Example of Risks Disclaimer on Ethereum

5.6 Incentives

Perhaps this section explains the main drive behind setting up an Ethereum validator being much desirable than setting up an Algorand relay node. The ethererum validator can earn 5.3%-7.3% [19]. As for Algorand, there are no validation rewards as of 2021. There is 9.4%-19% governance rewards only that you can earn leaving your stake for over 3 months. [20]

5.7 Security

The setup process warns users of security bugs and potential risks for Ethereum. It is even stated as a disclaimer on the launch pad that usesrs are responsible for the risk of being part of this community. This could be very deterring to join the netowrk. Whereas on Algorand, there are many asssuring statement and a sense of high security on the network.

5.8 Implementation Guidelines

Our experience setting up a Ethereum validator was dependent upon community guidelines. There are several sound and well documented steps on popular websites or provided by Ethereum client software companies. Nevertheless, documentation is not standardized and requires intense amount of time to go through and understand how to implement. For Algorand, there is much less guidelines available online but the documentation on the organization's main website was very easy to follow and implement.

6 Beacon Chain Structure & Behavior Analysis Methodology

6.1 Incentives and Penalties

The new consensus mechanism inserts new roles for validator nodes and uses rewards and penalties (*"slashing for protocol misbehaviour"*) to incentivize honest behaviour. It is defined as follows [21]:

$$base\ reward = \left[effective\ balance * \frac{base\ reward\ factor\ (64)}{base\ reward\ per\ epoch\ (4) * \sqrt{\sum active\ balance}} \right] \tag{1}$$

$$inclusion\ speed\ reward = base\ reward * \frac{1}{inclusion\ distance} * \frac{7}{8} \tag{2}$$

Rewards and penalties are based on the correctness of the source, head and Target and, can be either positive or negative, however, the inclusion delay can just be positive. The Source has to be correct.[21] The worst inclusion speed reward is $base\ reward * 1/32 * 7/8$ with an inclusion distance of 32 and the best is $base\ reward * 1/1 * 7/8$ with an inclusion distance of 1.[21]

"The so-called honest validators will be running well-designed clients, complying with the Beacon chain specifications, and avoiding penalties for incorrect voting. Or what could be worse, slashing for protocol misbehaviour"[21]. On the other side of the spectrum, dishonest validators will be *"slashed"* [22] i.e. means that the validator is forced to exit the beacon chain at a point in the future, receiving a number of penalties until it leaves.

In all cases, the offender needs to be caught in order for the slashing process to be triggered. The whistle-blowing validator will create and propagate a specific message containing the offence, for a proposer to include it in a block. Both the proposer and the whistle-blower will be entitled to a reward.[21]

The notion of incentives is intrinsically linked to system governance, where people respond to incentives or if badly designed can lead to unwanted behaviours (e.g collusion, under-performance, among others).

6.2 Validators and Attestations

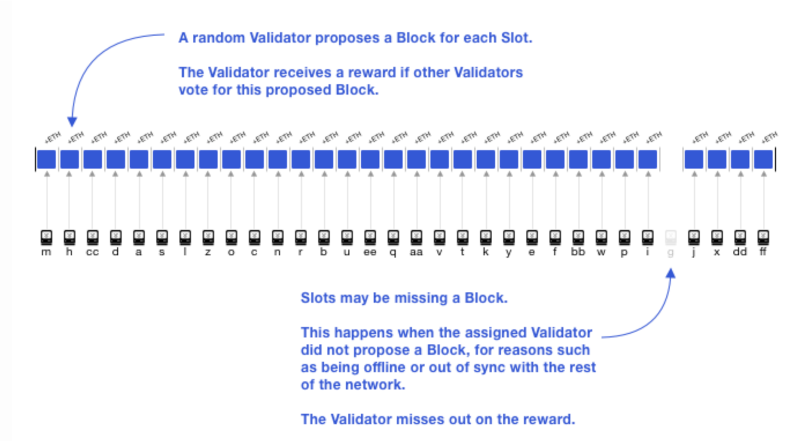
The flow of the Beacon chain is built on a unit of time called the slot. On average, every 12 seconds – a validator gets chosen to be the block proposer. Once the block is minted and propagated, an attester committee of validators vote for this block to be part of the canonical chain.

The purpose of committees in the Beacon chain is to distribute the validators, such that each one is able to vote once per epoch (every 32 slots). Validators within committees gossip among each other, enabling the aggregation of attestations.

If during a slot there is not a block proposed, it is identified as a skipped slot. In this situation, further proposals or attestations are built on the last block available from a former slot.

The proposer chooses over which block it will perform the state transition to the new canonical head of the chain. This election is made by the algorithm LMD GHOST fork choice (Latest Message Driven Greediest Heaviest Observed SubTree)[24]: The procedure picks the fork over which there is recursively the biggest weight in received votes. When validators attest this block, they are in fact, voting in favour of this fork choice.

In order to provide finality to the blockchain, i.e., the assurance that the state cannot be reversed, honest validators leverage the Eth2 implementation of Casper the Finality Gadget (FFG), providing



A random Validator proposes a Block for each Slot.

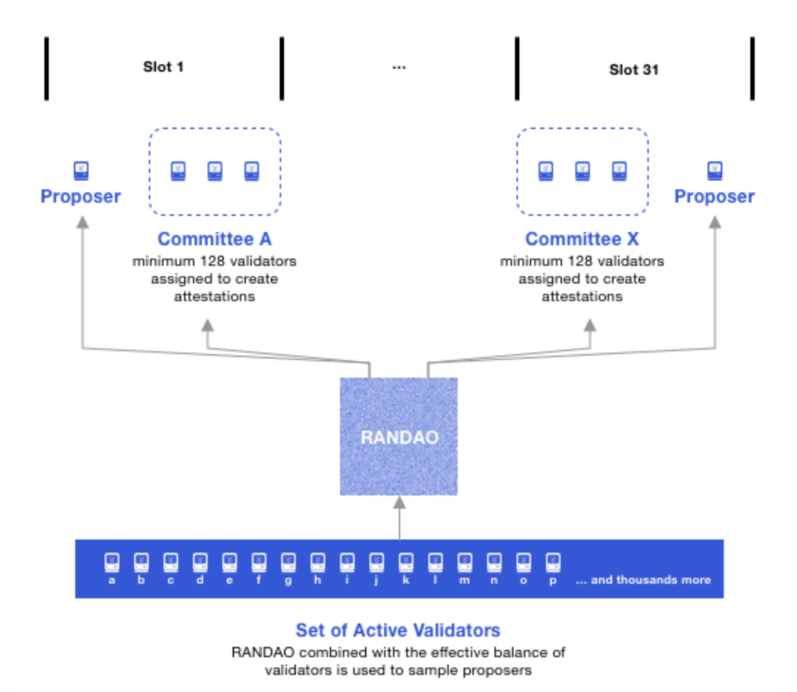
The Validator receives a reward if other Validators vote for this proposed Block.

