Research

What is Gas on Ethereum, Why is it Needed, and How Do I Get Some?

Alex Mead

August 5, 2022

Motivation

The Ethereum Ecosystem continues to see widespread growth, even in the midst of the latest "crypto winter" of Summer 2022. Whether it's ERC-20 tokens (e.g. USDT, USDC), lucrative ERC-721 (NFT's) drops (e.g. BAYC), DeFi protocols like Decentralized Exchanges (DEX) (e.g. Uniswap, SushiSwap), decentralized autonomous organizations (DAO's) (e.g. CityDAO, BitDAO), or the most burgeoning world of Layer-Two Systems (e.g.Optimism, Polygon, Arbitrum), it's all growing. Linking these exciting projects is Ethereum and with that, evolving gas usage and cost come straight to mind.

Outline

Beginning in Part I a theoretical explanation of the need for gas from a computational theory perspective is given. The section then expands on how the theory of computing is realized on Ethereum. In Part II, gas allocation and the pricing mechanism of how it is acquired by users is presented, with specific historical context with respect to EIP-1559 (London Hardfork - block: 12,965,000). Finally, Part III provides a brief introduction to auction theory to motivate the curious reader to learn more. Advanced topics are included to close, including the Mempool and Maximum-Extractable Value (MEV).

I Theory of Computation: The What and Why of Gas on Ethereum

What is Ethereum gas?

In Ethereum, gas is a measurement of computational effort and memory usage on-chain by transactions, very often executed by smart contracts. Similar to how a car needs gas to run, Ethereum smart contracts need gas in order to be executed. Each operation (opcode) in the Ethereum Virtual Machine (EVM) - smart contracts are made up of several opcodes linked together - is assigned a cost in gas, roughly equivalent to its computational and/or memory usage cost to execute.

For example, if a smart contract adds two numbers (Opcode 0x01 in the EVM) in the process of its execution, the cost will be 3 units of gas for this specific step. As such, one can reason about the relative complexity of execution of different smart contracts by examining various execution paths and their expected cost in gas to execute by summing each step.

With this, a rough estimate of the computations that will take place and therefore the gas usage in a contract execution, can be determined. For example, a transfer of ETH between accounts costs

21,000 gas, while more complex smart contracts operations may cost 100,000 or more gas. Figure 1 below shows the daily total of gas used by Ethereum and how it has changed over time.

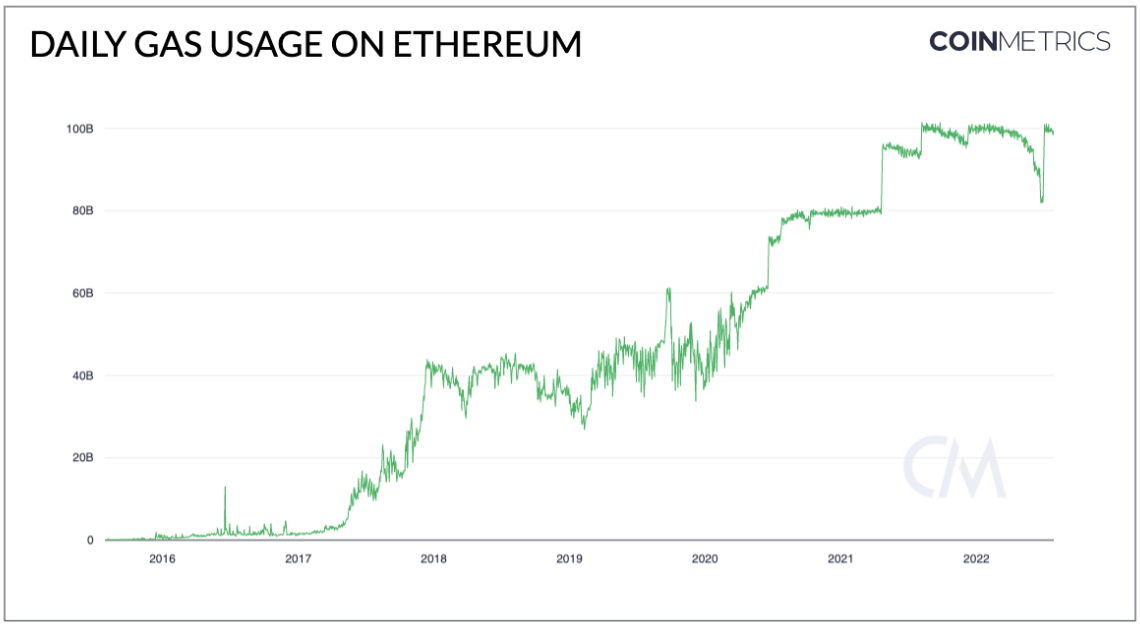


Figure 1: Total Daily Gas Usage [Gwei] on the Ethereum Network. (Source: Coin Metrics Network Data Pro)

Why is gas needed at all?

The first and primary reason for gas within Ethereum is to prevent a single smart contract (aka program) from overwhelming the EVM via non-termination.

Non-terminating smart contracts are programs that once started fail to stop running (often caused by infinite loops or dead-locks within the code logic), resulting in a lockup of the EVM, never allowing another smart contract to get a turn to execute. Given Ethereum is a Turing Complete, decentralized computational platform, these types of smart contracts are a distinct possibility and can be either maliciously or accidentally deployed to the EVM. Regardless of origin or intent, a non-terminating smart contract can occupy the EVM forever, rendering the Ethereum Ecosystem useless.

