Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559[*]

Tim Roughgarden[†]

December 1, 2020

Abstract

EIP-1559 is a proposal to make several tightly coupled additions to Ethereum's transaction fee mechanism, including variable-size blocks and a burned base fee that rises and falls with demand. This report assesses the game-theoretic strengths and weaknesses of the proposal and explores some alternative designs.

Contents

[*] This work was funded by the Decentralization Foundation (`https://d24n.org/`). This report has benefited from comments by and discussions with a number of people: Maryam Bahrani, Abdelhamid Bakhta, Tim Beiko, Vitalik Buterin, Matheus Ferreira, Danno Ferrin, James Fickel, Hasu, Georgios Konstantopoulos, Andrew Lewis-Pye, Barnabé Monnot, Daniel Moroz, Mitchell Stern, Alex Tabarrok, and Peter Zeitz. Thanks to James also for introducing me to the problem.

[†] Author's permanent position: Professor of Computer Science, Columbia University, 500 West 120th Street, New York, NY 10027. Email: `tim.roughgarden@gmail.com`. Disclosure: I have no financial interests in Ethereum, either long or short.

1 TL;DR

1.1 A Brief Description of EIP-1559

In the Ethereum protocol, the transaction fee mechanism is the component that determines, for every transaction added to the Ethereum blockchain, the price paid by its creator. Since its inception, Ethereum's transaction fee mechanism has been a *first-price auction*: Each transaction comes equipped with a bid, corresponding to the gas limit times the gas price, which is transferred from its creator to the miner of the block that includes it.

EIP-1559 proposes a major change to Ethereum's transaction fee mechanism. Central to the design is a *base fee*, which plays the role of a reserve price and is meant to match supply and demand. Every transaction included in a block must pay that block's base fee (per unit of gas), and this payment is burnt rather than transferred to the block's miner. Blocks are allowed to grow as large as double a target block size; for example, with a target of 12.5M gas, the maximum block size would be 25M gas. The base fee is adjusted after every block, with larger-than-target blocks increasing it and smaller-than-target blocks decreasing it. Users seeking special treatment, such as immediate inclusion in a period of rapidly increasing demand or a specific position within a block, can supplement the base fee with a transaction tip that is transferred directly to the miner of the block that it includes it.

1.2 Ten Key Takeaways

The following list serves as an executive summary for busy readers as well as a road map for those wanting to dig deeper.

1. No transaction fee mechanism, EIP-1559 or otherwise, is likely to substantially decrease average transaction fees; persistently high transaction fees is a scalability problem, not a mechanism design problem. (See Section 3.2.1 for details.)

2. EIP-1559 should decrease the variance in transaction fees and the delays experienced by some users through the flexibility of variable-size blocks. (Section 3.2.2)

3. EIP-1559 should improve the user experience through easy fee estimation, in the form of an "obvious optimal bid," outside of periods of rapidly increasing demand. (Section 6.3)

4. The short-term incentives for miners to carry out the protocol as intended are as strong under EIP-1559 as with first-price auctions. (Sections 6.2 and 6.4)

5. The game-theoretic impediments to double-spend attacks, censorship attacks, denial-of-service attacks, and long-term revenue-maximizing strategies such as base fee manipulation appear as strong under EIP-1559 as with first-price auctions. (Section 7.5)

6. EIP-1559 should at least modestly decrease the rate of ETH inflation through the burning of transaction fees. (Section 9.1)

7. The seemingly orthogonal goals of easy fee estimation and fee burning are inextricably linked through the threat of off-chain agreements. (Sections 8.1–8.2)

8. Alternative designs include paying base fee revenues forward to miners of future blocks rather than burning them; and replacing variable user-specified tips by a fixed hard-coded tip. (Sections 8.3 and 8.5)

9. EIP-1559's base fee update rule is somewhat arbitrary and should be adjusted over time. (Section 8.6)

10. Variable-size blocks enable a new (but expensive) attack vector: overwhelm the network with a sequence of maximum-size blocks. (Sections 8.6.5–8.6.6)

1.3 Organization of Report

Section 2 reviews Ethereum's current transaction fee mechanism and provides a detailed description of the changes proposed in EIP-1559. Section 3 considers the market for computation on the Ethereum blockchain and the basic forces of supply and demand at work. Section 4 formalizes the concepts of a "good user experience" and "easy fee estimation" via posted-price mechanisms. Section 5 defines several desirable game-theoretic guarantees at the time scale of a single block, and Section 6 delineates the extent to which the transaction fee mechanism proposed in EIP-1559 satisfies them. Section 7 investigates the possibility of collusion by miners over long time scales. Section 8 spells out the fatal flaws with some natural alternative designs and identifies worthy directions for further design experimentation. Section 9 covers additional benefits of the mechanism proposed in EIP-1559, along with a short discussion of EIP-2593 (the "escalator"). Section 10 concludes.

