Transaction Fees on a Honeymoon: Ethereum's EIP-1559 One Month Later

Daniël Reijsbergen[1], Shyam Sridhar[2], Barnabé Monnot[2], Stefanos Leonardos[1], Stratis Skoulakis[1], Georgios Piliouras[1]

[1]Singapore University of Technology and Design (SUTD), [2]Ethereum Foundation

daniel_reijsbergen,stefanos_leonardos,efstratios,georgios@sutd.edu.sg, shyam.sridhar,barnabe.monnot@ethereum.org

Abstract—Ethereum Improvement Proposal (EIP) 1559 was recently implemented to transform Ethereum's transaction fee market. EIP-1559 utilizes an algorithmic update rule with a constant learning rate to estimate a *base fee*. The base fee reflects prevailing network conditions and hence provides a more reliable oracle for current gas prices.

Using on-chain data from the period after its launch, we evaluate the impact of EIP-1559 on the user experience and market performance. Our empirical findings suggest that although EIP-1559 achieves its goals *on average*, short-term behavior is marked by intense, chaotic oscillations in block sizes (as predicted by our recent theoretical dynamical system analysis [1]) and slow adjustments during periods of demand bursts (e.g., NFT drops). Both phenomena lead to unwanted inter-block variability in mining rewards. To address this issue, we propose an alternative base fee adjustment rule in which the learning rate varies according to an *additive increase, multiplicative decrease (AIMD) update scheme*. Our simulations show that the latter *robustly outperforms the EIP-1559 protocol* under various demand scenarios. These results provide evidence that *variable learning rate* mechanisms may constitute a promising alternative to the default EIP-1559-based format and contribute to the ongoing discussion on the design of more efficient transaction fee markets.

I. INTRODUCTION

Transaction fees have been an integral part of the emerging blockchain economies and the topic of heated debates in the blockchain community [2]–[4]. Ethereum, the second largest blockchain platform in terms of market value (as of September 6th, 2021), recently launched its long-awaited *London hard fork* which implements – among other protocol upgrades – Ethereum Improvement Proposal (EIP) 1559, which aims to radically transform its transaction fee market [5], [6].

In Ethereum's original fee market, users selected a *gas price* that indicated how much they were willing to pay for their transactions to be included on the blockchain. This mechanism behaved like a (generalized) first-price auction in which miners, the "auctioneers", allocated the available transaction slots in each block to the highest paying users or "bidders". Unfortunately, it also shared all of a first-price auction's drawbacks [7], [8]: untruthful bidding, excessive intra- and inter-block variation of transaction fees, and (over-)bidding competition for faster inclusion.

The EIP-1559 transaction market reform was put forward to address these shortcomings. After a period of consultation and implementation, EIP-1559 came into effect on August 5, 12:33:42 PM UTC, at block 12,965,000 [9], [10]. Its main goals are to allow for more flexibility during changing market conditions by aiming for a long-term average target block size (half-full blocks) instead of blocks that are consistently full, and to make transaction fees more predictable (but not necessarily lower).

To achieve these goals, EIP-1559 sets a *base fee* that is algorithmically updated after each block to reflect the prevailing market demand. Critical in these updates is a hard-coded learning rate (currently set at 12.5%) which calibrates the size of the adjustment according to the current block size. Unlike the traditional formats in which all fees were transferred from users to miners, the base fee is now *burnt* [11], i.e., permanently removed from the total supply of Ethereum tokens (ether or ETH). Alongside newly minted tokens, miners are compensated for their work by an additional *tip* set by the users, typically much lower than the base fee.

Motivation and Contributions: Due to Ethereum's prominence in the blockchain ecosystem, EIP-1559 naturally attracted widespread academic and community attention even before its launch. In [4], [12], its economic properties were studied, proving that it is *incentive compatible* in the sense that it is rational for users to follow a simple bidding strategy whenever the current base fee is comparable to market demand, i.e., does not under-price the limited supply of gas in a block. Meanwhile, [1], [13] analyzed its dynamics under different demand scenarios and [3], [14] initiated the discussion about alternative (EIP-1559-like) transaction fee market mechanisms.

In this paper, we contribute to this growing literature on EIP-1559. We use blockchain data from the first month after EIP-1559's launch – the "*honeymoon*" period during which legacy transactions are still accepted – to evaluate EIP-1559's impact on user experience and network performance. We use this empirical data to simulate and argue about the performance of EIP-1559 under different learning rate configurations. Our contributions can be summarized as follows.

- We find that the current learning rate leads to slow adjustments during periods of demand peaks (e.g., NFT [15] drops) and to intense oscillations in block sizes during periods of stable demand.
- Both slow adjustments and oscillations have a negative effect on the user experience and, more importantly, on mining rewards. In particular, they can potentially generate strategic incentives for miners and hence undermine the stability of the mechanism.
- We then use the empirical data to run simulations with different learning rates. Our results confirm the intuition that higher (lower) learning rates perform well (poorly) during demand peaks and poorly (well) during periods

of demand stability. On the positive side, all EIP-1559 variations achieve (in fact, slightly overshoot) the target block size (50%) on average.

- Based on the above, we propose an alternative mechanism that utilizes an additive increase, multiplicative decrease (AIMD) update scheme to implement a variable learning rate. Our empirical results suggest that the AIMD mechanism combines the best of both worlds: it selects a low learning rate during periods of stable demand and reacts quickly during demand peaks to reflect increasing gas prices. The AIMD not only achieves the target block size on average (half-full blocks) but also considerably dampens the oscillations between almost full and almost empty blocks, which leads to more stable mining rewards and a more uniform user experience.

While our empirical results do not prove the optimality of the proposed AIMD mechanism under arbitrary conditions, they provide evidence that variable learning rates may considerably improve upon the current EIP-1559 format and may thus constitute a promising direction for future work in the ongoing discussion about EIP-1559-based transaction fee mechanisms [3], [14].