To fix this potential failure mode of the EVM, the system needs to identify non-terminating smart contracts and not allow them into the Ethereum ecosystem. This identification process is known in computer science as the Halting Problem. Unfortunately, we know thanks to Alan Turing, this is in fact an intractable challenge. Put simply, it is impossible to determine a priori whether a given smart contract and set of inputs will terminate or not. Fortunately, there do exist practical solutions to the Halting Problem, and *Gas* is the chosen solution for Ethereum.

How does gas solve the Halting Problem?

The Ethereum gas usage protocol therefore goes roughly like this: a certain amount of gas is pre-allocated to a smart contract for execution as a user-defined parameter. As the smart contract executes, more and more gas is consumed as opcodes are performed in the EVM[1]. This process continues, ending either with the successful termination of the smart contract as intended and a refund of the unused gas, or the total usage of the allocated gas, and the forced termination of the smart contract and the roll-back of all the changes it performed during its partial execution.

[1] A good resource to visualize the gas usage of each step of an operation in the EVM is the playground on evm.codes

Thus, gas is a mechanism to quantify the computational requirements of a given smart contract within Ethereum and prevent non-terminating smart contracts from overwhelming the EVM by creating an ultimate user defined ceiling for the quantity of computation a smart contract is permitted to perform. So you may now ask, "Why don't I just put a very large number for the gas limit of my smart contract execution so I guarantee it won't run out?"

Gas Limits in Ethereum

For various practical and philosophical reasons, the total usage rate of gas across all Ethereum users is capped. First, only so much gas can be spent per block, and second, block production is limited per time. Therefore gas, and the underlying computational and memory resources it represents, is a limited resource and must be allocated in some manner to those wanting to use Ethereum.

The solution to the gas allocation problem chosen by the Ethereum designers involves the sale of gas to the users, with those willing to pay more getting priority access to the limited amount of gas per block. The gas available is auctioned off to the highest bidder. In effect, Ethereum usage is thus given to the users who are willing to pay for it the most.

Figure 2 below shows the per block gas limit for Ethereum as a function of block height. Notice the evolution of the gas limit trending up with its current limit of 30,000,000 Gwei per block. Each "step" can be traced to a protocol level change increasing the gas limit, for example the 15 to 30 million Gwei jump implemented with EIP-1559 at block 12,965,000 is discussed further in this paper below.

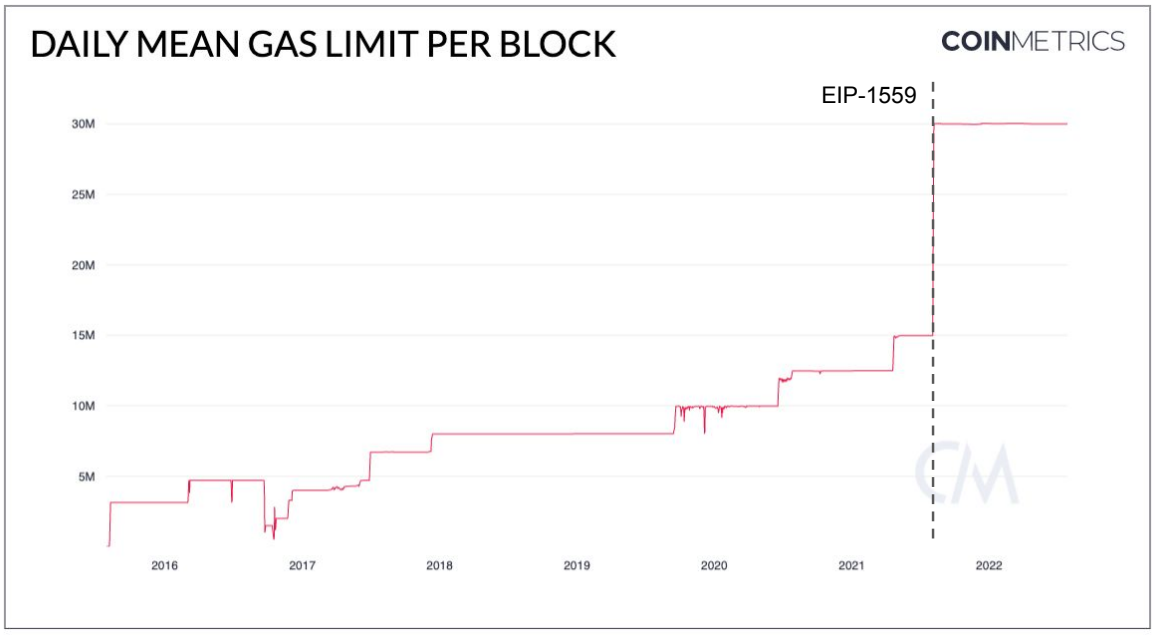


Figure 2: Daily Mean Gas Limit Per Block [Gwei] for the Ethereum Blockchain. (Source: Coin Metrics Network Data Pro)

The challenging part about allocating gas this way - to those willing to pay the highest price - is that auction design is complicated. Complicated and opaque enough in fact, many designers get it wrong for their application and end up with unanticipated results. Next, we will explore the two main time periods of gas allocation within Ethereum: before and after The London Hardfork which came into effect at block height 12,965,000 (August 5, 2021).

II Implementation of Theory in Practice on Ethereum Gas

How is Gas Actually Implemented on Ethereum?