Slots may be missing a Block.

This happens when the assigned Validator did not propose a Block, for reasons such as being offline or out of sync with the rest of the network.

The Validator misses out on the reward.

Figure 5: Beacon Chain [23]



Slot 1
...
Slot 31
Proposer
Committee A
minimum 128 validators assigned to create attestations
Committee X
minimum 128 validators assigned to create attestations
Proposer
RANDAO
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
... and thousands more

Set of Active Validators

RANDAO combined with the effective balance of validators is used to sample proposers

Figure 6: Beacon Chain RANDAO[23]

in their attestations two additional votes: One for the latest justified epoch (source), and one for the latest epoch boundary (target).

At the beginning of each epoch, attestations are counted. If there exists a supermajority (two thirds), the latest justified epoch checkpoint will be moved forward in time, and, under certain rules, finalization will be achieved either for the prior epoch, or for its antecessor.

If the system has not achieved finality in a number of epochs (4 by the current specification), all the validators in the beacon chain are hit with an inactivity penalty.

6.3 Transaction lifecycle and transaction flow

The block construction process involves the following steps to (see also Figure 11):

- Users and searchers send transactions to block builders through the public p2p txpool or through direct channels;
- Builders construct execution payloads using these transactions as well as header parameters provided by validators. Builders may directly set the validator's fee Recipient address as the coinbase address of the payload, or the builders may set their own address and include a transaction which transfers to the fee Recipient in the payload;
- Relays receive execution payloads from builders and verify the validity of the payloads as well as calculate the payload value (amount of ETH paid to the fee Recipient).
- Escrows receive the full execution payloads from relays to provide data availability;
- Validators receive execution payload headers from relays (execution payloads stripped of the transaction content). The relay must attach an indication of the payload value to each header. The validator selects the most valuable header, signs the payload, and returns it to the relay and escrow to be propagated to the network (see Figure 12).

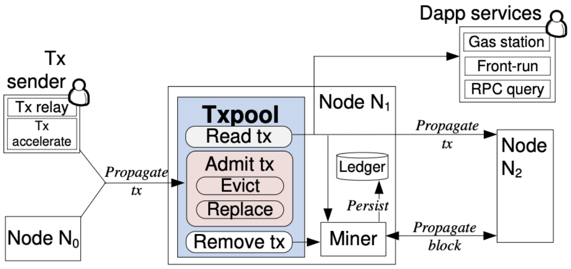


Figure 7: Ethereum transaction workflow [25]

As shown in Fig.7 [25], validators play a crucial role in each transaction, by selecting how transactions are organized and bundled.

The general "price" for each transaction is paid in gas, as shown in Figure 8 [26]

Once the transaction has been submitted the following happens:

- Once a transaction is sent, cryptography generates a transaction hash;
- The transaction is then broadcast to the network and included in a pool with other transactions;
- A validator must pick your transaction and include it in a block in order to verify the transaction and consider it *"successful"*;
- As time passes the block containing the transaction will be upgraded to *"justified"* and then *"finalized"*.
- These upgrades make it much more certain that the transaction was successful and will never be altered.
- Once a block is *"finalized"* it could only ever be changed by an attack that would have a great cost.

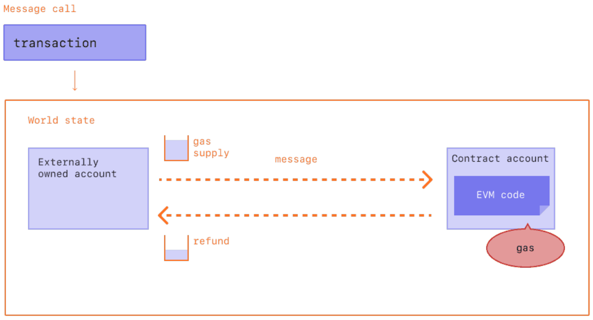


Figure 8: Gas Tx[26]

6.4 Mempool/txpool

When a transaction is sent to the blockchain it is received by an Ethereum node. This Ethereum node propagates the transaction to other peer Ethereum nodes (gossip algorithm[27]). While the transaction is waiting to be added to the next block it resides in a staging area called the mempool/txpool. This staging area contains a list of all *"pending transactions"*. Once the transaction is added to the next block it is considered a confirmed transaction (Fig.9)

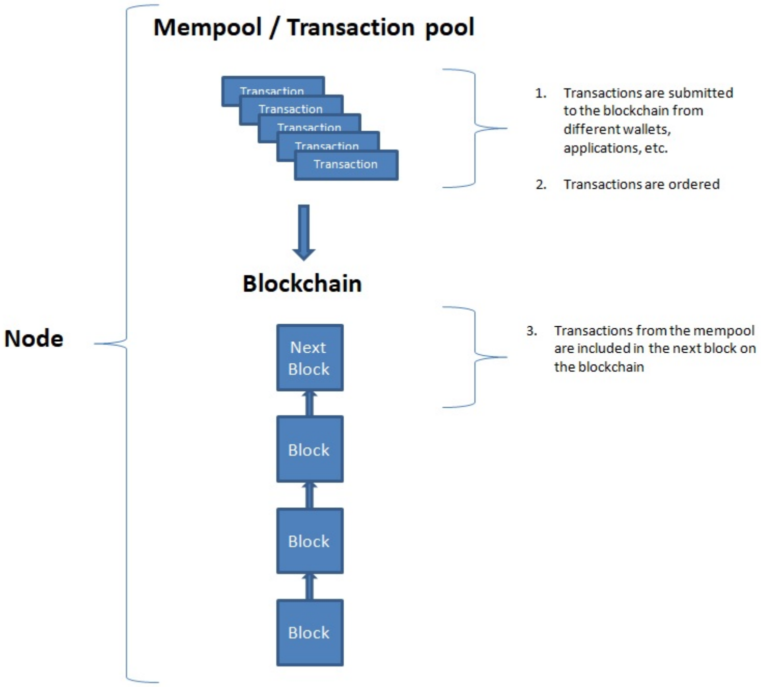

Mempool / Transaction pool
Transaction
Transaction
Transaction
Transaction
Transaction
1. Transactions are submitted to the blockchain from different wallets, applications, etc.
2. Transactions are ordered
Blockchain
Node
Next Block
Block
Block
Block
3. Transactions from the mempool are included in the next block on the blockchain

Figure 9: Mempool/Tx Pool [28]

Transactions from different wallets, applications, are submitted to the blockchain for processing. These transactions are pending and ordered in the mempool/txpool. This is how a blockchain node deals with transactions that have not yet been included in a block. Pending transactions in the mempool/txpool are a good indication of what is going to happen next on the blockchain. Examining transactions in the mempool will give additional information that you might need when trading, building applications, watching addresses or listening transactions.

