Prior Discussions

@matt : An Optimistic Generic Gas Market Executor for Phase 2

@vbuterin: Phase 2 Proposal 2

@vbuterin: One fee market EE to rule them all

Background

To give credence to this discussion, we need a good, consolidated summary on the background of the fee market discussion for eth 2.

First Proposal

The first iteration looked similar to the following:

[

Relayer%20Diagram

1652×528 17.9 KB

](https://ethresear.ch/uploads/default/original/2X/b/bd7e8f34af53ca1bdf0fbfa8a45d1b228377c58a.png)

A user would submit a transaction to a relayer

. A relayer

would be responsible for adding witness data in a stateless system and collect a proposed block of transactions. In this system, transaction fees would be paid directly to the relayer

. The relayer

would then offer a flat fee, F

to the block producer to publish its block.

This system introduced a number of issues. To summarize:

- A JMRS scheme would need to be introduced - Optimised proposal commitment scheme.
- This adds significant implementation complexity and introduces other incentive questions.
- This scheme assumes only one relayer

is awarded per execution environment which essentially strengthens the relay market as a centralization risk.

- This adds significant implementation complexity and introduces other incentive questions.
- This scheme assumes only one relayer

is awarded per execution environment which essentially strengthens the relay market as a centralization risk.

- A number of asymmetries introduced. I summarize below, but please read the post by@benjaminion here for a detailed analysis: Exploring the proposer/collator split.
- Depending on the market around the flat fee offered from relayers, a block producer may be incentivized to be its own relayer and pick transactions randomly without maintaining full state.
- Malicious relayers can attack an entire block by offering a high fee.

- Depending on the market around the flat fee offered from relayers, a block producer may be incentivized to be its own relayer and pick transactions randomly without maintaining full state.
- Malicious relayers can attack an entire block by offering a high fee.

- Elegant POS is throttled by the relay market's cheapest computation, cheapest hardware approach.
- Relayers with low cost hardware and computation will have an advantage on the lowest fee. This could lead to a cartel

of relayers that own the relay market and can censor transactions at will.

- Relayers with low cost hardware and computation will have an advantage on the lowest fee. This could lead to a cartel of relayers that own the relay market and can censor transactions at will.

The original rationale behind the relay market was to manage witness/state data in a stateless system and provide incentives for maintaining state. However, it introduced a number of issues and similar incentives can be introduced in other ways (read further).

Second Proposal

After time, @vbuterin introduced a new proposal: One fee market EE to rule them all It was a significant improvement and removed many concerns listed above.

The proposal introduced a universal fee market or execution environment. In this fee market, all the execution environments could own a balance. Transaction fees would pay into the balance of the execution environment. At the end of a set of transactions, a receipt

would be generated. This receipt

could be used by the block producer to claim funds from the fee market EE and withdraw the funds to their validator account on the beacon chain or to another EE (depending on what the block producer wanted to do

).

Some major improvements in this proposal:

- Fees are paid to the EE and then to the block producer. There is no longer a relay market with a potential cartel which may act as a throttling function. Relayers exist in a new capacity

. Lets call them state providers for now.

- Because it is in the interest of all validators to receive the highest fee transactions, the mempool would be introduced again, encouraging a more distributed system.

This process and proposal requires additional steps of complexity and introduces a few questions:

- Complexity in the multi-step process of moving funds from one EE to another through the beacon chain.
- Who has the incentive to maintain state and provide witness or state data?

More Light Client Servers - State Providers

As a result, this proposal introduced even more demand for the class of actors, light client servers

. These light client servers

or state providers

would be incentivized to maintain state. @vbuterin describes further here: One fee market EE to rule them all He also suggests an elegant system where transaction senders can maintain a state channel with the state provider

and pay to add state/witness data to their transaction. Assuming tooling around a light client server is simple and easily runnable as a daemon, there should be a fairly distributed network of state providers

.

The writeup also suggests state providers

would package transactions together before sending them to the block proposers. The basis for this approach was to have a package that generates one payment receipt that can be claimed later, see One fee market EE to rule them all It also still refers to this actor as a relayer

.

In reality, there no longer needs to be a concept of a relayer

as it was originally. They would just broadcast the tx to a mempool now and add the necessary witness data. Additionally, the state providers

do not need to create transaction packages. They already received their payment through a state channel. From there, they

can submit each transaction to a network of mempool

s and be done. In turn, the block producer can create transaction packages as 1:1 to the number of EEs the set of transactions represent. 3 EEs would result in 3 transaction packages and 3 receipts to claim.

Iterating Further - Sync Calls Between EEs

We can continue to simplify the proposal if we support synchronous calls between EEs within the same shard. In this case, we no longer need receipts to be claimed through this multi-step process. The entry point to a set of transactions would be directly through the fee market EE. This fee market EE could iterate through each of the transactions and delegate its call through the proper EE each transaction is associated with. Through each call, the funds can be transferred directly to the block producer.

Additionally, state channel payments to the state providers

are no longer needed. The user may sign their transaction with a state provider

id and therefore payment can be transferred synchronously.

In order to determine if synchronous calls between EEs within the same shard are possible, benchmarking within scout is needed.

Another Step Forward - Generic Asset EE

We can simplify our system as a whole even further assuming we have sync calls between EEs. We may have one EE dedicated to the balances of accounts across all EEs. It is one shared EE to rule the token (ERC20) and beacon eth balance of every EE and every account within the EEs. @matt shared a quick concept: One fee market EE to rule them all

Additionally, this approach may support gas payments in tokens other than pegged beacon eth.

Summary and Conclusion

In closing, the main points are as follows:

- We no longer need relayers

in the same capacity. Lets call them state provider

s as part of a light client server

for now.

- Transaction packages (if needed in a non-sync inter-ee system) can be 1:1 between the number of EEs represented in a block. They would be packaged by the block proposer.
- Mempools should be introduced again.
- Synchronous calls between EEs make everything simpler (but we need to benchmark if it can work).

Questions

- How do we ensure the network of state provider

s is distributed and open vs a centralization risk?

- Since opening a state channel requires a tx and a fee, will this system encourage users to rely on only a small set of infrastructure providers to add state?
- What type of additional layers of frontrunning does this system introduce?
- Is the mempool

a part of the phase 1 or phase 2 specification?