

Estimating Gas Fees

Both [OApp](#) and [OFT](#) come packaged with methods you can implement or call directly in order to receive a quote for how much native gas your message will cost to send to the destination chain.

Both the [OApp](#) and [OFT](#) implementations for estimating fees require some knowledge of how [_options](#) work. We recommend reviewing the [OApp](#) or [OFT Quickstart](#) and [Message Options](#) guides first to better understand [_options](#) usage.

OApp

To estimate how much gas a message will cost to be sent and received, you will need to implement a `quote` function to return an estimate from the Endpoint contract to use as a `recommendedmsg.value`.

function

`quote (uint32 _dstEid ,`

`// destination endpoint id bytes`

`memory payload ,`

`// message payload being sent bytes`

`memory _options ,`

`// your message execution options bool`

`memory _payInLzToken // boolean for which token to return fee in)`

`public`

`view`

`returns`

`(uint256 nativeFee ,`

`uint256 zroFee)`

`{ return`

`_quote (_dstEid , payload , _options , _payInLzToken) ; }` The `_quote` can be returned in either the native gas token or `inLzToken`, supporting both payment methods.

In general, this quote will be accurate as the same function is used by the Endpoint when pricing an `_lzSend` call:

`// How the _quote function works. // This function is already defined in your OApp contract. /// @dev the generic quote interface to interact with the LayerZero EndpointV2.quote() function`

`_quote (uint32 _dstEid , bytes`

`memory _message , bytes`

`memory _options , bool _payInLzToken)`

`internal`

`view virtual returns`

`(MessagingFee memory fee)`

`{ return endpoint . quote (MessagingParams (_dstEid ,`

`_getPeerOrRevert (_dstEid) , _message , _options , _payInLzToken) , address (this)) ; }` tip Make sure that the arguments passed into the `_quote` function identically match the parameters used in the `lzSend` function. If parameters mismatch, you may run into errors as `yourmsg.value` will not match the actual gas quote.

note Remember that to send a message, `amsg.sender` will be paying the source chain, the selected DVNs to deliver the message, and the destination chain to execute the transaction.

OFT

To estimate how much gas an OFT transfer will cost, call the `quoteSend` function to return an estimate from the Endpoint contract.

// @dev Requests a nativeFee/lzTokenFee quote for sending the corresponding msg cross-chain through the layerZero Endpoint function

```
quoteSend ( SendParam calldata _sendParam ,  
  
// send parameters struct bytes  
  
calldata _extraOptions ,  
  
// extra message options bool _payInLzToken ,  
  
// bool for payment in native gas or LzToken bytes  
  
calldata _composeMsg ,  
  
// data for composed message bytes  
  
calldata _oftCmd // data for custom OFT behaviours ) public view virtual returns  
  
( MessagingFee memory msgFee ,  
  
// fee struct for native or LzToken OFTLimit memory oftLimit , OFTReceipt memory oftReceipt , OFTFeeDetail [ ]  
  
memory oftFeeDetails // @dev unused in the default implementation, future proofs complex fees inside of an oft send ) { ( oftLimit , oftReceipt )  
  
=  
  
quoteOFT ( _sendParam ) ;  
  
( bytes  
  
memory message ,  
  
bytes  
  
memory options )  
  
=  
  
_buildMsgAndOptions ( _sendParam , _extraOptions , _composeMsg , oftReceipt . amountCreditLD ) ;
```

msgFee

_quote (_sendParam . dstEid , message , options , _payInLzToken) ; [Edit this page](#)

[Previous](#) [Execution Options](#) [Next](#) [Transaction Pricing](#)