6.5 Maximum extractable value (MEV) and Proposer/builder separation (PBS)

Maximal extractable value was first applied in the context of proof-of-work, and initially referred to as *"miner extractable value"* [29]. This is because in proof-of-work, miners control transaction inclusion, exclusion, and ordering. However, since the transition to proof-of-stake via the Merge validators have been responsible for these roles, and mining is no longer part of the Ethereum protocol. The value extraction methods still exist, though, so the term *"Maximal extractable value"* is now used instead.

MEV corresponds to all the possible value that can be reaped by a block proposer from the re-ordering, censorship or insertion of transactions within the block they are proposing. In order to understand transaction ordering in eth2, we first focus on the inner workings of the software that will be used for it.

As eth2 is essentially 2 chains merged together, the client is composed of two sub-clients (Fig. 10[30]), one for the execution engine and one for consensus. Current PoW Ethereum clients continue to exist in eth2 and are run alongside beacon clients, splitting responsibilities between each other.

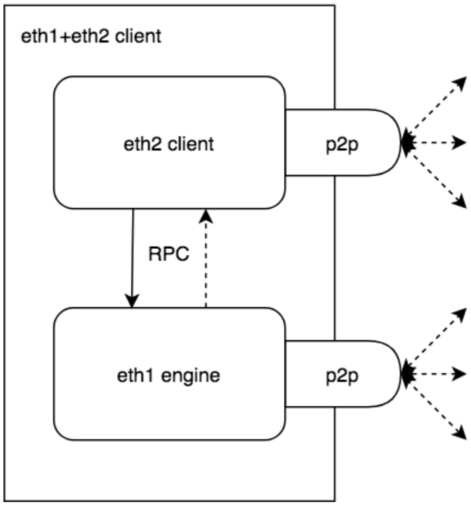


Figure 10: ETH2 Client [30]

The eth1 client is a PoW Ethereum client stripped of its consensus responsibilities to focus solely on the eth1 tx pool, eth1 execution validity and the EVM. The beacon client on the other hand is responsible for consensus and validator duties (eg. beacon block attestation and proposal). They are run concurrently, each maintaining their own p2p networking stack (libp2p for beacon, devp2p for eth1).[1]

6.6 MEV-Boost relays

Some MEV-Boost relays are regulated under OFAC and will censor certain transactions.

With the emergence of Maximum extractable value (MEV) and Proposer/builder separation (PBS), the control of transaction inclusion, exclusion, and ordering has been playing a crucial rule (.e.g. `https://boost.flashbots.net/`). See Figures 11,12.

7 Results

7.1 General description

As of December 11, 2022 there 48 7920 validators (only 1257 inactive), and few exits (946 exits and 728 voluntary exits (`https://beaconcha.in/validators`)

The participation rate has been high (more than 95% as shown in Figure 14).

[1] See, i.e.`https://ethereum.org/en/developers/docs/mev/`

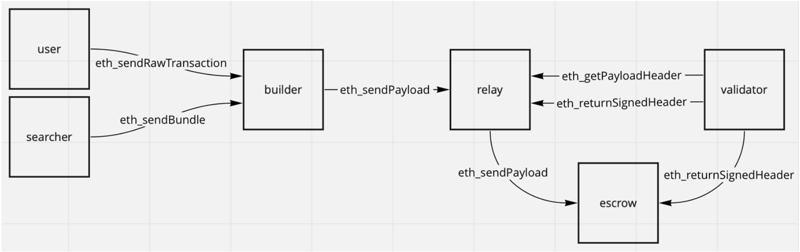


Figure 11: Architecture (MEV-Boost) [26]

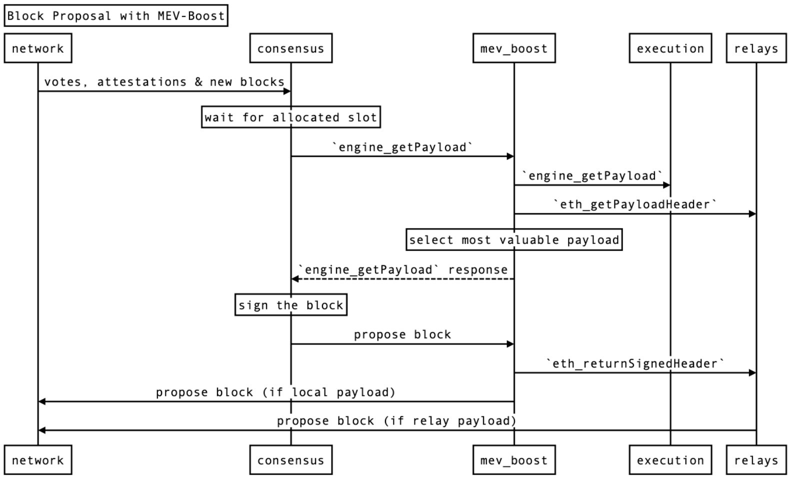


Figure 12: Communication (MEV-Boost)[26]

Most validators are still concentrated in pools, where ”lido”, has 128 858 validators (26.46%), followed by ”coinbase”, with 59 010 (12.12%), ”kraken” 35 791 validators (7.35%) and ”binance” 27 047 (5.55%). The average gas used per block has been increasing, as shown infra in figure 15, surpassing today more than 15 M gwei.

The average time per block is 12 seconds (see Figure 16), namely after the *”Merge”* (September 2022), much lower than the all-time high of 2017, with more than 30 seconds.

Finally, the average balance of validators, per bin, is 33.6-33-8 ETH as shown in figure 17.

7.2 Slashable Offenses

Validators in Ethereum 2.0 will be penalized if their activities are too slow, or if they engage in malicious behaviors. When a validator fails to provide the vote for the source checkpoint and the target checkpoint in the current epoch, it's assumed that the validator is acting too slowly and the penalty is equal to the amount of reward the validator would receive if it actually provided the vote [31]. This is not a slashable activity and the validator will not be evicted from the network. On the other head, there are also three behaviors that are most likely to be conducted by malicious parties: proposer violations, double votes, and surround votes [23]. The last two behaviors can also be categorized as attestation violations since they both involve conflict votes for checkpoint attestation. A validator will be slashed and forced to leave the network if their slashable behaviors are discovered by a whistleblower (another honest validator) in the network. Unlike voluntary exit, in which validators leaving the network can withdraw their stakes within 27 hours, a slashed validator will wait for around 36 days to withdraw its stake. In this 36-day period, if more validators are caught conducting slashable behaviors, additional penalties will be applied to those slashed validators. And if 1/3 validators are slashed, the entire balance of those validators will be drained away [23].

