This post examines the potential timing dynamics at play between validators and builders, particularly in the context of post EIP-4844 with blob transaction enabled. We analyze various scenarios, discussing preventable measures for each.

Vanilla Validators Without MEV-Boost:

Approximately 8% of blocks are produced without MEV-Boost, suggesting a similar proportion of validators do not use MEV-Boost. These validators are straightforward to reason, and they adhere to the default behavior of standard client implementation, including transaction from public mempool based on predetermined code. If a transaction meets the base fee requirements and sometimes a priority fee as per EIP-1599, it gets included in the block. These validators do not engage in censorship or MEV extraction based on the nature of transactions. Representing the essence of blockchain in its purest form, they are integral to its fundamental principles.

In the post EIP-4844 world, validators maintain a consistent approach, including blob transactions from the blob mempool as long as they are accompanied by adequate fees. This consistency simplifies economic modeling and analysis for entities like Layer 2 solutions or others aiming to publish blobs on the Ethereum network.

Validators Using MEV-Boost

The remainder of validators utilize MEV-Boost, indicating that block construction is outsourced to external builders, leading to increased centralization. In this scenario, validators play a limited role, focusing solely on constructing the consensus aspect of the block while deferring the execution component to the builders. This process involves validators signing the block and broadcasting it before a deadline (4s into the slot) to avoid reorg.

In most client implementation, a validator's local execution block can override a builder block under specific triggers:

1. The local block contains a higher payout than the builder block.
2. The local execution client signals to override the builder block. The Execution API already supports this, but it requires implementation by the client. This feature is anticipated to be live in 2024. Potential signals include a transaction paying significantly more than the usual priority fee, not being included for several slots, or being included in many blocks that have been reorg out.
3. The proof of stake chain is not healthy, referred to as the MEV-Boost circuit breaker. Indicators of this issue include missing consecutive blocks or more than a certain number of slots in an epoch.

In the post EIP-4844 world, builders, not validators, have the final say on the contents of blobs. Builders can choose which blob transactions to include or exclude entirely. However, why is this process complex, and why wouldn't builders include certain blob transactions as the default behavior of the software? Here are some speculative guesses:

1. Reorg risk

: Including blob transactions may increase the orphan rate. In the latest Deneb Goerli shadow fork, it was observed that blob transactions can extend the time to process a block from 400ms to 2.5s at the 95th percentile. Although no reorgs were recorded during this period, why would builders want to assume this additional risk, especially in blocks with significant MEV?

1. Builders may exhibit bias

: They can prioritize blob transactions from certain sources. Treating not all blob transactions equally. A builder might favor a particular transaction from an entity over others due to factors such as shared VC backing, alliance membership, or personal relationships.

To guarantee the inclusion of their blob transaction, blob publishers are compelled to increase their fees, despite already meeting the chain-reported sufficient amount. This results in a poor UX. Moreover, they must continue raising their fees until the transaction is accepted. Additionally, this approach complicates the economic modeling of the blob market, as it introduces uncertainty due to the potential for builder bias.

What can the Ethereum community do?

Before implementing protocol enforcement measures like a forced inclusion list, the best current approach is monitoring and alerting. Clients can implement plugins to detect short-term blob censoring. As a blob publisher, it is crucial to recognize that a blob might not be included despite paying a sufficient fee. In such cases, the fee may need to be increased over time.

Validator delaying block proposal for a higher block reward

It has been publicly acknowledged that validators are delaying block proposals to compete for higher rewards based on the belief that more time leads to more MEV and greater rewards. Recent developments have introduced counterpoints explaining why delaying block proposals is detrimental and contributes to poor chain health. The rest of this post will discuss

how post EIP4844 blob transactions further complicates this issue.

Why does the introduction of blobs further complicate this issue? To understand this, let's first examine the consensus specification. The is_data_available

definition in the consensus specs states:

The block MUST NOT be considered valid until all valid Blobs have been downloaded. Blocks that have been previously validated as available SHOULD be considered available even if the associated Blobs have subsequently been pruned.

(Source)

Let's consider why blobs would arrive after the block. This is due to two main reasons:

1. A blob is usually larger than a block.
2. The number of blobs (maximum of 6) exceeds the number of blocks.

Clients will often have to wait sometime after the arrival of a block before they can accurately deem it valid if the block includes blobs.