Outline: The rest of the paper is structured as follows. In Section II, we present the main elements of Ethereum's blockchain protocol and EIP-1559. In Section III, we describe our dataset and in Section IV our simulation environment. Section V contains the evaluation of the simulation results. Section VI concludes the paper with questions and directions for future work.

II. PRELIMINARIES

A. Ethereum

Ethereum has a prominent position in the growing blockchain ecosystem, as the first platform to support decentralized, self-executing pieces of software known as *smart contracts* [16].

Block Creation: The creation and execution of smart contracts on the Ethereum blockchain is achieved through *transactions* which are ordered and grouped into *blocks*. Whenever a new smart contract is deployed or a function of an existing contract is called, the operation is embedded in a transaction. The transaction is then broadcast to consensus nodes – which in Ethereum's current *proof-of-work* mechanism are called *miners* – through a peer-to-peer network [17]. The miners vie for the right to create new blocks for which they earn *block creation rewards* [18]. Eventually, a new block that includes a set of new, previously pending transactions is created and broadcast to the nodes of the peer-to-peer network, who then update their view of the blockchain. Valid blocks that are not propagated across the network quickly enough can be superseded by a competing block – the losing blocks are called *uncles*.

Transaction Fees: The maximum size of the blocks that encapsulate these transactions is limited as per the specifications of the protocol and is measured in terms of the total amount of computational effort required for executing all transactions in that block. It is chosen as a trade-off between 1) higher throughput and 2) making it possible for more low-power nodes to participate. Thus, there is a demand for block space among users that wish to get their transactions included in the blockchain, which is supplied by the miners who create the blocks. To find a balance between this supply and demand, users must pay a *transaction fee* to the miners for including their transaction in a block. This provides a second form of protocol-supported payment for the miners, alongside the block creation rewards.

Ethereum Gas: The unit that measures the amount of computational effort required to execute transactions on the Ethereum blockchain is referred to as *gas*. For example, creation of a new smart contract may require $500,000$ units of gas, but a simple transfer of ETH tokens typically requires only $21,000$ units of gas. The use of gas units (or simply gas) allowed for a uniform implementation of transaction fees in Ethereum's original fee market design: for each transaction, the user would specify a *gas price* that indicates the fee paid by the user per unit of gas used to execute the transaction, and a *gas limit* that indicates the maximum amount of gas that the transaction is able to consume.

Ethereum's Legacy Fee Market: Ethereum's legacy fee market mechanism, thus, closely resembled that of a *(generalized) first-price auction*. Users would publish a transaction with a fee, miners would choose the highest-paying transactions, and each user would pay what they bid. The higher the user was willing to pay in transaction fees, the higher their chances of having their transactions included in a block. However, such auctions are well-known to be highly inefficient, leading to extreme fee volatility and frequent over-payments [8], [19]. This is especially observed when the network's transaction load is high, e.g., due to new project launches or periods of token price volatility. To remedy these shortcomings, EIP-1559 was proposed [7].

B. EIP-1559 Transaction Fee Market

EIP-1559 is an upgrade on Ethereum's design which aims to address the above challenges by reforming the transaction fee market. With its introduction on August 05, 2021 on Ethereum's mainnet, the maximum block size was doubled from roughly 15 million to $T = 30$ million gas. However, the long-term average block size that the system targets is half that limit, namely $T/2$.

Base Fee: The main element of EIP-1559 to achieve this target is a dynamically adjusted *base fee*, b_t, that is updated after every block, indexed by block height $t > 0$, according to the equation

$$b_{t+1} = b_t \left(1 + d \cdot \frac{G_t - T/2}{T/2} \right). \qquad (1)$$

Here, d denotes the *learning rate* (or *adjustment parameter*), currently set at $d = 0.125$, and G_t the total

gas used by transactions that were included in block $t > 0$. Intuitively, if $G_t > T/2$, i.e., if more transactions than targeted are included in a block, then the base fee increases. The increase is scaled by d and is proportional to the amount by which the current block load G_t exceeds the target $T/2$. Similarly, if $G_t < T/2$, then the base fee decreases, whereas if $G_t = T/2$, then the base fee remains unchanged.

Bids, Tips, and Effective Gas Prices: According to EIP-1559, instead of a single gas price, users now submit two parameters, (f, p), as shown in Table I.

Parameter	Description
f: max fee	maximum amount per gas unit that the user is willing to pay for their transaction to be included
p: max priority fee	maximum tip per gas unit that the user is willing to pay to the miner who includes their transaction

Table I: EIP-1559 user bid parameters.

If the max fee that a user is willing to pay for their transaction is less than the algorithmically computed base fee, i.e., if $f < b_t$, then the transaction cannot be included in a block. Conversely, if $f \geq b_t$, then the miners can include the transaction and hence earn the so-called *miner's tip*, which is calculated as miner's tip $= \min\{f - b_t, p\}$ In practice, if the miner's tip is very small (less than 2 Gwei per gas unit, where 1 Gwei $= 10^{-9}$ ETH), it may be beneficial for miners not to include the transaction. Larger blocks are harder to transmit over the network, making it easier for a competing block to reach other nodes faster, causing the miner to lose out on the block creation rewards and fees ("uncle risk").

Importantly (and in sharp contrast to the original fee market), the base fees are not transferred to miners: instead they are *burnt* and are permanently removed from the total supply of ETH.[1] The sum of the base fee and tip fee taken per unit gas is said to be the *effective gas price* and is equivalent to the gas price that a user would have paid to include their transaction before EIP-1559.

Legacy Transactions: EIP-1559 was introduced in a backward-compatible manner allowing both *legacy* and native EIP-1559 transaction formats to be included. However, all transactions included after block 12,965,000 are subject to the base fee, regardless of their format. In particular, when a legacy transaction is submitted to the network, the max priority fee, p, and the max fee, f, are set by default equal to the gas price specified by the user. The miner then earns $f - b_t$ if the transaction is included in their block, possibly leading to overpayment at inclusion, compared to the same transaction following the simple EIP-1559 bidding strategy. With an EIP-1559 transaction, the user places a cap on the miner's tip by setting a max priority fee, whereas in a legacy transaction all the remaining fees, after subtracting the base fee, go to the miner.