In this section, we will focus on analyzing slashable offenses. Specifically, we want to answer: how many validators are slashed? How frequently do whistleblowers report the three types of slashable violations? What's the pattern of slashable offenses over time? We first dive into the offenses that

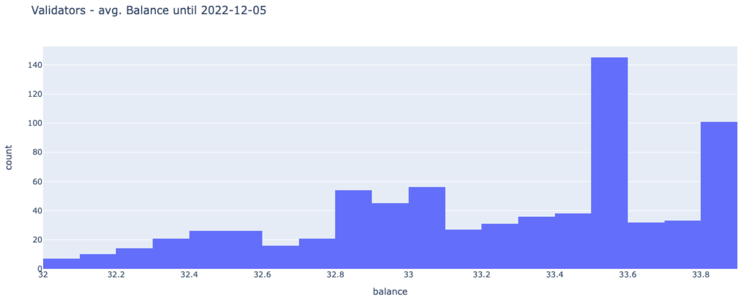


Figure 13: Number of Validators (as 05/12/2022

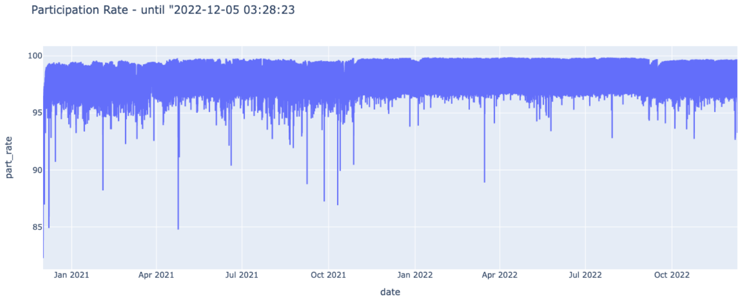


Figure 14: Participation rate (as 05/12/2022

were reported on the blockchain. We then try to sample some attestation data from the blockchain, trying to find if there is any offense going unreported.

7.2.1 Proposer Violations

The definition of proposer violations is that a proposer crafts and signs two different blocks at the same slot. To fetch all reported proposer violations, we first crawl the slashing data on https://beaconcha.in/validators/slashings and filter it by slashing reasons (Proposer Violation). We then made requests to the Beacon API (https://beaconcha.in/api/v1/block/{SLOT}/proposerslashings) to fetch the detail of each slashing. Figure 18 shows the 18 reported proposer violations over time.

Compared to attestation violations, proposer violations are relatively easy to detect by whistleblowers. As a result, malicious validators certainly do not want to engage in this behavior, which is why proposer violations are rare over time as we can see from Figure 18. A possible explanation for these known proposer violations is that the client software has a bug, or the client software is getting an upgrade in one slot and losing the knowledge that it has proposed a block for that specific slot [32]. The latter one is more plausible before epoch 50,000 as three proposer violations took place with almost similar time intervals, which resembles a periodical software upgrade.

7.2.2 Attestation Violations

Similarly, we fetched all reported attestation violations by crawling https://beaconcha.in/validators/slashings and then making requests to the Beacon API (https://beaconcha.in/api/v1/block/{SLOT}/attesterslashings). Figure 19 shows that the number of reported attestation violations is ten times more frequent than that of proposer violations.

We then dive deep into how frequent are double votes and surround votes respectively by analyzing the detailed attestation violation data. Concretely, each item of the attestation is a structure containing the hash of the source checkpoint (s), the hash of the target checkpoint (t), the epoch of the source checkpoint($E(s)$), and the epoch of the target checkpoint($E(t)$). Since Ethereum uses Gasper Finality

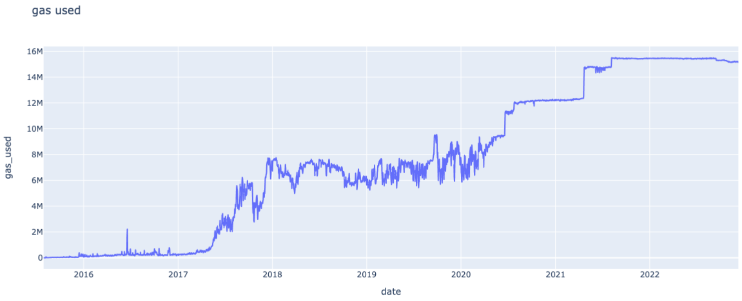


Figure 15: Average gas used per block

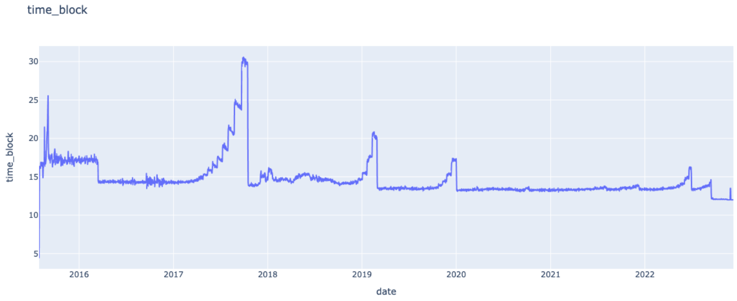


Figure 16: time block

System [33], we can derive the slashing conditions from [33] that it's a double vote if a validator signs two conflict attestations(attestation 1 and attestation 2) that satisfy

$$E(t_1) = E(t_2) \tag{3}$$

and that it's a surround vote if a validator signs two conflict attestations(attestation 1 and attestation 2) that satisfy

$$E(s_1) < E(s_2) < E(t_2) < E(t_1) \tag{4}$$

Based on the above two slashing conditions, we can distinguish two types of attestation violations in our slashing data. The pie chart in Figure 20 shows that surround votes only account for around 5% of total attestation violations. One possible explanation is that surround votes are more complex for validators to discover (The whistleblower can build a hash table for each target checkpoint of each validator so that discovering double votes is $O(1)$. However, to detect surround votes, the time complexity changes to $O(N)$). It's reasonable to speculate that the actual number of surround votes is much larger than that of reported ones, which leads to the question we want to answer in the next section: is there any violation that is going unnoticed and fails to be reported?

7.2.3 Unreported Violations

In this section, we want to detect any unreported attestation violations on the blockchain. A comprehensive approach would require us to scan all attestation data included in each historical block since genesis and compare each pair of attestation data for each validator, which is quite infeasible to us due to resource limitations. One thing is that the Beacon API has a limit of 10 requests per minute per IP. It would take one machine $165700 * 32/(10 * 60 * 24) = 368$ days to scan the entire blockchain. Another thing is that storing attestation data for each validator in memory is also infeasible. It would require us to build a complex, disk-based approach to walk through the attestation data, which is both I/O intensive and CPU intensive. Therefore, we decided to adopt a sampling approach to infer the frequency of unreported violations on the blockchain.

