Prior Discussions

State Providers, Relayers - Bring Back the Mempool

@adlerjohn: Relay Networks and Fee Markets in Eth 2.0

@vbuterin: Eth2 shard chain simplification proposal

Background

The latest phase 0 & 1 proposal addresses a number of the concerns posted within the writeup,State Providers, Relayers - Bring Back the Mempool. In the new proposal, validators, EEs and block producers may transfer ETH between shards within a 1 block latency. This behavior is enshrined and gives a block producer the ability to deterministically run a transaction and confirm payment without having to trust the mechanics of an EE. For a more detailed description, read the “basic operating system” section in the proposal.

Comparison to Eth 1.x Research

The Eth1.x research group is actively working on transitioning eth1 into a stateless model. However, in the current transition, the miner (analogue to a block producer) holds the account state. In eth2, the block producers and validators would not

be assumed to hold state. This difference introduces a number of questions around the stateless mechanics of eth2 that we do not have to think about currently within the effort around the eth1 stateless transition. Eventually, eth1 will likely

transition to a model where the miners are not assumed to be stateful. This would need to occur before theeth1 -> eth2 switchover.

Current Open Questions

- In eth1, transactions are of course propagated through a mempool. In eth2, alternate models could be explored since the block producer is predictable, although there are benefits to pursuing a mempool model from eth1. In a unicast model, the system is highly susceptible to DoS attacks. Is a mempool the right direction?

- Validators may likely keep some

popular segments of the state tree and have basic cacheing capabilities on top of what is provided by cacheing through the EE layer, described here. There could be incentives or economic benefits to doing this depending on how state provider incentives operate. An analysis on this piece could be interesting.

- If a mempool is used, it will need to actively refresh its witnesses (stale transactions) as it prepares transactions for each block. This means EEs need to be deployed with a refresh script and we should benchmark this approach. Also, the mempool will need to refresh witnesses for multiple accumulators since it would support transactions for multiple EEs.

- Prior to preparing a block, the block producer will need to access a merge function from each EE it includes a transaction for. This merge function will give the instructions on merging the individual transaction proofs into a multiproof. We should also look at the mechanics of this script which would likely need to be deployed alongside a new EE.

- Should transactions from state providers (or the network) already be packaged in a multiproof? Would prepackaged multiproofs from state providers prevent block producers from taking standalone transactions due to additional complexities? Could this still skew towards some centralized behaviors as described in State Providers, Relayers - Bring Back the Mempool

- Do we foresee issues around bombs? As an example, how do we manage witnesses that go extremely

deep and take a non-significant amount of execution time only to fail near the end as it ismissing access to a particular region of state. We cannot charge the user in this case since it may not be malicious. If a contract dynamically accesses an account, then a prior transaction in the block may change the account it accesses. In this case, the transaction would no longer contain its needed witness. Distinguishing malicious behavior is quite difficult in cases such as this. This makes the mempool highly susceptible

to a DoS attack.

- Miners are incentivized to choose transactions where the accounts are close to each other within the state tree (therefore decreasing the overall size of the multiproof). They are also likely incentivized to prioritize accounts which are frequently used. Do we see any issues or concerns with this?

- How do we price a user for their witness data when multiple transactions use the same witness in a multiproof. For example, if 3 transactions share the same witness, does the price get split in 3?

Questions Around State Provider Incentives

- What is the best material currently on state provider or light client server incentives?
- Do we open a network of payment channels to provide micropayments to state providers or light client servers? @vbuterin originally brought up the state channel approach here
- If a developer is running local tooling such as ethers or web3.js, is it strange that the developer should be charged for test runs? How about estimating gas in a wallet?
- What is the viability of this payment network? Has anyone evaluated how this would look in practice? This would be a fairly large operation and we should be thinking about this soon.
- Since there is friction opening or switching between payment channels, will we skew towards a couple wallets/providers owning this due to network effects therefore solidifying services like infura as a centralized party?

Conclusion

There remains a number of open questions around stateless networks in eth2 - in addition to what the eth1.x group is investigating. It would be valuable to get more involvement and more eyes/criticism on these pieces.