[1]As of September 3rd, 2021, over 180,000 ETH were burned through EIP-1559 transactions (source: watchtheburn.com).

III. DATA

To analyze the impact of EIP-1559 during the honeymoon period, we use publicly available data recorded on the Ethereum blockchain. Our dataset covers a period starting from block $l = 12935000$, which was mined on 3 August 2021, 18:43, until block $u = 13079999$, which was mined on 23 August, 06:38. EIP-1559 came into effect in block 12965000, which was mined on 5 August 2021, 12:33 (the mentioned times are all in UTC). Our data set is described in detail in Table II and visualized in the next paragraphs, which a focus on both "long-term" trends (i.e., observations from the entire study period) and "short-term" behavior (i.e., observations regarding sudden demand peaks or other transient phenomena).

Parameter	Description
$x \in \mathbb{N}$	transaction, identified by its hash
$t_x \in \mathbb{N}$	height of the block in which x was included
$g_t \in [0, 1]$	*relative* block size of the block at height t, i.e., the total used gas G_t divided by the gas limit T
$b_t \in \mathbb{N}$	base fee in the block at height t
$\tau_x \in \{0, 1\}$	type of x, which is "0" for legacy transactions and "1" for EIP-1559 transactions
$f_x \in \mathbb{N}$	gas price of x for legacy transactions, and the max fee of x for EIP-1559 transactions
$p_x \in \mathbb{N}$	priority fee of x (for legacy transactions this is set equal to f_x)
$u_x \in \mathbb{N}$	gas limit of transaction x (used as a proxy for the gas that was actually used by the transaction)
$X(t) \subset \mathbb{N}$	set of all transactions included in block at height t

Table II: Transaction information in the dataset.

A. Long-term Trends

Average Gas Prices: First, we study the *average block gas prices* during the sample period. To compute this metric over time, we first compute the average gas price $\hat{f}_t$ for each block at height t, where each transaction x is weighted by its gas limit u_x:

$$\hat{f}_t = \left(\sum_{x \in X(t)} u_x f_x\right) / \left(\sum_{x \in X(t)} u_x\right).$$

We then divide the set $\{l, l+1, \ldots, u\}$ into 1000 batches of size $w = \frac{u-l+1}{1000}$, and compute the average $\hat{f}_i^*$ for each batch $i \in \{1, \ldots, 1000\}$ as

$$\hat{f}_i^* = \frac{1}{w} \sum_{t=l+(i-1)w}^{l+iw} \hat{f}_t. \tag{2}$$

The result is displayed in Figure 1, which shows the evolution of the average block gas, $\hat{f}_i^*$.

We observe that the (average) gas prices exhibit sharp bursts. We have found that these bursts typically correspond to so-called "NFT drops", in which a creator (or project) announces the release of a limited number of Non-Fungible Tokens (NFTs) [15]. This leads to a demand surge where purchasers put in high bids to be included first by miners. Other, lower-scale fluctuations are due to diurnal variations in Ethereum's usage over different geographic locations. In Section III-B, we return to the base fee and gas price dynamics and examine them in the
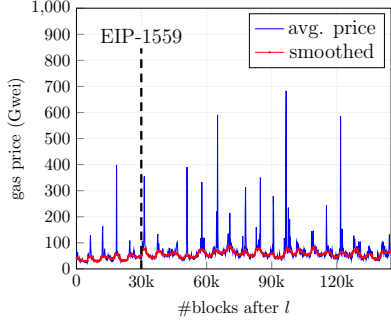
(a) Gas price
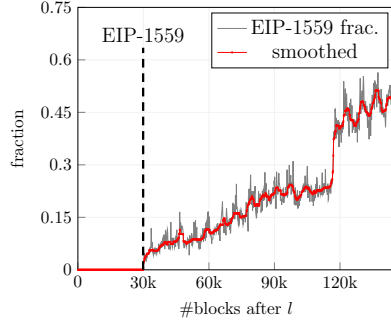
(b) Fraction of EIP-1559 transactions

Figure 1: Top: average (blue) and smoothed averaged (red) gas prices over time. Bottom: average fraction (gray) and smoothed averaged fraction (red) of EIP-1559 transactions over time.

locality of periods of both sharp demand increases and of relatively stable demand.

Smoothed Average Gas Prices: To dampen the effect of the sharp bursts on the long-term trend of gas prices, we apply a median smoothing filter to $\hat{f}_i^*$, which is unaffected by large outliers unlike the moving average. To compute the filtered value $m(\hat{f}_i^*, \eta)$ of $\hat{f}_i^*$, where η is the half-width of the filter, we first determine the set

$$V(\hat{f}_i^*, \eta) = \{\hat{f}_j^* \in \mathbb{N} : j \geq l, j \leq u, |j - i| \leq \eta\}.$$

We then compute the median $m_i = m(\hat{f}_i^*, \eta)$ as the value such that at least 50% of the elements in $V(\hat{f}_i^*, \eta)$ are greater or equal than m_i and at least 50% of the elements in $V(\hat{f}_i^*, \eta)$ are smaller or equal than m_i. The result with half-width $\eta = 10$ is displayed by the red line in Figure 1.

Fraction of EIP-1559 Transactions: We apply the same methodology as above to the weighted fraction, $\hat{\tau}_h$, of EIP-1559 transactions (versus legacy transactions) which for the block at height t is given by

$$\hat{\tau}_t = \left(\sum_{x \in X(t)} u_x \tau_x\right) / \left(\sum_{x \in X(t)} u_x\right).$$

The result is displayed in Figure 1b. The gray line represents $\hat{\tau}_i^*$ averaged, as in equation (2), over 1000 batches and the red line represents the corresponding smoothed

values. The i-th value in the plot corresponds to the middle block of the i-th batch for $i \in \{1, 2, \ldots, 1000\}$.