[

Screenshot 2023-12-30 at 4.44.46 PM

1842×1124 244 KB

](https://ethresear.ch/uploads/default/original/2X/f/febff28ed83dd535f926874c88a5d830f463c04c.png)

Diagram:

1. The block, with six blobs, was released together after the slot began.
2. Subsequently, the block, along with the blobs, is received from peers at different times.
3. The fourth blob is the last to arrive.
4. Peers process the block and wait for the arrival of the fourth blob before considering the block valid.
5. The block will likely become the head if received before the attestation cutoff, which occurs 4 seconds into the slot.

Validator and builder (independent) delay block

Two opposing forces are at play:

- Validators seek to delay the block for higher MEV.
- Builders aim to prevent the block from being reorg.

In this scenario, if the validator requests the block late, the builder might choose not to include the blob transaction, anticipating that the block could be reorg later. (H/T @potuz) In this situation, the transaction fee for the blob may begin to rise. Consequently, the validator may realize that not delaying the block is more profitable than delaying it, as the increased fee from the blob transaction compensates for the potential gains from delay. This scenario leads to the creation of an oscillating feedback loop. As validators adjust their strategies between delaying and not delaying the block based on profitability, and builders respond to these changes, the dynamics between transaction fees and block timing continue to fluctuate.

[

Screenshot 2024-01-02 at 3.17.41 PM

966×426 19.2 KB

](https://ethresear.ch/uploads/default/original/2X/8/813213036ac001711c56256b1d3b27c5bac0f2f9.png)

Validator and builder (vertically-integrated) delay block

A more refined prediction in the vertically integrated realm of builders and validators is that validators delaying block proposals could perceive the waiting period as a chance to justify the delay. Ideally, they would aim to time the delay of the block proposal to offset the variance in arrival times between the block and the slowest-arriving blob. This strategy ensures that the block and the blobs arrive simultaneously for their peers. It will consistently subscribe to the maximum number of

peers possible on the blob subnet to minimize network hops. Some may even use heuristics to dynamically figure out how long to delay, based on the propagation of the last few blocks and blobs.

[

Screenshot 2023-12-30 at 4.48.33 PM

1848×1178 257 KB

](https://ethresear.ch/uploads/default/original/2X/a/ab6fe3b3135632ad4a3edb5f7487fa6ce93e4230.png)

Diagram:

1. The proposer picks six blobs from the mempool and broadcasts them at the start of the slot, preparing everything for the block except the execution data.
2. Upon seeing the first blob from the network (random heuristic), the proposer prepares the execution data, signs it, and then broadcasts the block.
3. The block and the final blob are received simultaneously, eliminating the need for peers to wait.

Caveat:

Our current understanding of blob arrival delay is based exclusively on data from previous devnets and the Goerli shadowfork. The network topology of this setup is distinct from that of other testnets or the mainnet. Continual monitoring is essential, as a minimal delay might render our present conclusions obsolete.

Possible short-term solutions

1. Implement monitoring infra to track the duration a blob transaction remains in the mempool before its inclusion in a block. Observe if specific reorg behavior is associated with blob transactions where the submitter is a certain blob poster but not others. Alert the community on abnormal behaviors.
2. Collaborate with MEV-Boost relayers to implement a getHeader

cutoff time, which prevents proposers from using a builder to construct a late block. However, this measure may be ineffective if the builder and validators are vertically integrated or if a defect exists in one of the many relayers. (h/t original timing game post)

Conclusion

Validators are currently engaged in timing games, but their behavior may become more unpredictable post EIP4844, especially considering that blob transactions are likely to introduce additional delays before a block is validated and becomes the head.

Validators have several strategies they can adopt:

1. Adhere to an honest strategy, which involves including blob transactions as per the default behavior of the stock software.
2. Pursue a rational profit-maximizing strategy by including only those blob transactions that meet their profit margins, and pay sufficient fees to mitigate the risks of reorg.
3. Engage in an additional game by including blob transactions and exploiting the extra delay to propose a block late, thereby extracting more MEV.

How these strategies will play out in practice remains to be seen. It will require enhanced infra monitoring capabilities to observe their effects over the upcoming testnets and eventually on the mainnet.