EIP-1559 Support

EIP-1559 is a new upgrade to the Ethereum network that changes how you pay for transactions. It introduces a base fee that varies depending on the network demand, and a priority fee that you can set to get faster confirmation. The base fee is burned, while the priority fee goes to the miner who includes your transaction in a block. Flashbots, starting frommev-geth v1.10.5-mev-0.3.0 , has integrated support for EIP-1559 transactions.

While users of the legacy transaction type don't need to make any configuration changes, they should be aware that it's now mandatory to include agasPrice that is at least equal to the base fee. Coinbase transfer can still be used to incentivize faster inclusion, but it cannot be used to bypass the base fee requirement.

Legacy Transaction example

Below is an example of signing bundles with a legacy transaction:

const signedTransactions =

await flashbotsProvider . signBundle ([{ signer : authSigner , transaction :

{ to :

"0xf1a54b075fb71768ac31b33fd7c61ad8f9f7dd18" , gasPrice :

10 , gasLimit :

33000 , chainId :

5 , value :

0 , } , } ,]) ; The full amount ofgasPrice will be consumed first to clear the base fee, and the remaining will be used as priority fee.

EIP-1559 Transaction example

Below is an example of signing bundles with EIP-1559 transactions (note:chainId is a required attribute for 1559 or type2 transaction):

const block =

await provider . getBlock ("latest") ; const maxBaseFeeInFutureBlock = FlashbotsBundleProvider . getMaxBaseFeeInFutureBlock (block . baseFeePerGas ,

1) ; const priorityFee =

BigNumber . from (2) . pow (9) ; const signedTransactions =

await flashbotsProvider . signBundle ([{ signer : authSigner , transaction :

{ to :

"0xf1a54b075fb71768ac31b33fd7c61ad8f9f7dd18" , type :

2 , maxFeePerGas : priorityFee . add (maxBaseFeeInFutureBlock) , maxPriorityFeePerGas : priorityFee , gasLimit :

33000 , chainId :

5 , value :

0 , } , } ,]) ; Here the priorityFee is set to 2 Gwei, and the maxFeePerGas is set to be exactly equal to the max base fee in the next block plus the priority fee.

FAQ

Can a transaction specifymaxFeePerGas=0

?

No, all transactions must have maxFeePerGas greater than or equal toblock.baseFeePerGas , or they are not eligible for inclusion in a block.

Can a transaction specifymaxPriorityFeePerGas=0

Absolutely, although the builder will need some incentive to include this transaction. With a Flashbots bundle, you can incentivize a builder/validator to include your transactions withblock.coinbase.transfer() paymentsOR viamaxPriorityFeePerGas . You can also use both at the same time; the incentive is cumulative.

Will reverting transactions still be discarded?

Flashbots still uses the same reverting transactions logic after EIP-1559: unless specified inrevertingTxHashes ineth_sendBundle , a transaction that reverts invalidates an entire bundle and will not appear on chain.

However, with the new requirement for searchers to provide a gas price that meets thebase fee for each transaction, successfully included bundle transactions may end up in the mempool following block re-organizations. IfgasPrice=0 is used, re-organized transactions are swiftly dropped from the gossip network, making it unlikely for them to appear in a future block unless reintroduced by another searcher. Transactions that pay at least the base fee will remain in the mempool and have a higher chance of appearing in future blocks, potentially conflicting with expectations regarding reverting transactions.

How can I send a transaction from an account with 0 ETH, like one with a malicioussweeper

running against it?

We have a working example of how to accomplish this in ourSponsored Transaction Github Repository , which has been updated to work with EIP-1559.

Where can I learn more about EIP-1559?

EIP-1559 Hackmd Cheat Sheet Edit this page Last updatedonJan 30, 2024 Previous Testnets Next Bundle Inclusion Troubleshooting