We observe that EIP-1559 adoption increases steadily within our sampling period to around 60% at the end of August. The sharp increase around block 13051500 (8 August) corresponds to the (actual date of) adoption of EIP-1559 by the popular Ethereum wallet MetaMask. After that point, we observe strong diurnal fluctuations as a result of differences in wallet usage over different geographic locations.

B. Short-term Behavior

We now take a more detailed view at the short-term behavior of the base fee and gas price dynamics during periods of both sharp demand increases (e.g., NFT drops) and of relative demand stability.

Demand peaks: Figure 2a zooms in on one of the price bursts that are visible in Figure 1, namely the one that occurs around block 13025775 that was mined on August 14, 2021, at 09:35. The displayed period covers 450 blocks between blocks 13025550 and 13026000.

Before the burst, average gas prices (depicted by the blue line without marks) fluctuate between 30 and 60 Gwei, with several short bursts in which the average fee jumps to around 120 Gwei before returning to around 45 Gwei, and several empty blocks (in which the average fee is set to 0). The average gas price then shoots from 40.82 Gwei in block 13025736 to 207.33 and ultimately to 1360.02 Gwei within the next two blocks.

To compare, the miner of block 13025736 earns a total of 2.019 ETH: 2 ETH for the block creation reward, plus 0.137 in transaction fees minus 0.118 in burnt fees. In this case, as is a typical for blocks outside price bursts, the transaction fees are small compared to the block rewards [20]. However, the miners of blocks 13025737 and 13025738 earn 3.472 ETH and 45.646 ETH, respectively. In the latter case, the earnings from transaction fees are more than 20 times greater than the block creation reward!

Concerning the base fee (depicted by the brown line with marks), we observe that shortly after the price burst it increases rapidly until it reaches a level slightly below the average gas price. After that point, it starts to decline together with gas prices which gradually return to normal. In sharp contrast to the above, block 13025768 is ultimately far from target, netting 0.844 ETH in transaction fees for 20% block utilisation (10% of the block gas limit). Comparing this revenue with that of the next miner 13025769, who received 4.868 ETH for 180% block utilisation (90% of the block gas limit), 13025768 misses out on some share of these fees. With a uniform process generating the transaction arrival time, this share may be estimated to 2 ETH (an equal split between 0.844 and 4.868 ETH), worth around $\$6,500$ USD at the time. These points demonstrate an instability in inter-block mining rewards under EIP-1559 that may require closer monitoring.

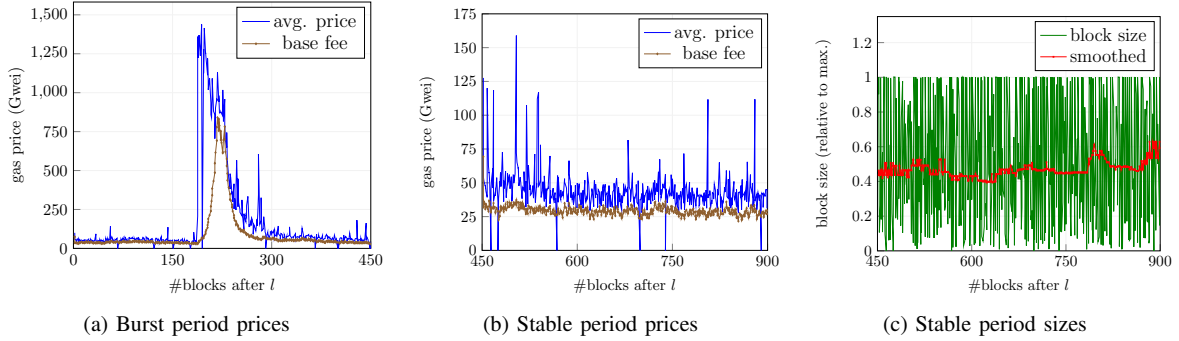(a) Burst period prices   (b) Stable period prices   (c) Stable period sizes

Figure 2: (a) Average gas prices (blue) and base fees (brown) per block in the volatile period after block 13025550. (b) Average gas prices (blue) and base fee per block (red) in Gwei, in the period of relatively stable demand that directly followed the burst of (a). (c) Relative block sizes, $\hat{g}_t$, in the same period of relatively stable demand as in (b).

Demand Stability: Figure 2b displays the gas price and base fee dynamics for the 450-block period of relatively stable demand that starts immediately after the burst of Figure 2a. During this period, both the gas price and the base fee remain roughly stable: the former at around 40 Gwei and the latter slightly below 30 Gwei. However, despite the fact that the base fee remains within a relatively narrow range of 10 Gwei, i.e., between 25 and 35 Gwei, block sizes fluctuate wildly between the two extremes (empty to full and vice versa). This can be seen in Figure 2c, where the green line (without marks) displays the relative block size as a number between 0 (empty) and 1 (full). This is exactly the behavior predicted in [1], [13], and can be attributed to the discrete nature of the updates and the size of d.

However, it is interesting to observe that block sizes do, in fact, seem to reach (or, actually, only slightly overshoot) the target block size *on average*. As shown in Figure 2c, the median filter with half-width 30 applied to $\hat{g}_t$ (red line with marks) is indeed centered around 0.5 (half-full blocks). Providing theoretical reasons for or empirically testing this observation over longer periods of data is an interesting direction for future work.

IV. SIMULATIONS: LEARNING RATE IN EIP-1559

As established by the short-term analysis in the previous section, there are two opposing effects concerning the learning rate d. In particular, the default d (set at 0.125) is *too low* during periods of short bursts (during which the base fee lags in catching up to actual demand) but is also *too high* during periods of relative stability (where it induces intense oscillations in block sizes despite keeping the base fee relatively stable). While long-term trends suggest that block-sizes achieve the target of half-full blocks *on average*, these fluctuations have significant effects on miners' earnings and thus, potentially, on protocol stability.

As such, we investigate the impact of choosing a different d (constant or variable) on the network. To do so, we run a series of simulations in which we observe the base fee after transaction bids are drawn randomly.