The allocation of gas within the Ethereum Ecosystem can be divided historically into two distinct phases: before and after the London Hardfork of August 5, 2021 (taking effect at block height 12,965,000). The London Hardfork implemented Ethereum Improvement Proposal (EIP) EIP-1559, changing the mechanism design (i.e. auction) for how gas is purchased by users for their transactions to be executed within Ethereum[2]. Regardless of the phase, however, ultimately the user questions being answered are, "How much does gas cost?" and "Does the value I derive from my transaction warrant the cost I need to pay to execute my transaction?"

Pre EIP-1559: Math for Clarity and an Example of How it Works

For pre-London Hardfork transactions, a user submits two parameters with their transaction concerning gas: gas limit (g_{limit})[3] and gas price (p_{bid}). g_{limit} is the total number of units of gas the user is willing to spend on executing some transaction and it acts as the Halting Problem protection mechanism, see above. p_{bid} is the bid price per unit of gas a user is willing to pay, and has units of $\left[\frac{Gwei}{gas}\right]$. Gwei (i.e. "giga-wei") is an increment of 10^{-9} ETH, the native currency of the Ethereum Block chain. For example, if a user bids $p_{bid} = 50$ Gwei and $g_{limit} = 21,000$ gas (standard cost of a transfer of ETH from one account to another), the cost they would pay (e_{cost}) is:

$$e_{cost} = p_{bid} * g_{limit} = 50 * 21000 = 1,050,000\ Gwei, \tag{1}$$

or 0.00105 ETH[4]. ETH can then be further converted to the currency of a user's choosing to price the cost of a given transaction.

The challenging part on the user's side is picking g_{limit} and p_{bid}. To begin, g_{limit} can be judged based on the amount of computational steps the transaction is estimated to take. For well known transaction types, such as transferring ETH from account to account, g_{limit} can be estimated with fairly high certainty. Further, even if the user overestimates the unused gas will be converted back to ETH and returned to the user. Thus, assuming a user has sufficient balance to fund a high gas limit, they should. In contrast, the optimal choice for p_{bid} is constantly changing as a function of the willingness of a user to pay for that gas based on market conditions. This "willingness" is a measure of the value individual users place on their transactions. The users who are willing to pay the most per gas will have their transactions prioritized to use the gas made available in the nearest block.

It should be noted that the cost in ETH (e_{cost})[5] used by these transactions is transferred to the miner[6] who successfully builds the next block in the chain. This means the ETH balance for a user account submitting a transaction is debited and the corresponding miner balance is credited after the transaction has been successfully included in the next block.

Problems that Emerged In Gas Auctions on Ethereum

The above mechanism design for the auction of gas boils down to what economists call a First-Price Auction, sometimes also called an English Auction.This mechanism is used in many different markets, however, there are known shortcomings to this auction type that began to emerge in the Ethereum implementation. For example, there is no built in mechanism in the system itself to measure demand, hence the user is left to estimate this demand based on their perception of the current bids. Practically speaking, most users cannot see these "former bids," however, as they do not have access to the mempool - the location bids and transactions go to be uploaded to the network. Hence, the transaction bidding gas auction is mostly blind, and estimating gas demand to place a personalized gas bid is not easy. While some tools did exist in wallets to assist users, they often left users "overpaying" for transactions to guarantee their inclusion in the current block. High volatility in the gas price required to be included was also common.

[2]Technically users can still use the pre-London Hardfork mechanism as well, and some still do.
[3]See Glossary for an organized table of all definitions related to gas and transaction costs.
[4]Roughly $1.75 at the ETH/USD rate at time of writing
[5]See Glossary for definition.
[6]Validator in proof-of-stake.

Gas price is ultimately an indicator of demand of the underlying block space, hence its pricing mechanism cannot be expected to bring down gas prices. However, it was believed gas prices could become more predictable (i.e. less volatile) if a different pricing mechanism was used. As such, EIP-1559 was proposed, and ultimately adopted into Ethereum.

Post EIP-1559: Math for Clarity and an Example of How it Works

Before EIP-1559, block size[7] (g_{size}) was fixed at a hard maximum of 15 million gas and no mechanism was in place in Ethereum itself to communicate to users the aggregate demand for that block space. With EIP-1559 several changes were introduced to address these shortcomings and provide a more efficient gas market.

First off, the maximum block size (g_{max}) was increased to 30 million gas with a target size (g_{target}) of 15 million gas. Next, the price of gas itself as paid by the user is now broken down into two parts, a base gas fee ($p_{baseFee}$) and priority fee ($p_{priorityFee}$), both measured in Gwei. Meaning, each block with height $h > 12,965,000$ has a price, $p^h_{baseFee}$, calculated by Ethereum which is the minimum price a user must pay to be included in a block. This price is dynamically updated based on the previous block's size, hence it is designed to be a measure of the current demand for the gas in the current block. $p_{baseFee}$ is updated as follows between block height h and $h+1$:

$$p^{h+1}_{baseFee} = p^h_{baseFee}\left(1 + \frac{1}{8}\frac{g^h_{size} - g_{target}}{g_{target}}\right). \tag{2}$$

Notice, this means that $p^h_{baseFee}$ will continuously change as long as the block sizes differ from their target size, $g_{target} = 15 * 10^6$, with a rate between 12.5% and -12.5% as compared to the previous block. This mechanism is thus dynamically adjusting gas price based on the demand for block space in an automated fashion.

To further illustrate a user's preference for their transaction to be included beyond the $p^h_{baseFee}$, a user can also include a priority fee, $p_{priorityFee}$, sometimes called a Miner Tip. Finally, users also include a maximum price for gas they are willing to pay, p_{max}, as well as the maximum gas, g_{limit}, they will spend with each transaction. These values then combine into three possible cases for a transaction with respect to each block, illustrated below in Table 1, influencing the cost the user ultimately pays.

