

Thanks to [@simbro](#) for reviewing

## Abstract

For this post, block size refers to the number of serialised bytes in a block. Currently, the average block size is over 100k <https://etherscan.io/chart/blocksize>. Note that we are talking about bytes, not gas limit.

Every block builder is motivated to submit a block to the PBS auction as late as possible. The more time a block builder has, the more time they have to accumulate transactions and, therefore, priority fees. For the purpose of this discussion, we assume no MEV is at play.

## Discrete Blocks and Latency Games

Apologies if this is old news, but it's worth reiterating. The role of a block builder is multifaceted and requires proficiency in several infrastructure tasks.

1. Block builders must be able to access transactions; solo validators often miss out on priority fees because they are not well connected to the public mempool. A block builder with good connectivity to the mempool will likely win more blocks. They may even partner with wallets and other transaction originators to fast-track mempool transactions into their builder pipeline.
2. They must have good networking connectivity with the auctioneer, aka relay. The lower the latency between them and the auctioneer, the more time they have to build blocks.
3. Because the block builder is privileged, they can offer value-added features that no other entity can offer. Namely, revert protection through `eth_sendBundle`. The builder who can build a block the fastest whilst protecting private order flow from reverts will win more blocks (once again, we assume no mev).

Because blocks are discrete periods, pressure is applied to all parts of the stack towards the end of the block. Consequently, a block builder will do what it can to increase the amount of time it has to focus on its core activity, building the most profitable block.

## Observations

Block builders will likely submit multiple bids using multiple strategies. Sometimes, the bids for smaller blocks are received in time, while those for larger blocks are not. This is simply because larger blocks are slower. Consequently, transactions that could be included in a block are not being picked up.

## Hypothetical Scenario

A block builder has 100 transactions in their local mempool, totalling 0.5 eth in priority fees. The network is silent, and no other transactions are entering the mempool. The block builder submits the block (block a) to the auction. Near the very end of the block, another transaction enters the mempool with a whopping 1 eth in priority fees. The block builder now submits two more bids at the same time.

block b - containing our single juicy priority fee transaction for 1 eth.

block c - containing 101 transactions with all the transactions we have totalling 1.5 eth.

Both bids are now higher than the previous bid. One of three scenarios now stands:

1. The original bid wins as neither subsequent bid was reached in time by the auction before the deadline.
2. The small block reaches the auction before the deadline, pushing out the previously submitted block of 100 transactions.
3. The big block reaches in time, and all transactions are included.

It is easy to imagine an interplay between latency, block size and priority fees that are entirely opaque to users and sophisticated actors.

## Real-world example

JetBuilder built <https://etherscan.io/block/19598122> and only used 12% of the block space available, paying ~0.15 eth for the block. We observed them missing at least 40 transactions that could have been included in that block. The example transactions were in the mempool for at least five blocks (thanks to <https://www.etherbase.org>). They all landed on-chain in either the next block or the one after.

# Proposal

We should have some floor usage in gas terms to prevent transactions from bullying other transactions out of a block.

The gas floor target could be calculated in many ways, such as a predefined fixed target, half the gas limit, or a dynamic adjustment based on previous consumption (similar to 1559). It doesn't need to be elegant or exact; it just needs to be something to incentivise block builders to utilise block space.

The penalty for not 'filling' the block would be something like  $\text{gas target missed} * \text{base fee}$

. This is the same price as putting a transaction in the block, except the block builder doesn't get priority fees. Theoretically, a block builder could make a transaction with themselves, but the result is the same.

We are simply putting a price on what the network perceives as the underutilisation of block space.