A. Constant Learning Rate

The main challenge in simulating (i.e., replaying using a stochastic model) the evolution of the transaction fee market for different values of d (or alternative mechanisms) is to draw accurate estimates for the actual demand and the users' valuations from the available data. The difficulty stems from the fact that observable bids – i.e., those included on the blockchain – do not necessarily reflect users' true valuations, but merely their valuations/expectations about the current and future base fees and their urgency to get their transaction included. Furthermore, they do not include low-value bids that were not included.

Behavioral Model: As such, we make some assumptions – justified by our current data and Ethereum's previous simulations, see [1], [6], [13] – and leave a deeper investigation of bids, e.g., with questionnaires to users, as an open challenge for future research. Accordingly, we model the number of users in each time slot using a Poisson demand with arrival rate λ and a distribution of valuations, F. As mentioned above, both λ and F are subject to change over time – e.g., due to the effects of weekends and weekdays, day/night cycles in different regions, sudden bursts due to popular contracts and NFT launches – and therefore challenging to infer from the available data.

Execution Model: Given a Poisson distribution of transaction arrivals (i.e., demand) with time-dependent arrival rate λ_t and valuation distributions F_t, the simulator executes the steps of Algorithm 1 in each time slot t.

Algorithm 1 first calls draw_num_transactions(λ_t), which pseudo-randomly draws a value n_t from the Poisson distribution with mean $\lambda_t \equiv 3$ (adopted from [1]). The function draw_transactions($n_t, \hat{f}_t$) then draws n_t samples from F_t that represent the valuations of the transactions created at time t, given the observed median-smoothed average gas price in the block at height t, denoted by $\hat{f}_t$. These values are then stored in the set m'_t. We have observed that a mixture of uniformly distributed and Pareto-distributed random samples, with means that depend on the smoothed average fees, results in a good

---

Algorithm 1: Simulation of EIP-1559 Updates

Input: Demand parameters λ_t, $\hat{f}_t$, base fee b_{t-1} at block height $t-1$ and learning rate d.

Output: base fee at block height t.

1   $n_t \leftarrow$ draw_num_transactions(λ_t)
2   $m'_t \leftarrow$ draw_transactions$(n_t, \hat{f}_t)$
3   $m_t \leftarrow$ m_update(m_{t-1}, m'_t)
4   $B_t \leftarrow$ B_update(m_t)
5   $g_t \leftarrow |B_t|/T$
6   $b_t \leftarrow$ b_update$(b_{t-1}, d, |B_t|)$

Parameter	Description
g_{AVG}	average relative block-size of previous blocks
$n \in \mathbb{N}$	nr. of previous blocks over which g_{AVG} is computed.
$\alpha > 0$	size of an additive increase
$\beta \in [0, 1]$	factor of a multiplicative decrease
$\gamma \in [0, \frac{1}{2}]$	threshold for an additive increase
$d_{\min}, d_{\max}$	minimum and maximum values for d.

Table III: Parameters of the AIMD algorithm.

correspondence with the observed values' distribution. We then draw the valuations of $2.75 n_t$ transactions from a uniform distribution near $0.75 \hat{f}_t$, and draw the remaining $0.25 n_t$ valuations from a Pareto distribution with shape 1.35 and scale $\frac{1}{10} \hat{f}_t$. The reasoning behind this modelling choice is to divide users into two broad classes. The first class consists of *regular* users for whom the delay before the transaction appears on the blockchain is relatively unimportant, and who simply bid a value that (they believe) ensures that their transaction is eventually included. The second class consists of *high-urgency* users who make high bids to ensure that their transaction is included rapidly, and whose valuations more closely resemble a heavy-tailed distribution. The parameters were chosen to give a good correspondence with the studied dataset, but we have found that minor variations did not change the overall conclusions of our evaluation, cf. Table V.

After having drawn m'_t, Algorithm 1 combines the transactions in m'_t with the process m_t, which represents the *mempool*, i.e., the set of transactions that are pending to be included in the blockchain. The function m_update(m, m') returns a new mempool given an existing mempool m and a set m' of new transactions. In our simulations, transactions remain in the mempool until they are included. The function B_update(m) is then executed by the miner. B_update(m) returns a vector B with the max fees of the transactions that have been included in the new block given the mempool m. We denote the length of B as $|B|$. Finally, the function b_update$(b, d, |B|)$ returns the new base fee as a function of the current base fee b, the learning rate d and the block size $|B|$.

B. Variable Learning Rate: AIMD Updates

To address the issue that the current learning rate appears too high in stable-demand periods and too low in periods with demand peaks, we also propose and simulate a mechanism with a variable learning rate. In particular, motivated by the effectiveness of such schemes to control internet congestion [21], we study an *additive increase and multiplicative decrease* (AIMD) update scheme for the learning rate d.

According to the AIMD scheme, the learning rate is adjusted with respect to a parameter, g_{AVG}, that keeps track of the average relative block sizes ($g_t = G_t/T$, cf. Table II) during some pre-specified rolling window. The

intuition is the following: when g_{AVG} is close to the target block-size, 0.5, then the base fee is close to the right value, so the algorithm reduces the learning rate to reduce the size of the oscillations. By contrast, if g_{AVG} is too small or high, then the base fee is apparently far away from its equilibrium value, and the algorithm increases the learning rate.

Execution Model (AIMD): The AIMD update scheme uses the parameters in Table III. It then replaces line 6 in Algorithm 1 with the following two lines which update

---

Algorithm 2: Modified Alg. 1 with AIMD updates

---

6   $d_t \leftarrow$ d_update$(d_{t-1}, g_{t-n+1}, \ldots, g_t)$
7   $b_t \leftarrow$ b_update$(b_{t-1}, d_t, |B_t|)$

---

the learning rate parameter d according to Algorithm 3 (following the intuition described above):

Algorithm 3: AIMD update routine d_update.

Input: Learning rate, d_t, window n, block-sizes $g_{t-n+1}, \ldots, g_t$, update parameters α, β, γ.