Table 1: The three possible cases of gas price for a user to pay for a transaction. While inclusion is not guaranteed in either of the three cases, the stated aims of EIP-1559 were 1) reduce over payment of fees 2) faster and more certain inclusion in a block.

Case	Conditional	Gas Price (p_{paid})	Block Inclusion?
1	$p_{priorityFee} + p^h_{baseFee} < p_{max}$	$p_{paid} = p_{priorityFee} + p^h_{baseFee}$	Maybe
2	$p_{priorityFee} + p^h_{baseFee} > p_{max}$, $p_{max} \geq p^h_{baseFee}$	$p_{paid} = p_{max}$	Maybe
3	$p_{max} < p^h_{baseFee}$		No

The cost of the given transaction, assuming it is "picked up" by a miner and included in a block is thus,

$$e_{cost} = p_{paid} * g_{used}, \tag{3}$$

where $g_{used} \leq g_{limit}$.

While the gas cost and transaction pricing mechanism within Ethereum is straightforward, the reality of getting a transaction included in a timely manner is more complicated. Off-chain payments, sometimes called out-of-band payments, can occur between savvy users and miners directly. This makes the fee mechanism more complicated and adds significant market complexity. More is said below in the MEV section.

[7] One should take note, block size is actually measured in gas and is a unit representing computational and memory resources, however, gas price is measured in Gwei per gas.

Special Implications of EIP-1559 on Ethereum Supply

To avoid corrupting the gas market within the Ethereum system with outside intervention, the base fee ($p^h_{baseFee}$) component of the total gas fee is burned with every transaction, with only the priority fee ($p_{priorityFee}$) component going to the miner. While it is complex to understand the reasoning behind the choice to burn the base fee, it is straightforward to see that burning of ETH could, given certain market conditions, turn ETH into a deflationary asset as opposed to an inflationary asset, see Figure 3 below. This market-dependent conditioning, however, is very challenging if not impossible to predict a priori.

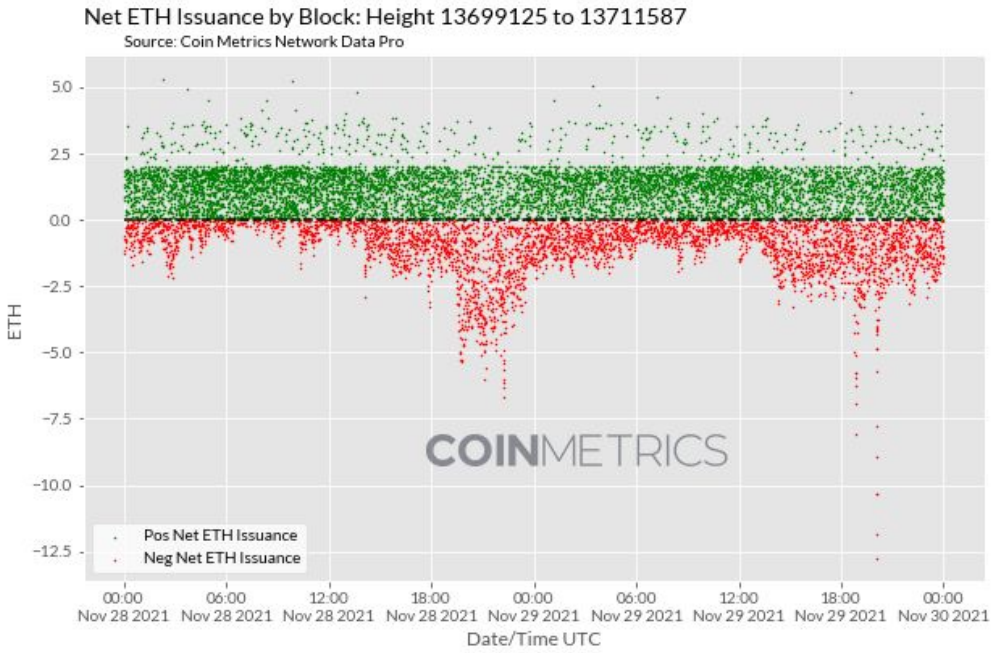


Figure 3: Net ETH Issuance by Block on the Ethereum Blockchain, two-day sample. (Source: Coin Metrics Network Data Pro)

