

Transaction Fees on OP Mainnet

OP Mainnet is designed to be [EVM equivalent\(opens in a new tab\)](#), which means it reuses the same Ethereum code you're already familiar with and behaves as much like Ethereum as possible. However, transaction fees on all Layer 2 systems need to diverge from Ethereum to some extent for a number of reasons. This page provides a detailed look at exactly how transaction fees work on OP Mainnet so you can properly account for them in your application.

OP Mainnet transaction fees are composed of an [Execution Gas Fee](#) and an [L1 Data Fee](#). The total cost of a transaction is the sum of these two fees. Continue reading to learn more about exactly how each of these fee components are charged.

Execution Gas Fee

A transaction's execution gas fee is exactly the same fee that you would pay for the same transaction on Ethereum. This fee is equal to the amount of gas used by the transaction multiplied by the gas price attached to the transaction. Like Ethereum, OP Mainnet uses the [EIP-1559\(opens in a new tab\)](#) mechanism to set the [base fee\(opens in a new tab\)](#) for transactions. The total price per unit gas that a transaction pays is the sum of the [base fee\(opens in a new tab\)](#) and the optional additional [priority fee\(opens in a new tab\)](#).

Because OP Mainnet is EVM equivalent, the gas used by a transaction on OP Mainnet is exactly the same as the gas used by the same transaction on Ethereum. If a transaction costs 100,000 gas on Ethereum, it will cost 100,000 gas on OP Mainnet. The only difference is that the gas price on OP Mainnet is much lower than the gas price on Ethereum so you'll end up paying much less in ETH.

For this component of the fee, you can estimate the total cost of a transaction using the same tools you would use to estimate the cost of a transaction on Ethereum. You can read more about how Ethereum's gas fees work over on [Ethereum.org\(opens in a new tab\)](#).

Base Fee

The [base fee\(opens in a new tab\)](#) is the minimum price per unit of gas that a transaction must pay to be included in a block. Transactions must specify a maximum base fee higher than the block base fee to be included. The actual fee charged is the block base fee, even if the transaction specifies a higher maximum base fee.

The OP Mainnet base fee behaves exactly like the Ethereum base fee with a few small parameter changes to account for the much shorter block times on OP Mainnet. None of these parameters should significantly impact your application, but you can read more about each of these parameters on the [OP Mainnet differences](#) page. Read more about the base fee in the [Ethereum.org documentation\(opens in a new tab\)](#).

Priority Fee

Just like on Ethereum, OP Mainnet transactions can specify a priority fee. This priority fee is a price per unit of gas that is paid on top of the base fee. For example, if the block base fee is 1 gwei and the transaction specifies a priority fee of 1 gwei, the total price per unit of gas is 2 gwei. The priority fee is an optional component of the execution gas fee and can be set to 0.

The OP Mainnet sequencer will prioritize transactions with a higher priority fee and execute them before any transactions with a lower priority fee. If transaction speed is important to your application, you may want to set a higher priority fee to ensure that your transaction is included more quickly. The [eth_maxPriorityFeePerGas \(opens in a new tab\)](#) RPC method can be used to estimate a priority fee that will get your transaction included quickly.

L1 Data Fee

The L1 Data Fee is the only part of the OP Mainnet transaction fee that differs from the Ethereum transaction fee. This fee arises from the fact that the transaction data for all OP Mainnet transactions is published to Ethereum. This guarantees that the transaction data is available for nodes to download and execute. The L1 Data Fee accounts for the cost to publish an OP Mainnet transaction to Ethereum and is primarily determined by the current base fee on Ethereum. After the Ecotone upgrade, OP Stack chains will have the option of posting transactions using [blobs \(opens in a new tab\)](#). For blob-enabled chains, the current Ethereum blob data gas price will largely determine the L1 data fee.

Mechanism

The L1 Data Fee is automatically charged for any transaction that is included in an OP Mainnet block. This fee is deducted directly from the address that sent the transaction. The exact amount paid depends on the estimated size of the transaction in bytes after compression, the current Ethereum gas price and/or blob gas price, and several small parameters.