We first draw a figure to illustrate attestation violations over time, which is shown in Figure 21. Then we pick two periods, one is epoch 14,000 to epoch 14,500 which has a spike in reported attestation

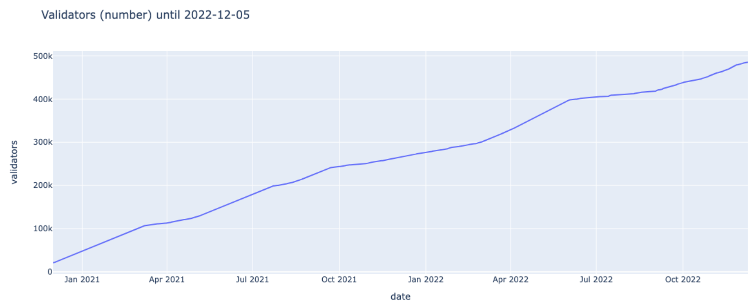


Figure 17: Validator's balance (as 05/12/2022

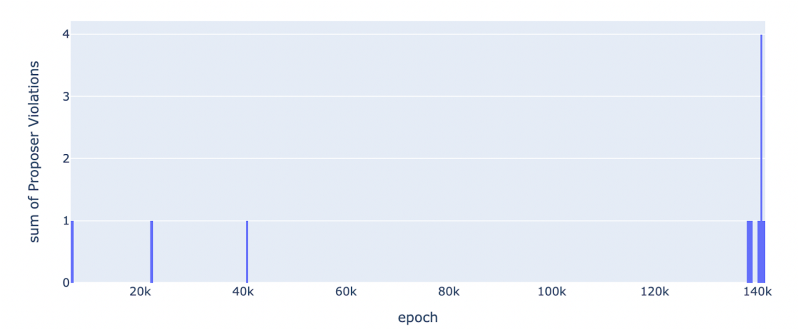


Figure 18: Reported proposer violations over time.

violations, and the other is epoch 136,000 to epoch 136,500 in which no violation is reported. Scanning attestation data in each period takes us about $500 * 32/(10 * 60) = 26$ hours and the entire dataset is dumped to the disk. After that, we load the dataset part by part to the 16-GB memory (carefully avoiding memory shortage), record attestation for each validator, and go through each validator to look for slashable offenses. Since computation for each validator is independent, we use ThreadPool to compute each validator in parallel, which reduces the computation time for 250 epoch from around 2 hours to less than 1 hour. In the end, it turned out that no unreported offense is found for both epoch 14,000 to epoch 14,500 and epoch 136,000 to epoch 136,500.

7.3 Censorship

In this section, we want to analyze censorship from validators. Specifically, are there any transactions that are never included in the blockchain? Are there any transactions that are replaced and dropped?

7.3.1 Data Collection

Thanks to the CMU full node provided by the Professor, we can monitor the Ethereum transaction pool and collect data for analysis. We first launched a function that subscribes to the 'newPendingTransactions' event and records each arriving transaction to a dictionary in memory. The key is transaction ID and the value is a nested dictionary that contains: arrival time of the transaction, hangup time of the transaction (in the transaction pool), a *is_removed* flag indicating whether it's removed from the pool, and also transaction detail. We then launched another function in parallel, which polls the transaction pool every second and updates the transaction dictionary in memory. Specifically, if the transaction is still seen in the transaction pool, we update its hangup time. Otherwise, we set the *is_removed* flag and stop updating the hangup time. We kept the program running for an hour and dumped the transaction dictionary and the final transaction pool snapshot to the disk for further analysis.

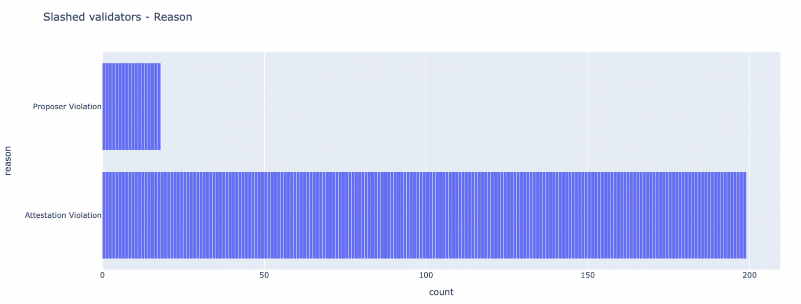


Figure 19: Frequency comparison between proposer violations and attestation violations.

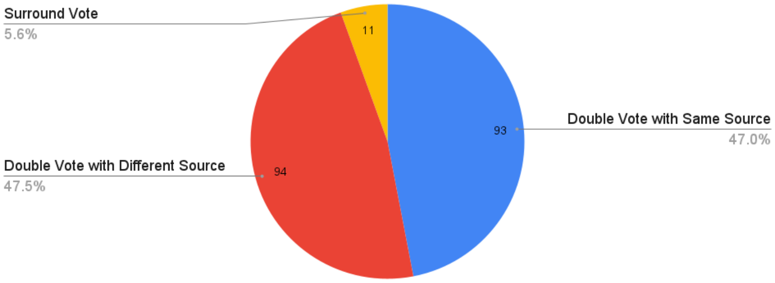


Figure 20: The frequency of different types of attestation violations.

7.3.2 Analysis of Transaction Hangup Time

During the 1-hour experiment, we found that 536 transactions are never removed from the transaction pool. By looking at the transaction detail of these transactions, we found that 99.8% of them didn't set the 'maxPriorityFeePerGas' field. This indicates that almost all of these transactions didn't provide a tip for the miner and that the miner lacks the incentive to include these transactions in the proposing block. On the other hand, there are also over 10,000 transactions that are actually removed from the transaction pool. Figure 22 illustrates the relationship between the amount of tip provided in the transaction pool and the transaction hangup time (Tip is calculated by $maxPriorityFeePerGas * gas$ and hangup time is in seconds). From this figure, we can conclude that a higher tip (above 0.01 ETH) almost certainly guarantees a low hangup time (lower than 100 seconds) in the transaction pool.

7.3.3 Analysis of Dropped and Replaced Transactions

Transactions on the blockchain can be uniquely identified by the sender address and a nonce which is essentially a sequence number. Under normal conditions, if a transaction is sent by the sender, the sequence number will be incremented so that no transactions from the same sender will have the same nonce. However, sometimes senders also send transactions that have the same nonce as previous transactions so that previous transactions can be dropped and replaced by the transaction pool [34]. We want to analyze how frequently this event occurs and also the incentives behind it.

