# Estimating Transaction Fees on OP Mainnet

Check out the guide on understandingTransaction Fees on OP Mainnet for an in-depth explanation of how OP Mainnet transaction fees work. It's important be able to properly estimate the cost of a transaction on OP Mainnet before submitting it to the network. Here you'll learn how to estimate both of the components that make up the total cost of an OP Mainnet transaction, theexecution gas fee and theL1 data fee . Make sure to read the guide onTransaction Fees on OP Mainnet for a detailed look at how these fees work under the hood.

## Execution Gas Fee

Estimating the execution gas fee on OP Mainnet is just like estimating the execution gas fee on Ethereum. Steps are provided here for reference and convenience, but you can use the same tooling that you'd use to estimate the execution gas fee for a transaction on Ethereum. 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. Refer to the guide onTransaction Fees on OP Mainnet for more information about the execution gas fee.

When estimating the execution gas fee for a transaction, you'll need to know the gas limit and themax fee per gas(opens in a new tab) for the transaction. Your transaction fee will then be the product of these two values. Refer to the guide onSetting Transaction Gas Parameters on OP Mainnet to learn how to select an appropriate gas limit and max fee per gas for your transaction.

### Estimate the gas limit

Using the same tooling that you'd use to estimate the gas limit for a transaction on Ethereum, estimate the gas limit for your transaction on OP Mainnet. OP Mainnet is designed to beEVM equivalent(opens in a new tab) so transactions will use the same amount of gas on OP Mainnet as they would on Ethereum. This means you can feed your transaction to theeth_estimateGas (opens in a new tab) JSON-RPC method just like you would on Ethereum.

### Estimate the max fee per gas

Like Ethereum, OP Mainnet uses an EIP-1559 style fee market to determine the current base fee per gas. You can then additionally specify a priority fee (also known as a tip) to incentivize the Sequencer to include your transaction more quickly. Make sure to check out the guide onSetting Transaction Gas Parameters on OP Mainnet to learn more about how to select an appropriate max fee per gas for your transaction.

### Calculate the execution gas fee

Once you've estimated the gas limit and the max fee per gas for your transaction, you can calculate the execution gas fee by multiplying these two values together.

For instance, suppose that your transaction has a gas limit of420000 gas , a base fee of0.05 gwei , and a priority fee of0.1 gwei . The execution gas fee for your transaction would be:

// Start with your parameters gas_limit =

420000 base_fee_per_gas =

0.05 gwei priority_fee_per_gas =

0.1 gwei

// Max fee per gas is the sum of the base fee and the priority fee max_fee_per_gas = base_fee_per_gas + priority_fee_per_gas =

0.15 gwei

// Execution gas fee is the product of the gas limit and the max fee per gas execution_gas_fee = gas_limit * max_fee_per_gas =

420000

*

0.15 gwei =

0.000063

ETH

# L1 Data Fee

The Optimism SDK provides a convenient method for estimating the L1 data fee for a transaction. Check out the tutorial onEstimating Transaction Costs on OP Mainnet to learn how to use the Optimism SDK to estimate the L1 data fee for your transaction. Keep reading if you'd like to learn how to estimate the L1 data fee without the Optimism SDK. The L1 data fee is a fee paid to the Sequencer for the cost of publishing your transaction to Ethereum. This fee is paid in ETH and is calculated based on the size of your transaction in bytes and the current gas price on Ethereum. Refer to the guide onTransaction Fees on OP Mainnet for more information about the L1 data fee.

Unlike the execution gas fee, the L1 data fee is anintrinsic fee for every transaction. This fee is automatically charged based on the size of your transaction and the current Ethereum gas price. You currently cannot specify a custom L1 data fee for your transaction.

The L1 data fee is paid based on the current Ethereum gas price as tracked within theGasPriceOracle (opens in a new tab) smart contract. This gas price is updated automatically by the OP Mainnet protocol. Your transaction will be charged the Ethereum gas price seen by the protocol at the time that your transaction is included in an OP Mainnet block. This means that the L1 data fee for your transaction may differ from your estimated L1 data fee.

## Serialize your transaction

The L1 data fee is calculated based on the size of your serialized transaction in bytes. Most Ethereum tooling will provide a method for serializing a transaction. For instance, Ethersjs provides theethers.utils.serializeTransaction (opens in a new tab) method.

## Estimate the L1 data fee

Once you have serialized your transaction, you can estimate the L1 data fee by calling thegetL1Fee (opens in a new tab) method on theGasPriceOracle (opens in a new tab) smart contract available on OP Mainnet and all OP Stack chains. This method takes the serialized transaction as input and returns the L1 data fee in wei using the formula described in theTransaction Fees on OP Mainnet guide.

Fee estimation is typically performed before the transaction is signed. As a result, thegetL1Fee method assumes that your input is anunsigned Ethereum transaction.

## Tooling

Several tools are available to help you estimate the L1 Data Fee for your transaction. Selecting the right tool for your use case will depend on your specific needs.

- Viem(opens in a new tab)
- provides first-class support for OP Stack chains, including OP Mainnet. You can use Viem to estimate gas costs and send cross-chain transactions (like transactions through the Standard Bridge system). It's strongly recommended to use Viem if you are able to do so as it will provide the best native support at the moment.
- If you are using Ethers v5, theOptimism SDK(opens in a new tab)
- provides methods for estimating the L1 Data Fee for your transactions and for sending cross-chain transactions. The Optimism SDK is designed to be used alongside Ethers v5 and does not yet support Ethers v6.
- If you are using Ethers v6, theOptimistic Utilities Extension(opens in a new tab)
- provides methods for estimating the L1 Data Fee. The Ethers v6 extension does not yet support sending cross-chain transactions. Use Viem or the Optimism SDK if you need to send cross-chain transactions.

## Future Proofing

The L1 Data Fee formula is subject to change in the future, especially as the data availability landscape evolves. As a result, it's important to future proof your transaction fee estimation code to ensure that it will continue to function properly as the L1 Data Fee formula changes.

- Use existingtooling
- to estimate the L1 Data Fee for your transaction if possible. This tooling will be updated to reflect any changes to the L1 Data Fee formula. This way you won't need to modify your code to account for any changes to the formula.
- Use thegetL1Fee
- method on theGasPriceOracle
- if you are unable to use existing tooling. ThegetL1Fee
- method will be updated to reflect any changes to the L1 Data Fee formula. It's strongly recommended that you donot
- implement the L1 Data Fee formula yourself.

Transaction Fees Setting Gas Parameters