Shared Mempool

Mempool

Pending user transactions are stored in the mempool of each validator until they are included in a finalized block. Pending transactions are shared to other validator mempools by erasure-coding the transaction and then communicating over a broadcast tree for efficiency.

Transaction hashing

MonadBFT is an efficient means of coming to agreement about an arbitrary payload. However, block propagation is still a significant bottleneck; for example, a block with 10,000 transactions with 500 byte transactions will be 5 MB; blocks of this size would put undue bandwidth requirements on validator nodes.

To alleviate this issue, block proposals refer to transactions by hash only - a significant savings, as hashes are 32 bytes. Because of this, all validator mempools need to have the transactions in their own mempool when voting on proposals and committing blocks. Transactions that are submitted to a validator's mempool are shared with other validator mempools by erasure-coding the transaction and then communicating over a broadcast tree for efficiency.

Previous MonadBFT Next Deferred Execution Last updated5 months ago

On this page * Mempool * Transaction hashing