Co-authored with Max Resnick

Expands on an initial forum post on Ethereum Research for the context of Eigenlayer

MEV and the Proposer's Monopoly on Inclusion

Modern blockchains operate on a leader-based system where a rotating proposer holds the power to decide which transactions are included in each block and what order to include them in. Some proposers collude with MEV-searchers to maximize the rents that they can extract from this temporary monopoly (e.g. MEV boost).

This collusion can warp the equilibria of on-chain mechanisms by allowing some users to pay to censor other users' transactions. Auctions – a key tool in on-chain mechanism design – are particularly susceptible to manipulation through censorship See Censorship Resistance in On-Chain Auctions for more details.

The proposer's monopoly on inclusion is the weak link, which allows for cheap censorship. In order to improve the censorship resistance of a chain we need to destroy this monopoly by allowing multiple concurrent block proposers (MCBP). In particular, we would like to expand the set of nodes who are able to contribute transactions in each block. Multiplicity is a practical gadget that allows non-leader nodes to include additional transactions into a block.

Multiplicity: A Practical Gadget for Multiple Concurrent Proposers

Informally, In the first stage every non-byzantine validator on the committee sends a signed special bundle of transactions, to the leader. In order to construct a valid block, the leader must include at least 2/3 (stake weighted) of these bundles. Therefore the leader cannot construct a valid block that omits a transaction which was included in >1/3 (stake weighted) of the special bundles.

More formally: a proposed block is only valid if the leader includes a sufficient quorum (defined either by stake-weight or number) of validator-signed special bundles of transactions.

The gadget adds the additional steps to proof-of-stake protocols:

1. Each validator constructs a special bundle of transactions from their local view of the mempool. They sign this and send it to the leader.
2. Based on payloads received from other validators, the leader creates, signs and broadcasts a block proposal containing at least 2/3 (stake weighted) of these special bundles. .
3. When determining the validity of the leader's proposal, validators check that sufficiently many special bundles are included in the block

3a. If the block contains a quorum of payloads the block is sufficient and consensus proceeds normally.

3b. If the block does not contain a quorum of payloads it is considered invalid and a new round of consensus starts the same way it would if a block contained a transaction with an invalid signature.

Conditional Tipping Logic to Incentivize Inclusion

Conditional tipping rules where the transaction tip is only split among the proposers who include a transaction can be used to improve censorship resistance even further. Conditional tipping logic increases the cost of censorship by making colluding equilibria less stable.

For example, say there are three validators, a transaction with a tip of $5 and a bribing agent who values censoring the transaction at $10. If there is a single leader, it is in the leader's best interest to take the bribe for > $5 in exchange for censoring the transaction. When more leaders are added, the bribing agent must bribe each of the leaders, eventually it becomes too expensive to bribe everyone and the transaction gets through. See Censorship Resistance in On-Chain Auctions for more details.

Sequencing for L2s

The mechanism for multiple concurrent block proposers can also be used for decentralized rollup sequencing. Since rollups inherit the censorship resistance of their base layer, current "fork choice sequencing rules" that rely solely on block inclusion are not economically compatible with the block leader's incentives. This is because the block leader can always make strictly more money by censoring all of the blocks for a given rollup other than their own with the minimum bid or cost - for reasons described in Censorship Resistance in On-Chain Auctions the application has no power to capture any surplus for itself or it's users.

Through base layer censorship resistance provided by Multiplicity, L2s can run decentralized sequencing rules that rely on fork choice rules without needing to worry about leader censorship. This also navigates around the lack of decentralized sequencing infrastructure which is currently an issue for Ethereum L2s.

If the rollup prefers using a secondary sequencing mechanism, Multiplicity using proof-of-stake leader selection serves as a way to not only decentralize leader selection (as in currently proposed designs) but to also decentralize individual block production. This ensures that the surplus of value created in a block can be returned to the application and its users instead of the leader by running censorship resistant blockspace auctions.

Not Just Auctions but Arbitrary Ordering Rules

A blockchain's state machine can impose arbitrary ordering rules for transaction execution. This is because the ordering of transaction inclusion does not necessarily need to be the order of transaction execution.

There are many competing opinions and mechanisms for ordering how transactions should be executed. Some mechanisms focus on whole block auctions such as MEV-Boost++ while others propose first-come-first serve mechanisms such as the receive-order fairness literature. By focusing only on guaranteeing inclusion we can build either of these ordering mechanisms into the state machine. For example, you can translate the transaction bundles included by validators into a DAG, where arrows represent the order they were received by the validator. With the censorship resistant properties from Multiplicity, they can be treated as "votes" on the canonical transaction ordering.

Notes

Duality Labs is building multiplicity for an on-chain MEV auction that redistributes value back to liquidity providers to reduce LVR. The v1 is largely inspired by ABCI++ and being built with custom add-ons to Tendermint and ABCI 1.0 (PrepareProposal and ProcessProposal), but is generalizable to most leader-based PoS consensus algorithms through the previously described steps. We plan on open sourcing a repo in Q2 2023. In the meantime feel free to reach out for partnerships and collaboration on twitter: @PossibltyResult

.

Big shoutout to Zaki Manian for leading us down the right path for a practical implementation of multiple concurrent block proposers.