Examples of Transaction Costs

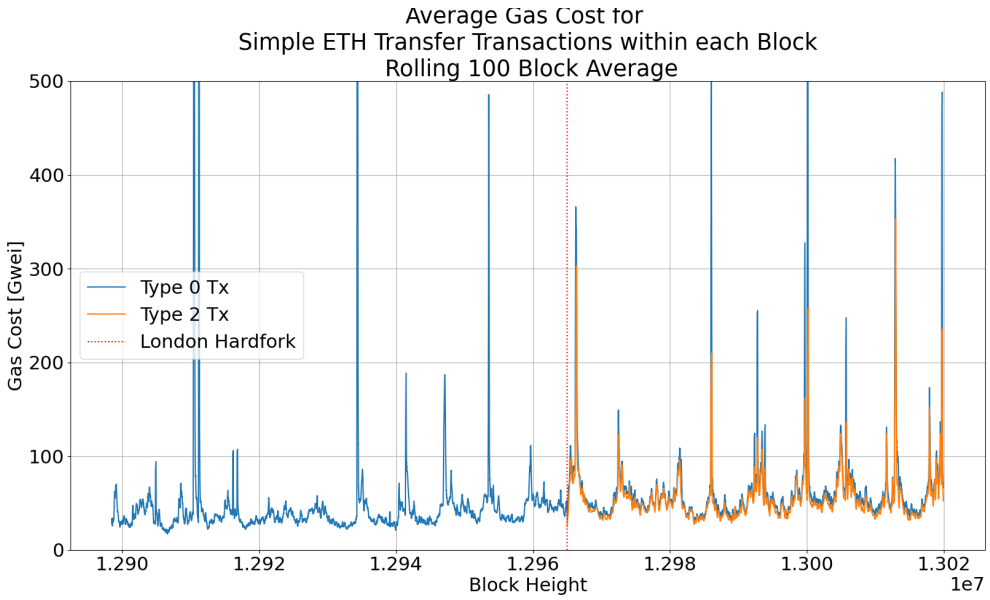
With the mechanism outlined above for how gas is purchased for a given transaction, examples are now given to provide points of reference to learn how the system actually works in practice.

Example 1: Simple Transfer of Funds

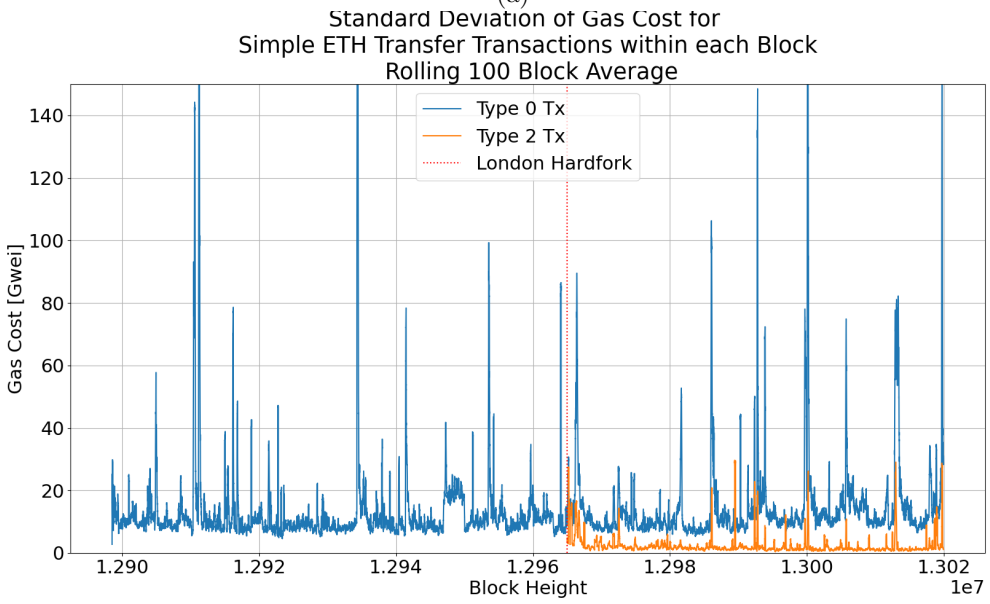
Imagine a user wanting to transfer a balance of ETH from their own account to another with the goal of sending "money" to a friend. While the transaction needs to happen soon, it doesn't need to happen now. For this example, the user may choose a small tip and a max fee below the daily average of the base fee, with the thought that this is a low priority transaction. In this case, the transaction will remain pending until the base fee drops during the low chain activity period and the transaction becomes viable. To add clarity, perhaps the user picks $p_{priorityFee} = 2$ and $p_{max} = 65$, during a period of $p^{7dAvg}_{baseFee} = 50$. The thought being, even if $p_{baseFee} > 65$ currently (i.e. Case 3 in Table 1), there will likely be a period in the next 24 hours where the transaction falls into case 1 or 2 and it will likely be executed then. For this type of low priority transaction, gas strategies such as this are easier to be executed successfully with the EIP-1559 mechanism.

On-Chain Data: Transaction Cost Variability

To explore the impact of EIP-1559, transaction data from July 26 to August 15, 2021 was collected (pre/post-EIP-1559 respectively). For each block on those days, the simple ETH transfer transactions ($g_{limit} = 21,000$) were selected and the mean and standard deviation was calculated on a block-by-block basis. For blocks before the hard fork on August 5, 2021 (height=12,965,000) all transactions use the original gas pricing method (called type 0), while for blocks after the hard fork, some use the original and some use the new method (called type 2). Figure 4 below shows the time series for these three separate data types. In order to make the trends more clear in the plots, 100 block moving averages are shown.



(a)

