I like the general idea but I'm concerned about a particular griefing vector that utilizes the interplay between p2p clients' DDoS protection, the minimum maxFeePerGas, and fulfilling a Summary. I can't think of a way to solve it but perhaps someone else can.

For context, in Geth, a transaction's gasPrice has to be >10% higher than a previous transaction with the same EOA / nonce in order to overwrite it. Otherwise, it is rejected and goes unpropagated.

The rationale is straightforward: by allowing txs to overwrite themselves with smaller gasPrice increments, priority gas auctions will place larger strain on the network due to a higher amount of transactions propagating and then being overwritten.

An example of the attack vector is thus:

1. Send a tx (#1

) with a maxFeePerGas of baseFee + 10% to a set of nodes (A)

1. Send a tx (#2

) with same nonce, same eoa, and a maxFeePerGas of baseFee + 13% to a different set of nodes (B)

1. Nodes in set A who have received tx #1

will reject tx #2

. Nodes in set B will accept tx #2

and could flag it for inclusion.

1. If a node in set A needs to make a block with tx#2, it won't have access to the tx because it would have rejected it.

2. And tx#1 may be invalid due to base fee rising, resulting in the summary being unfulfilled.

In this example, it's not hard to imagine that the validator could still request the transaction, but in reality it would be far trickier. This form of "gasPrice-induced intentional censorship" is a common method that MEV bots use to identify validators and each other - they send a unique tx to each

of their node's peers.

If all of the validators peers only ever saw different variants of the baseFee +10% txs, the validator may have no way to figure out who to request a needed baseFee +13% tx from.

Reducing the minimum gasPrice increment (which is not a protocol-level value) feels like it'd be trading one problem for another, and it would only reduce the odds of this situation developing organically - it couldn't prevent it from happening intentionally should the builder of block N wish to "grief" the builder of block "n+1."