L1 Pricing

ArbOS dynamically prices L1 gas, with the price adjusting to ensure that the amount collected in L1 gas fees is as close as possible to the costs that must be covered, over time.

L1 fee collection

A transaction is charged for L1 gas if and only if it arrived as part of a sequencer batch. This means that someone would have paid for L1 gas to post the transaction on the L1 chain.

The estimated cost of posting a transaction on L1 is the product of the transaction's estimated size, and the current L1 Gas Basefee. This estimated cost is divided by the current L2 gas basefee to obtain the amount of L2 gas that corresponds to the L1 operation (more information about this can be found in this article).

The estimated size is measured in L1 gas and is calculated as follows: first, compress the transaction's data using the brotli-zero algorithm, then multiply the size of the result by 16. (16 is because L1 charges 16 gas per byte. L1 charges less for bytes that are zero, but that doesn't make sense here.) Brotli-zero is used in order to reward users for posting transactions that are compressible. Ideally we would like to reward for posting transactions that contribute to the compressibility (using the brotli compressor) of the entire batch, but that is a difficult notion to define and in any case would be too expensive to compute at L2. Brotli-zero is an approximation that is cheap enough to compute.

L1 gas fee funds that are collected from transactions are transferred to a special L1PricerFundsPool account, so that account's balance represents the amount of funds that have been collected and are available to pay for costs.

The L1 pricer also records the total number of "data units" (the sum of the estimated sizes, after multiplying by 16) that have been received.

L1 costs

There are two types of L1 costs: batch posting costs, and rewards.

Batch posting costs reflect the actual cost a batch poster pays to post batch data on L1. Whenever a batch is posted, the L1 contract that records the batch will send a special "batch posting report" message to L2 ArbOS, reporting who paid for the batch and what the L1 basefee was at the time. This message is placed in the chain's delayed inbox, so it will be delivered to L2 ArbOS after some delay.

When a batch posting report message arrives at L2, ArbOS computes the cost of the referenced batch by multiplying the reported basefee by the batch's data cost. (ArbOS retrieves the batch's data from its inbox state, and computes the L1 gas that the batch would have used by counting the number of zero bytes and non-zero bytes in the batch.) The resulting cost is recorded by the pricer as funds due to the party who is reported to have submitted the batch.

The second type of L1 cost is an optional (per chain) per-unit reward for handling transaction calldata. In general the reward might be paid to the sequencer, or to members of the Data Availability Committee in an AnyTrust chain, or to anyone else who incurs per-calldata-byte costs on behalf of the chain. The reward is a fixed number of wei per data unit, and is paid to a single address.

The L1 pricer keeps track of the funds due to the reward address, based on the number of data units handled so far. This amount is updated whenever a batch posting report arrives at L2.

Allocating funds and paying what is owed

When a batch posting report is processed at L2, the pricer allocates some of the collected funds to pay for costs incurred. To allocate funds, the pricer considers three timestamps:

- currentTime
- is the current time, when the batch posting report message arrives at L2
- updateTime
- is the time at which the reported batch was submitted (which will typically be around 20 minutes before currentTime)
- lastUpdateTime
- is the time at which the previous reported batch was submitted

The pricer computes an allocation fractionF = (updateTime-lastUpdateTime) / (currentTime-lastUpdateTime) and allocates a fractionF of funds in theL1PricerFundsPool to the current report. The intuition is that the pricer knows how many funds have been collected betweenlastUpdateTime andcurrentTime , and we want to figure out how many of those funds to allocate to the interval betweenlastUpdateTime andupdateTime . The given formula is the correct allocation, if we assume that funds arrived at a uniform rate during the interval betweenlastUpdateTime andcurrentTime . The pricer similarly allocates a portion of the total data units to the current report.

Now the pricer pays out the allocated funds to cover the rewards due and the amounts due to batch posters, reducing the balance due to each party as a result. If the allocated funds aren't sufficient to cover everything that is due, some amount due will remain. If all of the amount due can be covered with the allocated funds, any remaining allocated funds are returned to theL1PricerFundsPool .

Adjusting the L1 gas basefee

After allocating funds and paying what is owed, the L1 Pricer adjusts the L1 Gas Basefee. The goal of this process is to find a value that will cause the amount collected to equal the amount owed over time.

The algorithm first computes the surplus (funds in theL1PricerFundsPool , minus total funds due), which might be negative. If the surplus is positive, the L1 Gas Basefee is reduced, so that the amount collected over a fixed future interval will be reduced by exactly the surplus. If the surplus is negative, the Basefee is increased so that the shortfall will be eliminated over the same fixed future interval.

A second term is added to the L1 Gas Basefee, based on the derivative of the surplus (surplus at present, minus the surplus after the previous batch posting report was processed). This term, which is multiplied by a smoothing factor to reduce fluctuations, will reduce the Basefee if the surplus is increasing, and increase the Basefee if the surplus is shrinking.

Getting L1 fee info

The L1 gas basefee can be queried viaArbGasInfo.getL1BaseFeeEstimate . To estimate the L1 fee a transaction will use, theNodeInterface.gasEstimateComponents() orNodeInterface.gasEstimateL1Component() method can be used.

Arbitrum transaction receipts include agasUsedForL1 field, showing the amount of gas used on L1 in units of L2 gasEdit this page Last updatedonMar 7, 2024 Previous Gas and Fees Next Inside Arbitrum Nitro