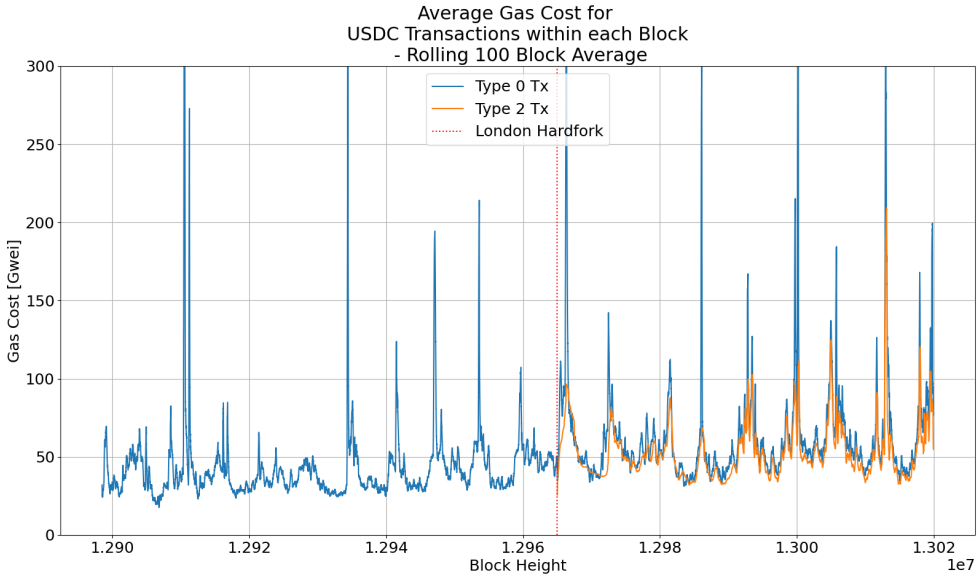


(b)

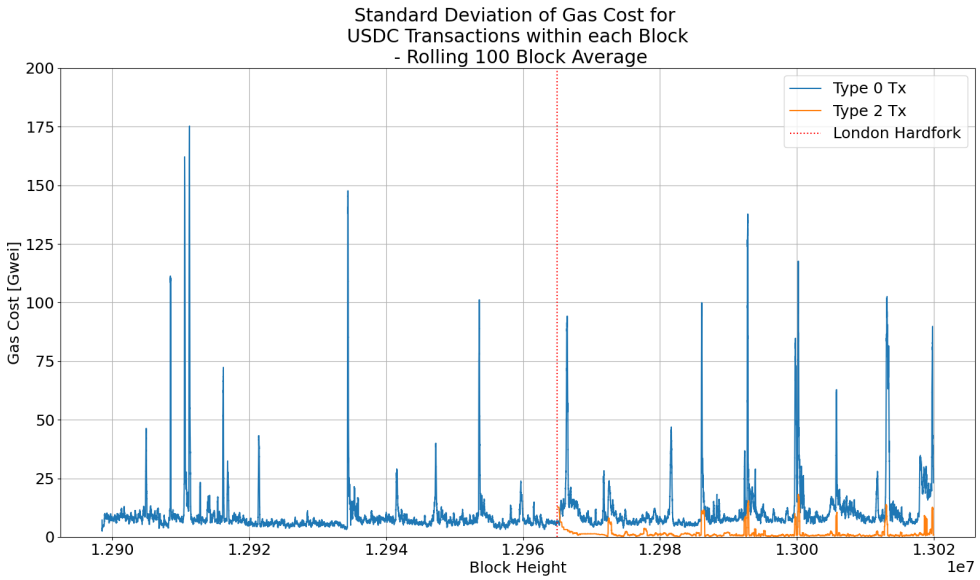
Figure 4: (a) 100 Block moving average, of average gas cost per block for simple ETH transfer transactions, (b) 100 Block moving average, of standard deviation for gas cost per block for simple ETH transfer transactions (Source: Coin Metrics Labs)

Notice in Fig. 4 (a) how the average cost of gas for transactions varies little for either type 0 or 2, however, in (b) the standard deviation - or the "spread" - of prices paid dropped significantly for type 2 versus type 0. This is as expected, because price is a signal of demand, and while the pricing mechanism can more effectively discover the "true market price" - hence less spread in the prices paid - it cannot significantly affect the demand and thus the average price paid.

Extending the analysis from above beyond simple Ether transfer transactions, the popular stablecoin USD Coin (USDC) was also examined for the period of July 26 to August 15, 2021, see Figure 5. Very similar results to the simple Ether transfers were also found. This adds evidence to the claim EIP-1559 did in fact make gas prices more stable and thus easier to estimate for the average user.



(a)



(b)

Figure 5: (a) 100 Block moving average, of average gas cost per block for USDC transactions, (b) 100 Block moving average, of standard deviation for gas cost per block USDC transactions (Source: Coin Metrics Labs)

Example 2: Popular NFT Mint

In contrast to the simple ETH transfer and token purchase above, suppose a user is 'bidding' on a popular, limited edition NFT mint, for example the Bored Apes Yacht Club (BAYC) Otherside mint of May 1, 2022. In this case, there are a limited number of the NFTs, made available for a limited time period, hence the gas bidding strategy must be very different than a simple ETH transfer. The user must carefully select their $p_{priorityFee}$ with the knowledge of the current bids on the Ethereum Network at large to be competitive, but also their p_{max} must be carefully selected to ensure it meets the $p_{baseFee}$ requirement. One must keep in mind, this is just to get the user's transaction included in the ETH blocks. Another auction entirely may be taking place with the value component those transactions are actually carrying. To put the impact of the BAYC mint in perspective, $p_{baseFee}$ exceeded 8,000 Gwei for extended periods of May 1, 2022 as seen in Figure 6, by far the highest base fee ever realized on Ethereum to-date. This means users must be bidding at least $p_{max} > 8000Gwei$ to even have the chance of their transactions being included in the latest blocks.