Output: Learning rate, d_{t+1}.

1   $g_{\text{AVG}} \leftarrow \frac{1}{n} \sum_{t-n+1}^{t} g_i$
2   **if** $g_{\text{AVG}} < \gamma$ **or** $g_{\text{AVG}} > 1 - \gamma$ **then**
3     $d_{\text{NEW}} \leftarrow \min\{d_{\max}, \alpha + d\}$
4   **else**
5     $d_{\text{NEW}} \leftarrow \max\{d_{\min}, \beta d\}$
6   **return** d_{NEW}

V. EVALUATION

We now proceed to compare the different update mechanisms described in the previous section. We present simulations of the standard EIP-1559 update rule with constant step sizes at $d = 0.125$ (default), $d = 0.25$ and $d = 0.0625$ which are representative of high and low rates, respectively, and of the AIMD (variable step size) update rule with parameters $n = 8, \alpha = 0.025, \beta = 0.95, \gamma = 0.25, d_{\min} = 0.0125$ and $d_{\max} = 1$. Note that n is selected to be low enough to reflect recent market conditions, but high enough to reduce its sensitivity to temporal fluctuations in demand.

A. Performance Metrics

To compare the performance of the different mechanisms, we use the two metrics described in Table IV.
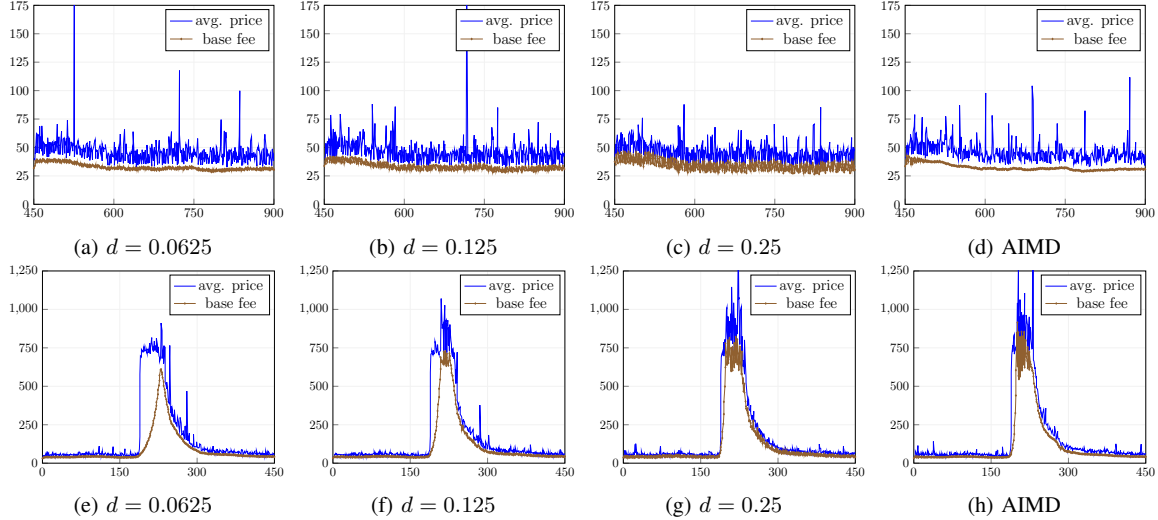
Figure 3: Simulation of average gas prices in Gwei (blue lines) and base fees (red lines) in stable (top) and peak demand periods (bottom) of the EIP-1559 market under different learning rates. Unlike standard EIP-1559 mechanisms with slow or fast learning rates ($d = 0.125$ being the default), the AIMD mechanism exhibits good performance in both scenarios.

Parameter	Description
$\hat{g}$	long-term average relative block size
$p_{g>0.95}$	long-term fraction of blocks whose relative size is greater than 0.95.

Table IV: Performance metrics.

The first metric, $\hat{g}$, should be as close to 0.5 as possible, as this is the target set by the miners. When $\hat{g}$ is (considerably) larger than 0.5, propagation and execution time of new blocks increase, putting stress on the network of miners to validate all new items. Conversely, when $\hat{g}$ is (considerably) lower than 0.5, network capacity is under-utilized. The situation where $\hat{g}$ is slightly below 0.5 more closely resembles the original target setting of the Ethereum network before EIP-1559 with max block size of $T/2$ making such states marginally preferable over ones where $\hat{g}$ is slightly above 0.5.

The second metric, $p_{g>0.95}$ should be as low as possible, even if $\hat{g}$ approximates 0.5 (e.g., due to full blocks counterbalancing empty blocks). When block-sizes are consistently high, the transaction fee market approaches the behavior of a first-price auction, losing some of the advantages of EIP-1559 [12].[2]

B. Simulation Results

The effect of choosing different learning rates in the standard EIP-1559 mechanism is displayed in Figure 3. We visualize two scenarios: the stable period from Figure 2b (top row), and the NFT drop from Figure 2a (bottom row). The blue lines show the simulated demand

[2]The reason that we do not consider $p_{g=1}$ is purely technical: since not all transaction have the same size, in some cases a block will only use, say, 99% of its gas and yet it will still be considered "full" in the sense that no other available transactions can be added to it.

which approximates the actual demand, cf. Figures 2a and 2b.

As intuitively expected, the oscillations of the base fee (brown line with marks) in the stable scenario decrease as the learning rate decreases. Similarly, in the NFT drop scenario, the fast learning rate mechanism, $d = 0.25$, swiftly adapts to the changing situation, whereas the other two learning rate mechanisms generate long periods of discrepancies between the base fee and market demand.