To collect the data, we added some code into the program in Section 7.3.1 so that when each new transaction arrives at the transaction pool, it's recorded in another dictionary. This time, the key is the combination of sender address and nonce and the value is a list of transaction detail. If multiple transactions with the same nonce are sent from the same sender, there will be more than one transaction in the transaction detail list. Similar to previous experiment settings, we ran the program for an hour and dumped the dictionary to the disk for further analysis.

According to Figure 23, we can see that 97.4% of total transactions (14706 transactions) during the experiment are never dropped and replaced. Around 2.4% transactions (361 transactions) are dropped and replaced once and around 0.2% transactions (33 transactions) are dropped and replaced more than once. After diving deep into the details of these 394 transactions that experienced being dropped and replaced, we found that 86% of them share the same transaction hash, which means that these

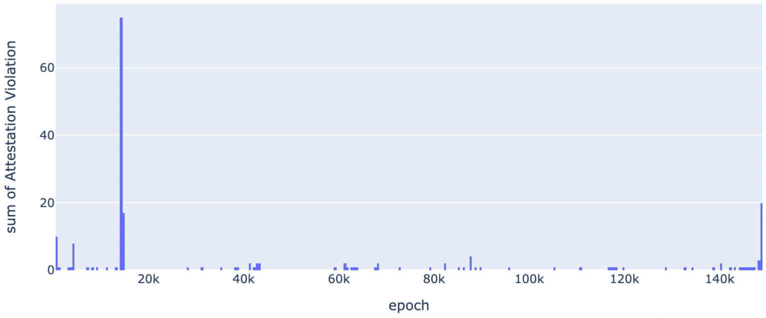


Figure 21: Reported attestation violations over time.

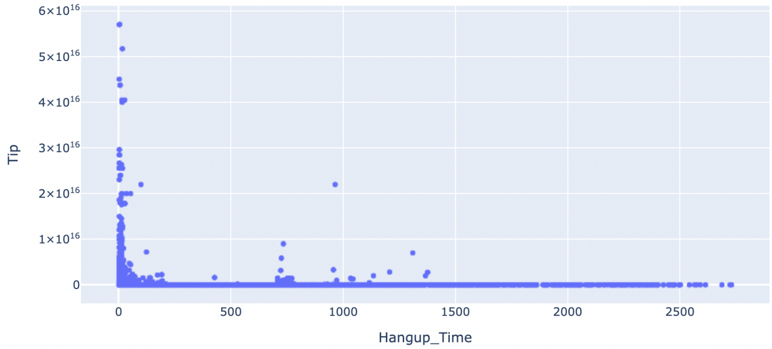


Figure 22: Relationship between tip provided by the transaction and its hangup time in transaction pool (in seconds).

transactions are completely identical. The reason behind this might be that a transaction is mistakenly sent more than once, or that a transaction is heard from different peers of the node more than once. On the other hand, 13.4% of them have different tips (calculated by $maxPriorityFeePerGas * gas$) included in the transaction, indicating that the sender is probably trying to increase the tip of the transaction so that it has a better chance of being picked up by the miner. Also, 0.2% of them have different data in the 'input' field of the transaction, indicating that the sender is probably trying to modify the input data.

7.4 MEV Relays

In this section it is analyzed how many blocks were built using relays and from those how many censor transactions. It was used a dataset with 6 weeks of analysed data [35], where it was used a variation of "mev inspect tool" from flash bots (`https://github.com/flashbots/mev-inspect-py`[35])[2].

Since *"the Merge"* it is visible the increase in the use of MEV, but also *"flash-bots"* taking the lead, as shown in Figures 24 and 25:

From a small sample (1000 blocks, from block 16157791 to 16158791) of 11 December 2022, extracted using beacon chain API (`https://beaconcha.in/api/v1/execution/block/`) is was analyzed the *"producer Reward"*, by comparing *"block rewards"* versus *"MEV rewards"* wherein the last, sometimes outperforms block rewards as shown in figure 26. The maximum difference found (producer Reward - MEV Reward) was of 0,3507352572793987 ETH, where the blocks that did not use any, the Producer Reward is equal to the block reward and the MEV reward is 0.

Regarding the use of relays more than 43% use *"flashbots"*, and only a few do not use any MEV relay as shown in figure 27.

[2] To use this approach would demand an achieve node, with EVM traces *"transaction tracing tool"* enabled, e.g. with Erigon. Currently, it is not supported by GETH

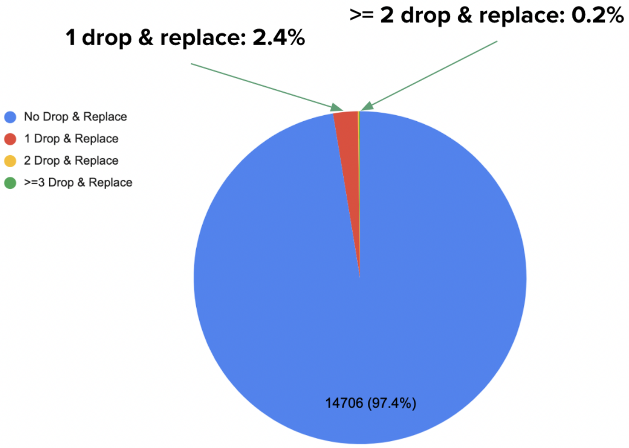


Figure 23: Pie chart for 'transaction drop and replace' behaviours in transaction pool.

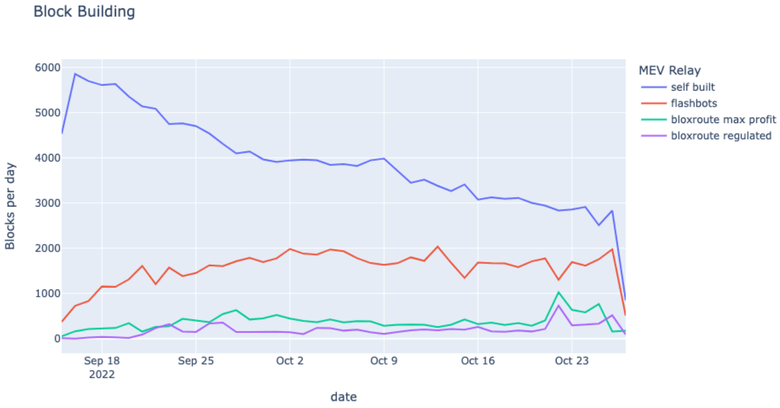


Figure 24: Self Built [35]

Similar to the case of mining pools there is also a high concentration[3] of MEV relays (or the building of blocks is let to few relays). MEV is one of Ethereum's biggest issues, with more than 680 million US$ extracted from users of the network year-to-date [36], whereas 99% are from arbitrage [36].