The L1 Data Fee is most heavily influenced by the Ethereum base fee that is continuously and trustlessly relayed from Ethereum to OP Mainnet. With the Ecotone upgrade, the Ethereum blob base fee also gets relayed to OP Mainnet, and will

become the most important factor in for chains configured to use blobs instead of base fee. The base fee and blob base fee is updated on OP Mainnet for every Ethereum block, and each fluctuates at most by 12.5% between updates. As a result, short-term fluctuations of the L1 Data Fee are generally quite small and should not impact the average transaction.

The L1 Data Fee is charged automatically. It is currently not possible to limit the maximum L1 Data Fee that a transaction is willing to pay. This is the result of limitations in existing Ethereum transaction formats and tradeoffs that the OP Stack makes to retain better EVM equivalence. Work is underway to improve this situation with the introduction of a new standardized transaction type. For now, you should be aware that the L1 Data Fee can fluctuate with the Ethereum gas price.

Formula

Bedrock

The L1 Data Fee formula would change with the proposed Ecotone upgrade. Ecotone has not yet been approved by Optimism Governance. Refer to the [Network Upgrade Overview](#) to check when the Ecotone upgrade is scheduled to activate on OP Sepolia and OP Mainnet. Prior to the Ecotone upgrade, the L1 Data Fee is calculated based on the following parameters:

- The signed
- transaction serialized according to [the standard Ethereum transaction RLP encoding \(opens in a new tab\)](#)
- .
- The current Ethereum base fee (trustlessly relayed from Ethereum).
- A fixed overhead cost for publishing a transaction (currently set to 188 gas).
- A dynamic overhead cost which scales with the size of the transaction (currently set to 0.684).

The L1 Data Fee calculation first begins with counting the number of zero bytes and non-zero bytes in the transaction data. Each zero byte costs 4 gas and each non-zero byte costs 16 gas. This is the same way that Ethereum calculates the gas cost of transaction data.

tx_data_gas

count_zero_bytes (tx_data)

*

4

+

count_non_zero_bytes (tx_data)

*

16 After calculating the gas cost of the transaction data, the fixed and dynamic overhead values are applied.

tx_total_gas

(tx_data_gas + fixed_overhead) * dynamic_overhead Finally, the total L1 Data Fee is calculated by multiplying the total gas cost by the current Ethereum base fee.

L1_data_fee

tx_total_gas * ethereum_base_fee

Ecotone

The L1 Data Fee formula would change with the proposed Ecotone upgrade. Ecotone has not yet been approved by Optimism Governance. Refer to the [Network Upgrade Overview](#) to check when the Ecotone upgrade is scheduled to activate on OP Sepolia and OP Mainnet. The pricing function changes with the proposed Ecotone upgrade because of the introduction of the option to post transaction batches to the L1 using blobs instead of through L1 calldata. This updated function uses the following parameters:

- The signed
- transaction serialized according to [the standard Ethereum transaction RLP encoding \(opens in a new tab\)](#)
- .

- The current Ethereum base fee and/or blob base fee (trustlessly relayed from Ethereum).
- Two new scalar parameters that independently scale the base fee blob base fee.

At the exact point of the Ecotone upgrade, the dynamic overhead parameter value is used to initialize the Ecotone base fee scalar, and blob base fee is set to 0. The overhead parameter from the previous function becomes ignored.

The Ecotone L1 Data Fee calculation begins with counting the number of zero bytes and non-zero bytes in the transaction data. Each zero byte costs 4 gas and each non-zero byte costs 16 gas. This value, when divided by 16, can be thought of as a rough estimate of the size of the transaction data after compression.

tx_compressed_size

$$[(\text{count_zero_bytes}(\text{tx_data}) * 4 + \text{count_non_zero_bytes}(\text{tx_data}) * 16)] /$$

16 Next, the two scalars are applied to the base fee and blob base fee parameters to compute a weighted gas price multiplier.

weighted_gas_price

$16 * \text{base_fee_scalar} * \text{base_fee} + \text{blob_base_fee_scalar} * \text{blob_base_fee}$ The L1 data fee is then:

l1_data_fee

$\text{tx_compressed_size} * \text{weighted_gas_price}$ Recall that `base_fee_scalar` is set to `dynamic_overhead` and `blob_base_fee_scalar` is 0 immediately following the upgrade. Because the old overhead parameter becomes ignored, new L1 data prices will be (slightly, since overhead is typically a very small) lower than before the fork. Chain operators will likely want to retune the parameters appropriately after the fork, particularly if they plan on enabling blobs.

[Cost Optimization Estimating Fees](#)