Gas Fee Refunds

Introduction

Searchers and private transaction API users are automatically eligible to receive gas fee refunds. If Flashbots can include a bundle on chain for a lower price, you are eligible to receive a refund.

Gas fee refunds do not change how bundles are executed and searchers do not need to make any changes to be eligible for them.

Where do refunds come from

Gas fee refunds include both priority fees and coinbase transfers.

In an optimal case, searchers are refunded the difference between their bid and the bid of the next-best bundle or transaction targeting the same state. Ie. the refund effectively results in the searcher paying the second price. In practice, searchers will receive some fraction of this amount depending on how much profit the Flashbots builder makes.

Which bundles receive refunds

Flashbots provides refunds for bundles in blocks landed by the Flashbots block builder. Whether a bundle receives a refund depends on a few factors that vary from block to block:

- How much network congestion and competition there was
- Whether the Flashbots builder made a profit and how much
- How much the specific bundle contributed to the value of the block
- If the bundle was sent directly to Flashbots, or shared with other block builders by the searcher

Note that transactions seen in the public mempool are excluded and bundles containing only public mempool transactions do not receive refunds.

How to maximize both refunds and speed

Transactions which are sent directly to the Flashbots Builder via the Bundle Relay, and not multiplexedby the searcher to other block builders, are likely to receive higher refunds. This is because they increase the profit of the Flashbots builder which is used to provide refunds.

The Flashbots block builder does not land 100% of blocks. In order to land bundles in all blocks, searchers can ask Flashbots to share their bundles with other block builders in cases where the Flashbots builder does not win a block. Flashbots will automatically share with all specified builders on the searcher's behalf.

Smart multiplexing

To share bundles with other builders, add thebuilders field to youreth_sendBundle request. Thebuilders field accepts a list of strings which correspond to the "name" tags ofregistered builders .

Alleth_sendBundle requests are shared with the Flashbots builder. They are multiplexed to other block builders at the end of the slot if the Flashbots builder determines it will not win that block.

For example:

{ "jsonrpc": "2.0", "id": 1, "method": "eth_sendBundle", "params": [{ "txs": ["0x123abc...", "0x456def..."], "blockNumber": "0xb63dcd", "minTimestamp": 0, "maxTimestamp": 1615920932 "builders": ["builder0x69", "beaverbuild.org"] }] } Searchers can also usemev_sendBundle to multiplex bundles if they prefer. Though this method is more complex and not necessary for gas fee refunds.

Note: Smart multiplexing has a 1% rate of false positives, meaning that in 1% of MEV-Boost blocks there is a risk that searcher bundles will not be landed.

Bundle stats for multiplexed bundles

To view bundle stats on multiplexedeth_sendBundle requests, use theflashbots_getSbundleStats API. You will see a new "smart" field in the response to multiplexedeth_sendBundle which indicates that the sbundle stats endpoint should be used.

How are refunds calculated

The Flashbots builder uses a refund rule to retroactively calculate refunds for all bundles landed in its blocks.

The refund rule aims to have bundles make the minimum net payment so that bidding optimally is as straight forward as possible. We do this by measuring the contribution of bundles above the other bundles the builder has received, and refunding as much of that as possible.

Bundles sent by the same signer will be treated as non-competitive.

The Flat Tax Rule

- B
- (
- T
-)
- B(T)
- B
- (
- T
-)
- is the most profitable block produced from bundles inT
- T
- T
- .
- v
- (
- T
-)
- v(T)
- v
- (
- T
-)
- is the value ofB
- (
- T
-)
- B(T)
- B
- (
- T
-)
- .
- b
- i
- (
- T
-)
- b_i(T)
- b
- i
-
- (
- T
-)
- is the payment of all bundles sent by identityi
- i
- i
- if blockB
- (
- T
-)
- B(T)
- B
- (
- T
-)
- is realized.
- μ
- i

- (
- T
-)
- =
- min
-
- {
- b
- i
- (
- T
-)
- ,
- v
- (
- T
-)
- −
- v
- (
- T
- \
- {
- i
- }
-)
- }
- \mu_i(T) = \min{b_i(T), v(T) - v(T \setminus {i})}
- μ
- i
-
- (
- T
-)
- =
- min
- {
- b
- i
-
- (
- T
-)
- ,
- v
- (
- T
-)
- −
- v
- (
- T
- \
- {
- i
- }))}
- is the marginal contribution of all bundles sent by identityi
- i
- i
- ifB
- (
- T
-)
- B(T)
- B
- (
- T
-)

- is realized. We bound the marginal contribution so that the net payment can't be negative.
- c
- c
- c
- is the amount the builder pays to the proposer to win the block.

ϕ i (T , c) = μ i (T) ∑ j μ j (T) min { v (B (T)) − c , ∑ j μ j (T) } \phi_i(T, c) = \frac{\mu_i(T)}{\sum_j \mu_j(T)} \min{v(B(T)) - c, \sum_j \mu_j(T)} ϕ i (T ,

c)

= ∑ j

μ j (T)

μ i (T)

min { v (B (T))

− c ,

j ∑

μ j (T)} So the net payment per identity (assuming it's included) isp i (T) = b i (B (T)) − ϕ i (T , c) p_i(T) = b_i(B(T)) - \phi_i(T, c) p i (T)

= b i (B (T))

− ϕ i (T ,

c) .

Notice that if the block generates enough value after paying the proposer, everyone should be refunded their contribution, meaning everyone pays the minimum they need to pay to beat competition.

Identity constraint

To avoid the rule being gamed by submitting bundles from multiple identities, we impose an additional constraint that no set of identities can receive in total more refunds than they contribute to the block.

For each set of identitiesI I I we define

μ I (T) = min { ∑ i ∈ I b i (T) , v (T) − v (T ∖ I) } , \mu_I(T) = \min{\sum_{i\in I} b_i(T), v(T) - v(T \setminus I)}, μ I (T)

= min { i ∈ I ∑

b i (T) ,

v (T)

− v (T

∖ I)} , to be the joint marginal contribution of the identities inI I I to the block. Then we choose rebates that are minimally different from the flat-tax rule subject to the constraint that they don't rebate a set of bundles more in total than its joint marginal contribution. This means the vector of rebatesψ (T , c) \psi(T, c) ψ (T ,

c) solves

min r ∈ R + n ∑ i (r i − ϕ i (T , c)) 2 \min_{r\in\mathbb{R}^n_+} \sum_i (r_i - \phi_i(T, c))^2 r ∈ R + n min

i ∑ (r i

− ϕ i (T ,

c)) 2 subject to ∑ i ∈ I r i ≤ μ I (T) for each I ⊆ B (T) , \text{subject to} \sum_{i\in I} r_i \leq \mu_I(T) \text{ for each } I \subseteq B(T), subject to

i ∈ I ∑

r i

≤ μ I (T) for each I

⊆ B (T) , ∑ i r i ≤ v (T) – c \sum_i r_i \leq v(T) - c i ∑

r i

≤ v (T)

– c whereϕ (T , c) \phi(T, c) ϕ (T ,

c) are the orginal flat-tax rebates as defined above.

Who receives refunds

By default, the refund recipient is the signer used on theeth_sendBundle ,mev_sendBundle , oreth_sendPrivateTransaction request. You can delegate your recipient to a different address using theflashbots_setFeeRefundRecipient API.

How to track refunds

Refunds are tracked from a start date of July 8, 2024. Users will be able to view refund amounts via an upcoming APEdit this page Last updatedonSep 16, 2024 Previous Send transaction/bundle to multiple builders Next coinbase.transfer()