Considering the use of *"filtering or censorship"* some censor (namely *"flashbots"*) [37].

The most significant censorship observed is for the *"OFAC"* list (including addresses and contracts, namely *"Tornado Cash"* [4][38]. Even though most addresses have been *"dormant"* since OFAC ban back in January 2022, it is possible to see a few transactions [5], e.g. transaction of 10 ETH, with transaction hash: "0x6f60a4aa-7058dab153a859adfb139362d4bc395145528371ed90b127e528c7e7", of November 19, 2022, included on block 16001892 [6]) where the is no indication of the use of any MEV relay.

We also did not find any banned transactions, when listening to the "tx pool"(see section7.3).

7.5 Impact of different censorship inclusion rates

Another supposition is the impact of different censorship inclusion rates for future blocks, by extrapolating different inclusion rates, with the average block taking 12 seconds (as observed in Figure 21 on 7.1 and increasing n blocks:

$$tx\ inclusion\ rate = 1 - (compliant\ blocks\ rate^{n}) \tag{5}$$

[3] see, e.g.https://github.com/eth-educators/ethstaker-guides/blob/main/MEV-relay-list.md

[4] see https://home.treasury.gov/policy-issues/financial-sanctions/recent-actions/20220808

[5] https://etherscan.io/address/0x910cbd523d972eb0a6f4cae4618ad62622b39dbf

[6] see https://beaconcha.in/block/16001892

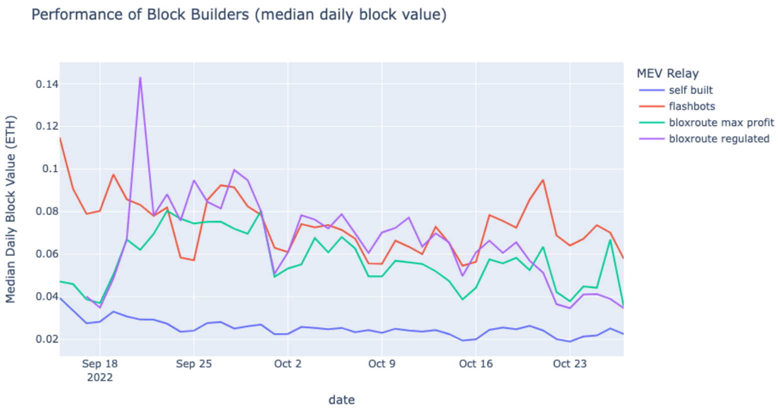


Figure 25: Performance[35]

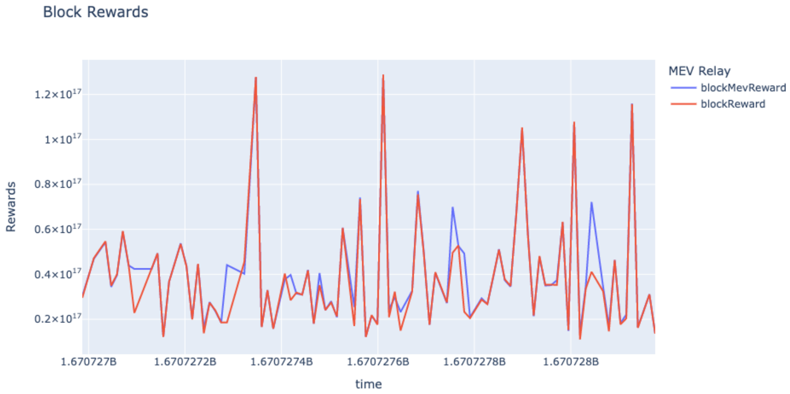


Figure 26: Block Rewards

Where *compliant blocks rate* is the percentage of compliance rate and n the number of blocks. Assuming the average 70% rate of censorship [7] (mostly driven by *"flashbots"*), the only impact will be on the delayed transaction time, as shown in the table 7.5

Even considering a 95% compliance rate, after 5 blocks (approx. 60 seconds) the censorship rate drops to 22.62%. Transaction censorship seems to only have a drawback of a longer period to confirm transactions, but as not 100% of blocks are built with transaction-filtered relays, will always be included in the block.

8 Discussion

The separations of builders, can change incentives but also increase opportunities for opportunist behaviour, like price manipulation, leading to market volatility, front and back running, sandwich attacks, liquidation and time bandit attacks [39] among others.

MEV transactions are often concentrated in the hands of a few *"builders"*, leading to a lack of decentralization and should supposition also be analysed.There is also the lack of transparency, as MEV transactions are often not transparent, making it difficult to track and audit.

MEV transactions can lead to network congestion, which can cause delays in processing transactions,

The problem of MEV was first identified in 2014 (Reddit) — a year before Ethereum launched by

[7] usig "mevwatch.info" API: `https://www.mevwatch.info/api/blockStats`

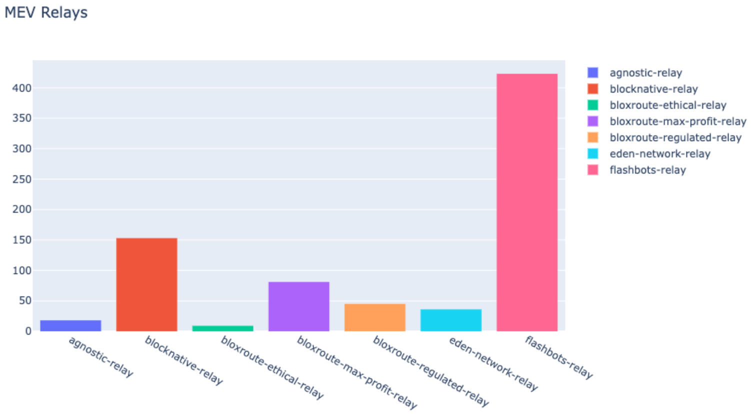


Figure 27: MEV Relays

number of blocks	time (block x 12 sec.)	tx inclusion rate (%)
1	12	30.00%
2	24	51.00%
3	36	65.70%
4	48	75.99%
5	60	83.19%
6	72	88.24%
7	84	91.76%
8	96	94.24%
9	108	95.96%
10	120	97.18%

Table 1: tx inclusion rate (70%)

an analyst operating under the pseudonym *"Pmcgoohan"* [8], where it was recognized that miners had total control of the transaction inclusion and ordering process, which meant that they could leverage this power to extract value from unsuspecting users of the protocol when it went live. In 2019, a group of researchers highlighted the same issue in a paper *"Flash Boys* 2.0"[40], where the term *"MEV"* was formulated to describe the problem. Thereafter, Georgios Konstantopoulos' and Dan Robinson's *"Ethereum is a Dark Forest"* [41], consolidated MEV as a fundamental concept in crypto-economics and highlighted its importance as one of the most challenging and pressing issues the Ethereum research today.