The important takeaway, however, concerns the AIMD mechanism (panels in the last column), which turns out to share the advantages of both approaches. In particular, when the system is relatively stable, it selects a learning rate of $d = 0.0125$ (lower than the current default of $d = 0.125$) that increases swiftly during demand peaks. This results in low oscillations of the base fee during the stable period (top row) and a timely increase during the demand peak period (bottom row). To emphasize this, we have also displayed in Figure 4 the block sizes and the evolution of the learning rate d for the AIMD scheme. We set $d = 0.125$ as the initial value, and observe that in both scenarios the learning rate decreases to its minimum values, $d_{\min} = 0.0125$. However, in the NFT drop scenario, the learning rate rapidly increases to over 0.45, and then decreases again when the base fee starts to reflect market conditions. In the stable scenario, we observe from Section V-C that although there are still oscillations, they are noticeably less intense than in Figure 2c.

To further quantify this performance, Table V contains the results for both metrics, $\hat{g}$ and $p_{g>0.95}$, over three scenarios: the stable period (450 blocks), the demand peak period (450 blocks), and our full dataset of blocks after EIP-1559 (115,000 blocks).

In Table V, below each point estimate of $\hat{g}$ and $p_{g>0.95}$

Stable periods		$\hat{g}$	$p_{g>0.95}$	Demand peaks		$\hat{g}$	$p_{g>0.95}$	Full dataset		$\hat{g}$	$p_{g>0.95}$
	empirical	0.523	0.261		empirical	0.528	0.281		empirical	0.523	0.244
simulation	$d = 0.0625$	0.505 ±0.000	0.150 ±0.005	simulation	$d = 0.0625$	0.511 ±0.000	0.222 ±0.004	simulation	$d = 0.0625$	0.505 ±0.000	0.186 ±0.000
	$d = 0.125$	0.510 ±0.000	0.214 ±0.004		$d = 0.125$	0.513 ±0.000	0.232 ±0.004		$d = 0.125$	0.510 ±0.000	0.232 ±0.000
	$d = 0.25$	0.520 ±0.000	0.255 ±0.002		$d = 0.25$	0.522 ±0.000	0.265 ±0.003		$d = 0.25$	0.520 ±0.000	0.264 ±0.000
	AIMD	**0.499** ±0.001	**0.058** ±0.003		AIMD	**0.481** ±0.002	**0.126** ±0.003		AIMD	**0.501** ±0.000	**0.047** ±0.000

Table V: Empirical evaluation of the different EIP-1559 mechanisms with low, default ($d = 0.125$), and high learning rates in terms of the performance metrics $\hat{g}$ (average block size) and $p_{g>0.95}$ (probability that a block is more than 95% full). The AIMD mechanism has the best performance in terms of $p_{g>0.95}$ in both the stable and unstable scenarios, and across the entire post-EIP-1559 dataset (bold values).

we also display the half-width of a 95% confidence interval for the corresponding value based on the normal distribution. For each entry in the table, we conducted 20 simulation experiments – despite the small sample size, we note that the variance is fairly low since each sampled estimate of $\hat{g}$ and $p_{g>0.95}$ is itself an average over at least 450 blocks. For each scenario, we also display the *empirical* result, i.e., the values of $\hat{g}$ and $p_{g>0.95}$ that were observed in the data set.

Importantly, the AIMD has a low number of full blocks in all three scenarios and hence exhibits the best performance in terms of $p_{g>0.95}$ by a wide margin when compared to the other methods. AIMD updates also achieve the desired average relative block size ($\hat{g} \approx 0.5$). Consistent with our empirical observations, all standard EIP-1559 updates also approximate, yet, consistently overshoot by a small margin, the target block-size on average, but perform poorly on the $p_{g>0.95}$ metric. This suggests the presence of more than 20% almost full blocks counterbalanced by a similar amount of almost empty blocks.

C. Discussion

Given the desirable properties showcased by the AIMD over the standard EIP-1559 update rules in the above simulations, it is natural to ask whether the AIMD update rule could actually fit in the current scope of EIP-1559. In particular, the default, more moderate EIP-1559 update that corresponds to a fixed $d = 0.125$ aims to achieve the following goals: (i) generate more predictable fees against comparable market conditions, (ii) adapt fees to prevailing demand, (iii) fight inflation by burning excessive fees instead of transferring them to miners, and (iv) tame strategic behavior by protocol participants, both miners and users. In the following, we discuss how an AIMD update rule would perform along these axes.

Predictability and Adaptation: Concerning the first two goals, there is a certain amount of trade-off between short-term fee predictability and swiftly responding to changing market conditions. However, the high-level goals of next-block inclusion and invariant block sizes tip the scale towards more responsive fee mechanisms.

Excessive Fees: Concerning the goal of controlling inflation, it is tempting to think that a slow-adapting mechanism may burn more fees by allowing full blocks during demand peaks. However, users' valuations are typically skewed towards higher values [2], [22] and higher fees at periods of increased demand can actually reap higher amounts of burned coins.

Strategic Behavior: The fourth goal, that of mitigating strategic behavior by miners or users, is more intricate and merits a separate discussion. Apart from general attacks on the PoS consensus layer [23], [24], there is an open debate on the vulnerabilities of the EIP-1559 fee market (in its original format) against exacerbated versions of existing or novel attack vectors [18]. One such example is the *zero-base fee attack*, in which a coalition of miners who control 51% of the network's resources collectively drive the base fee down to zero by continuously mining empty blocks. At that point the only determining factor for transaction inclusion are miner's tips, essentially moving the system back to the previous first price auction model. While the practical feasibility of such an attack is still unclear, it is a necessary to look at the effects of the more responsive AIMD update rule on miner's incentives via a more comprehensive threat model. We leave this as a direction for future work.

VI. CONCLUSIONS & OPEN QUESTIONS

Using on-chain data from the first month after its launch, we studied the early stages of Ethereum's EIP-1559 fee market upgrade. We found evidence that the current fixed learning rate – the parameter that is used to adjust the base fee after each block based on network congestion – achieves the target block-size on average (or rather, slightly overshoots it). However, it may have a considerable destabilizing effect on mining earnings – via slow adjustments during demand bursts and intense oscillations in block occupancies during periods of stable demand – and on the user experience. Such phenomena may lead to strategic behavior and ultimately pose a threat to the stability of the fee market.