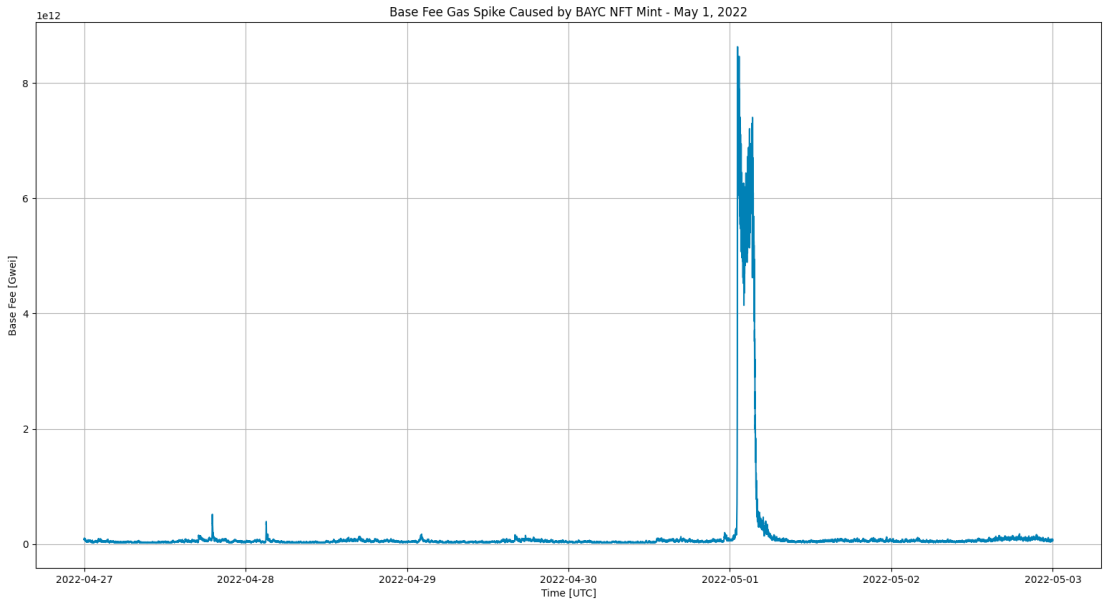


Figure 6: (BAYC NFT mint caused gas base fee spike, peaking over 8000 Gwei. (Source: Coin Metrics Network Data Pro)

The examples above of gas pricing show not all transactions are the same, and economic incentives for on-chain participants can skew their willingness to apparently overpay for inclusion in the next block. Although their behavior skews from the norm by a wide margin, it is ultimately rational given that failure to be included in a block surrenders your access to a time-constrained opportunity. We can therefore intuit that the maximum bidding range for fees can increase until it reaches parity with the economic value of blockspace and the expected value of a given transaction.

While endless strategies for gas bidding exist, experience has shown that using a more reason based approach typically results in better performance, regardless of the application domain. Gas bidding on Ethereum is no different, hence in the next section, economic models of auctions are presented. These models have shown to be effective tools that not only agree with empirical data, but also help sophisticated users make more informed decisions regarding gas price bids.

III Auction Theory and Advanced Gas Topics

Auction Theory: Mental Models for How to Think of Gas

Modern day system analysis often employs the usage of "models" to better understand the actual and theoretical behavior of some desired phenomena. Given the vast monetary stakes involved, financial markets have often attracted the latest in analysis techniques, and Ethereum is no exception.

In this section, a theoretical background to some of the components of the Ethereum system are presented. This presentation is intentionally brief as the subject is of great depth and could easily fill several university courses. The goal here is to provide an outline of the subject, complete with references to more complete resources for the motivated reader to explore.

First Price Auction

A First Price Auction, sometimes called an English Auction, is typically what someone thinks of by the word "auction." That is, participants in the auction progressively call out higher numbers, referred to as bids, communicating their inherent preference to purchase some item (e.g. an art auction). When a bid of sufficient magnitude is reached, and no one is willing to pay more, the user associated with that bid is chosen as the winner and upon payment of the bid, given the item at auction.

This model of scarce resource allocation fits the pre-EIP-1559 gas allocation system well. In that system, the miner will select the transactions with the highest bids for gas to be included in the next block. The miner picks these transactions because they receive the payments in these bids, thus they benefit directly from maximizing the transaction fees.

While this system certainly works, inefficiencies in the market arise. For example, the auction theory shows a user must reason about not only their bid and the inherent value preference in their own mind, they must play that off of other user's bids. This makes it challenging to isolate one's preference and willingness to pay against those one is biddings against to arrive at one's personal optimal bid. Arising from this challenge to users, rapid fluctuations in gas cost and "over paying" for transactions resulted. Hence, to improve the user experience on Ethereum a new transaction fee mechanism was proposed in EIP-1559.

Post-EIP-1559 Auction Models

While the Pre-EIP-1559 transaction fee mechanism lends itself to analysis via a classic first price auction model, the Post-EIP-1559 mechanism is a bit more complex. A now classic reference for its analysis is Tim Roughgarden's piece entitled, "Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559." This paper presents rigorous mathematical based reasoning for why the base fee must be burned, how a priority tip better represents a user's true preference for a transaction, and an explanation of how base fee adjustment acts is a measure of blockchain demand.

As a summary, the TL;DR for these concepts is as follows. For the burning of the base fee, if this is not done, there exist scenarios where both miner and user can increase their utility by colluding off-chain. Stated another way, if miners are paid the base fee, and their actions affect the base fee price, then miners' actions will be guided by their self interest to maximize fee collection at the expense of all other considerations. This could result in a "black market" for gas and transaction fees - clearly not advantageous for a blockchain system like Ethereum. Base fees in EIP-1559 provide a measure of demand, excluding situations with sudden increases in demand for blockchain space, and thus offer an obvious "optimal bid" for gas the user can choose if they are unsure. Further, to differentiate their implied value for a transaction to be included, the priority fee allows a user to further indicate their value beyond the base fee.