It is not clear the actual possibility or even benefit of banning certain addresses or contracts, besides a potential longer time to a transaction be included in a block.

9 Conclusions

Overall, the analysis of attestation violations showed that most double voting was due to synchronization problems. No unreported violations were found, and it was observed that a high *"tip"* almost certainly guarantees a low hangup time in the transaction pool. Few transactions are dropped and replaced in the mempool and, most of them are due to gas increases. The increased use of MEV relays may demand further analysis.

10 Limitations and future work

There we several limitations and potential future work:

[8] https://www.reddit.com/r/ethereum/comments/2d84yv/miners_frontrunning/

- Data collection, larger periods/sample;
- Compare with other data sources;
- Comparing mem pools (different sizes and setup rules, e.g. read/write when it is full);
- Tx censorship (the drawback is a longer period to confirm transactions, but as not 100 of blocks are built with transactions filtered relay, will always be included in the block);
- Attacks on mempool (e.g. Denial of Ethereum Txpool sERvices[25]);
- MEV and governance (front-running, sandwich, liquidation);
- Mining pools and transaction relay services centralization.

There were commons constraints, namely:

- Hardware, data, broadband, memory, requirements;
- Process flow;
- Synchronization;
- Access (tx pool and archive node, or more than 128 past blocks);
- Changes on the API (parameters, schema, etc).

Code Appendix

Code and data available at: https://github.com/d-vf/Ethereum-Validators-project

Acknowledgement

We would like to thank Professor Nicolas Christin, for the support, namely to allow the use of an already synchronized full node", besides all the patience and explanations provided.

References

[1] Ethereum, "The merge." https://ethereum.org/en/upgrades/merge.

[2] Digiconomist, "Ethereum energy consumption index." https://digiconomist.net/ethereum-energy-consumption.

[3] Ethereum, "Ethereum energy consumption." https://ethereum.org/en/energy-consumption/.

[4] M. C. Lab, "Mit cryptoeconomics lab." https://mitsloan.mit.edu/cryptoeconomics-lab/welcome-mit-cryptoeconomics-lab.

[5] Algorand, "Proof of work, proof of stake & pure proof of stake: An evolution in distributed consensus." https://www.algorand.com/resources/blog/proof-of-stake-vs-pure-proof-of-stake-consensus, Apr 2021.

[6] E. Shi, "Foundations of distributed consensus and blockchains." https://www.distributedconsensus.net/, 2020.

[7] "Intro to ethereum staking and validators." https://blockdaemon.com/products/white-label-validator/ethereum-introduction/.

[8] Alwin, "Proof-of-stake (pos)." https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/, Aug 2022.

[9] "Glossary." https://ethereum.org/en/glossary/#deposit-contract.

[10] J. Cook, "Keys in proof-of-stake ethereum." https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/keys/, Aug 2020.

[11] "Staking with ethereum." https://ethereum.org/en/staking/.

[12] "Solo stake your eth." https://ethereum.org/en/staking/solo/.

[13] "Staking as a service." https://ethereum.org/en/staking/saas/.

[14] "Pooled staking." https://ethereum.org/en/staking/pools/.

[15] "What is proof of stake? how it works (animated) + ethereum 2.0 upgrade!." https://www.youtube.com/watch?v=x83EVUZ_EWo&ab_channel=WhiteboardCrypto, Jul 2021.

[16] Cointelegraph, "Ethereum 2.0 staking: A beginner's guide on how to stake eth." https://cointelegraph.com/ethereum-for-beginners/ethereum-2-0-staking-a-beginners-guide-on-how-to-stake-eth, Dec 2021.

[17] JasonWeathersby, "Install a node." https://developer.algorand.org/docs/run-a-node/setup/install/, Feb 2020.

[18] D. Murphy, "Use quick-algo to start running an algorand node in under 1 minute." https://developer.algorand.org/solutions/use-quick-algo-to-start-running-an-algorand-node-in-under-1-minute/.

[19] V. Sopov, "Here's how much ethereum validators will earn post-merge." https://u.today/heres-how-much-ethereum-validators-will-earn-post-merge.

[20] A. Foundation, "Algorand governance rewards period 2." https://www.algorand.foundation/news/algorand-governance-rewards-period-2.

[21] Ethereum, "Ethereum 2.0 knowledge base - rewards and penalties.." https://kb.beaconcha.in/rewards-and-penalties.

[22] Ethereum, "slash validators." https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/beacon-chain.md#slash_validator.

[23] "The beacon chain ethereum 2.0 explainer you need to read first." https://ethos.dev/beacon-chain.

[24] V. Buterin, D. Hernandez, T. Kamphefner, K. Pham, Z. Qiao, D. Ryan, J. Sin, Y. Wang, and Y. X. Zhang, "Combining GHOST and Casper," May 2020. arXiv:2003.03052 [cs].

[25] K. Li, Y. Wang, and Y. Tang, "DETER: Denial of Ethereum Txpool sERvices," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, (Virtual Event Republic of Korea), pp. 1645–1667, ACM, Nov. 2021.

[26] Ethereum, "ETHEREUM DEVELOPMENT DOCUMENTATION," Aug. 2022.

[27] D. Shah, "Gossip Algorithms," *Foundations and Trends® in Networking*, vol. 3, no. 1, pp. 1–125, 2007.

[28] cryptomarketpool, "How to query the Ethereum mempool / txpool with Python."

[29] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the Instability of Bitcoin Without the Block Reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, (Vienna Austria), pp. 154–167, ACM, Oct. 2016.

[30] ethresear, "Eth1+eth2 client relationship," Apr.

[31] "Proof-of-stake rewards and penalties." https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/rewards-and-penalties/.

[32] "eth2 insights: slashings." https://www.coinbase.com/cloud/discover/dev-foundations/eth2-insights-slashings, Jan 2022.

[33] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[34] "What happens when a dropped transaction is replaced by another transaction?." https://info.etherscan.com/dropped-and-replaced-meaning/.

[35] pintail, "Since the Merge: How Are Things Changing?," Oct. 2022.

[36] flashbots, "Flashbots explorer."

[37] E. S. Educators, "ethstaker guides."

[38] "Cyber-related designation," Aug 2022.

[39] Ethereum, "Maximal extractable value (mev)."

[40] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges," Apr. 2019. arXiv:1904.05234 [cs].

[41] D. Robinson and G. Konstantopoulos, "Ethereum is a Dark Forest," Aug. 2020.