Our simulations suggest that a promising direction to improve upon the current EIP-1559 format is the use of variable learning rates. In particular, we studied an additive increase, multiplicative decrease (AIMD) update scheme for the learning rate that resulted in superior performance over standard EIP-1559 formats across periods of both demand bursts and demand stability.

Given this range of results, a number of interesting open questions emerge. Specifically, it will be instructive
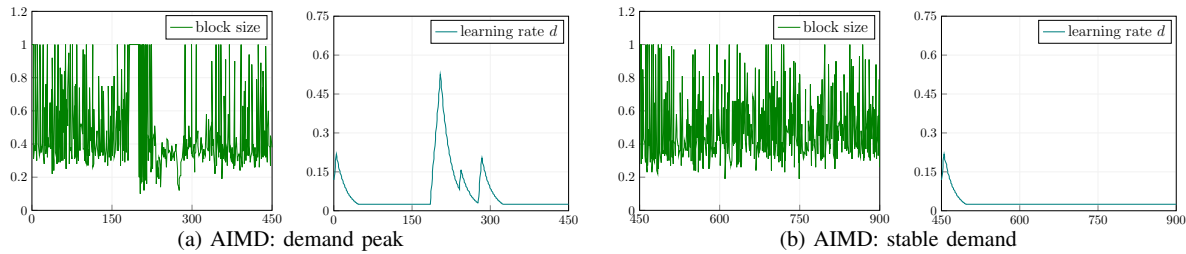
Figure 4: Block sizes and learning rate evolution under the variable learning rate AIMD scheme during the demand peak (first and second panel) and stable demand periods (third and fourth panels), respectively. AIMD adjusts the learning rate during the demand peak and uses a lower than default learning rate to dampen the oscillations (in particular, empty or almost empty blocks) during the stable demand period (cf. EIP-1559 performance in Figures 2a and 2b.

to monitor the impact of EIP-1559 over longer time spans (i.e., beyond the "honeymoon" period), to study different learning rate update schemes (e.g., multiplicative increase, additive decrease, or exponential updates) and to derive accurate estimates of Ethereum's recurring demand conditions to systematically optimize the parameters of the proposed AIMD scheme. Finally, as mentioned above, it will be interesting to develop a strategic threat model to analyze how miners' and users' incentives change between the default EIP-1559 mechanism and the proposed AIMD update rule. Any progress in the above directions will contribute to the ongoing discussion about EIP-1559 and to the design of more efficient transaction fee mechanisms in general.

ACKNOWLEDGMENT

REFERENCES

[1] S. Leonardos, B. Monnot, D. Reijsbergen, S. Skoulakis, and G. Piliouras, "Dynamical Analysis of the EIP-1559 Ethereum Fee Market," in *ACM AFT'21*, 2021.

[2] D. Easley, M. O'Hara, and S. Basu, "From mining to markets: The evolution of Bitcoin transaction fees," *Journal of Financial Economics*, vol. 134, no. 1, pp. 91–109, 2019.

[3] M. X. Ferreira, D. J. Moroz, D. C. Parkes, and M. Stern, "Dynamic Posted-Price Mechanisms for the Blockchain Transaction-Fee Market," in *ACM AFT'21*, 2021.

[4] T. Roughgarden, "Transaction Fee Mechanism Design," in *EC'21*, New York, NY, USA, 2021, p. 792, [arXiv].

[5] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norden, and A. Bakhta, "EIP1559: Fee market change for ETH 1.0 chain," [Online], 2019.

[6] Robust Incentives Group, "abm1559 notebooks," [Online], 2021.

[7] V. Buterin, "EIP-1559 FAQ," [Online], 2021.

[8] E. Maskin, "Auctions and Efficiency," in *Advances in Economics & Econometrics*. Cambridge University Press, 2001, pp. 1–24.

[9] T. Beiko, "London Mainnet Announcement," [Online], 2021.

[10] Ethereum Foundation, "EF-Supported Teams: Research & Development Roundup," [Online], 2021.

[11] K. Karantias, A. Kiayias, and D. Zindros, "Proof-of-burn," in *Financial Crypto'20*. Springer, 2020, pp. 523–540.

[12] T. Roughgarden, "Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559," [arXiv], 2020.

[13] B. Monnot, Q. Hum, C. S. M. Koh, S. Leonardos, and G. Piliouras, "Ethereum's Transaction Fee Market Reform of EIP 1559," *WINE'20 Workshop on Blockchain*, 2020.

[14] V. Buterin, "Make EIP 1559 more like an AMM curve," Ethereum Research [Online], 2021.

[15] L. Kugler, "Non-fungible tokens and the future of art," *Communications of the ACM*, vol. 64, no. 9, pp. 19–20, 2021.

[16] V. Buterin, "Ethereum white-paper: A next-generation smart contract and decentralized application platform," [Online], 2014.

[17] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," [Online], 2014.

[18] V. Buterin, D. Reijsbergen, S. Leonardos, and G. Piliouras, "Incentives in Ethereum's Hybrid Casper Protocol," in *IEEE-ICBC'19*. USA: IEEE, May 2019, pp. 236–244.

[19] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. USA: Cambridge University Press, 2007.

[20] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of Bitcoin without the block reward," in *CCS'16*, 2016, pp. 154–167.

[21] Y. R. Yang and S. S. Lam, "General aimd congestion control," in *Proceedings 2000 International Conference on Network Protocols*. IEEE, 2000, pp. 187–198.

[22] G. Huberman, J. D. Leshno, and C. Moallemi, "Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System," *The Review of Economic Studies*, vol. online, pp. 1–30, 03 2021, rdab014.

[23] M. Neuder, D. J. Moroz, R. Rao, and D. C. Parkes, "Low-cost attacks on Ethereum 2.0 by sub-1/3 stakeholders," 2021.

[24] C. Schwarz-Schilling, J. Neu, B. Monnot, A. Asgaonkar, E. N. Tas, and D. Tse, "Three Attacks on Proof-of-Stake Ethereum," 2021.