It should be noted, the reasoning behind these conclusions is highly non-trivial and involves deep game-theoretic reasoning. Curious readers are encouraged to explore this report to see the full conclusions of this thorough analysis.

Advanced Topics

Mempool and MEV

While the theory of auctions and mathematics of the gas pricing system appear simple, the reality of the Ethereum system is less straightforward. Next, the mechanics of submitting an Ethereum transaction are presented, specifically the Mempool. Furthermore, a fundamental complication in the Ethereum system, a behavior called Maximum Extractable Value (MEV) is introduced to close out the paper and motivate the reader to learn more about Ethereum.

The Ethereum Mempool

In Part II, the mechanisms for how the gas markets work are outlined. Whether the transactions include p_{bid} and g_{limit} for Pre-EIP-1559 or p_{max} and $p_{priorityFee}$ for Post-EIP-1559, all transactions are submitted to Ethereum and deposited together into a public queue called the mempool.

The mempool is simply the collection of all pending transactions in the Ethereum Ecosystem. This pool is populated by users submitting transactions to any node on the network, where the node then broadcasts that transaction to all other nodes on the network. The mining nodes then select transactions from that pool based on any algorithm they choose and build the blockchain competing in the Proof of Work protocol at the blockchain level. Typically miners pick transactions from the mempool based on maximizing the fees they collect from the block they produce. Simply put, the transactions that get included are the transactions which have the highest price per gas as this increases miner revenue. However, due to the discrete nature of blockchains, and the public nature of mempools, advanced users add a complication, called Maximum Extractable Value (MEV).

Maximum Extractable Value (MEV)[8]

The mempool is a public data structure, meaning anyone with an Ethereum Node can see all transactions as they are posted. Furthermore, because blockchains are inherently discrete in their progression, there will always be a "lag time" between a transaction's submission and its inclusion in the blockchain. This lag time period thus opens the door for malicious actors to read in a possible transaction from the mempool, submit the same transaction using their account addresses for payment, and "skip the line" to be included in the next block. This particular pattern is known as front-running and is only one of several known MEV exploits which have been observed on Ethereum and in its mempool. Fundamentally, this type of action is taking advantage of the fact that transaction position in a block has an impact on a transaction's value, and thus has an impact on the gas market as higher gas cost transactions will typically be included sooner in the block. In addition to simply increasing the gas price in a bid, advanced users partaking in MEV can also perform off-chain payments to miners directly, making their transaction inclusion in a given block at a particular position even more likely.

Other known MEV exploits include DEX Arbitrage, forced lending protocol liquidations, and Sandwich Trades. Each of these MEV strategies, and numerous others both known and unknown, introduce complexities that must be taken into account when using Ethereum. For the curious reader, the MEV paper Flash Boys 2.0 is a perfect starting place to learn more.

Conclusion

In this paper, gas on the Ethereum Network is presented at various levels of analysis. Beginning with a theoretical explanation of what gas is and why it is needed in Part I, it continues to the actual gas market implementation in Ethereum for both pre and post EIP-1559 regimes in Part II. Finally, Part III presents auctions as a mental model for how to think about gas on Ethereum. As a teaser, the mempool and maximum extractable value (MEV) are introduced to finish out the paper.

[8] Formerly known, and still known by some, as Miner Extractable Value (MEV)

Glossary

	Units	Definition
g_{size}	gas	size of an arbitrary block in the Ethereum mainchain
g^{h}_{size}	gas	size of a specific block at height h in the Ethereum mainchain
g_{max}	gas	maximum block size in Ethereum, a protocol level defined parameter, EIP-1559: 30 million
g_{target}	gas	the target block size in Ethereum, a protocol level defined parameter, EIP-1559: 15 million
g_{limit}	gas	user defined parameter indicating the maximum amount of gas they desire to spend during the execution of a given transaction
g_{used}	gas	amount of gas actually consumed in the execution of a transaction
$p_{baseFee}$	$\frac{Gwei}{gas}$	minimum price for gas for a transaction to be included in a block, determined algorithmically via EIP-1559 for each block with height $\geq 12,965,000$, see equation 2.
$p^{h}_{baseFee}$	$\frac{Gwei}{gas}$	$p_{baseFee}$ for a specific block with height h
$p^{7dAvg}_{baseFee}$	$\frac{Gwei}{gas}$	Seven day moving average of base fee for inclusion in a block
$p_{priorityFee}$	$\frac{Gwei}{gas}$	user defined transaction parameter indicating the price per gas a user is willing to pay to prioritize their transaction above the $p_{baseFee}$, user defined for $h \geq 12,965,000$
p_{bid}	$\frac{Gwei}{gas}$	user defined transaction parameter indicating their bid price for a unit of gas
p_{max}	$\frac{Gwei}{gas}$	user defined transaction parameter indicating the maximum price per gas a user is willing to pay, user defined for block height $\geq 12,965,000$
p_{paid}	$\frac{Gwei}{gas}$	the price paid for gas by a user, measured in Gwei, see Table 1
e_{max}	$Gwei$	the maximum possible cost of a translation in Ethereum, measured in Gwei, and calculated, $e_{max} = g_{limit} * p_{max}$
e_{cost}	$Gwei$	the actual cost of a transaction executed in Ethereum, $e_{cost} = p_{paid} * g_{used}$ and $0 \leq e_{cost} \leq e_{max}$