Sections 2–4, 7, and 9–10 are relatively non-technical and meant for a general audience. Sections 5–6 and 8 are more mathematically intense and aimed at readers who have at least a passing familiarity with mechanism design theory (see e.g. [54] for the relevant background).[1]

2 Transaction Fee Mechanisms in Ethereum: Present and Future

This section reviews the economically salient properties of Ethereum transactions (Section 2.1), the status quo of a first-price transaction fee mechanism (Section 2.2), the nuts and bolts of the new transaction fee mechanism proposed in EIP-1559 (Section 2.3), and the intuition behind the proposal (Section 2.4).

2.1 Transactions in Ethereum

The Ethereum blockchain, through its Ethereum virtual machine (EVM), maintains state (such as account balances) and carries out instructions that change this state (such as transfers of the native currency, called ether (ETH)). A *transaction* specifies a sequence of instructions to be executed by the EVM. The creator of a transaction is responsible for specifying, among other fields, a *gas limit* and a *gas price* for the transaction. The gas limit is a measure of the cost (in computation, storage, and so on) imposed on the Ethereum blockchain by the transaction. The gas price specifies how much the transaction creator is willing to pay (in ETH) per unit of gas. For example, the most basic type of transaction (a simple transfer) requires 21,000 units of gas; more complex transactions require more gas. Typical gas prices reflect the current demand for EVM computation and have varied over time by orders of magnitude; readers wishing to keep a concrete gas price in mind could use, for example, 100 gwei (where one gwei is 10^{-9} ETH). The total amount that the creator of a transaction offers to pay for its execution is then the gas limit times the gas price:

$$\text{amount paid} := \text{gas limit} \times \text{gas price}. \tag{1}$$

For example, for a 21,000-gas transaction with a gas price of 100 gwei, the corresponding payment would be 2.1×10^{-3} ETH (or 1.26 USD at an exchange rate of 600 USD/ETH).

A *block* is an ordered sequence of transactions and associated metadata (such as a reference to the predecessor block). There is a cap on the total gas consumed by the transactions of a block,

[1] Other economic analyses of EIP-1559 include [8, 30, 50, 51, 59].

which we call the *maximum block size*. The maximum block size has increased over time and is currently 12.5M gas, enough for roughly 600 of the simplest transactions. Blocks are created and added to the blockchain by *miners*. Each miner maintains a *mempool* of outstanding transactions and collects a subset of them into a block. To add a block to the blockchain, a miner provides a proof-of-work in the form of a solution to a computationally difficult cryptopuzzle; the puzzle difficulty is adjusted over time to maintain a target rate of block creation (roughly one block per 13 seconds). Importantly, the miner of a block has dictatorial control over which outstanding transactions are included and their ordering within the block. Transactions are considered confirmed once they are included in a block that is added to the blockchain. The current state of the EVM is then the result of executing all the confirmed transactions, in the order they appear in the blockchain.[2]

The *transaction fee mechanism* is the part of the protocol that determines the amount that a creator of a confirmed transaction pays, and to whom that payment is directed.

2.2 First-Price Auctions

Ethereum's transaction fee mechanism is and always has been a *first-price auction* [15].[3]

First-Price Auctions

1. *Who pays what?* The creator of a confirmed transaction pays the specified gas limit times the specified gas price (as in (1)).

2. *Who gets the payment?* The entire payment is transferred to the miner of the block that includes the transaction.[4]

A user submitting a transaction is sure to pay either the amount in (1) (if the transaction is confirmed) or 0 (otherwise). A miner who mines a block is sure to receive as revenue the amount in (1) from each of the transactions it chooses to include. Accordingly, many miners pack blocks up to the maximum block size, greedily prioritizing the outstanding transactions with the highest gas prices.[5,6]

2.3 EIP-1559: The Nuts and Bolts

2.3.1 Burning a History-Dependent Base Fee

EIP-1559, following Buterin [16, 17, 18], proposes a mechanism that makes several tightly coupled changes to the status quo.

[2] Technically, a longest-chain rule is used to resolve forks (that is, two or more blocks claiming a common predecessor). The confirmed transactions are then defined as those in the blocks that are well ensconced in the longest chain (that is, already extended by sufficiently many subsequent blocks).

[3] First-price auctions are also used in Bitcoin [47].

[4] We will ignore details concerning transactions that run out of gas or complete with unused gas.

[5] Technically, because different transactions have different gas limits, selecting the revenue-maximizing set of transactions is a knapsack problem (see e.g. [55]). The minor distinction between optimal and greedy knapsack solutions is not important for this report.

[6] We use the word "greedy" without judgment—"greedy algorithm" is a standard term for a heuristic that is based on a sequence of myopic decisions.

EIP-1559: Key Ideas (1–3 of 8)

1. Each block has a protocol-computed reserve price (per unit of gas) called the *base fee*. Paying the base fee is a prerequisite for inclusion in a block.[7]
2. The base fee is a function of the preceding blocks only, and does not depend on the transactions included in the current block.
3. All revenues from the base fee are burned—that is, permanently removed from the circulating supply of ETH.

Removing ETH from the supply increases the value of every ether still in circulation. Fee-burning can therefore be viewed as a lump-sum refund to ETH holders (à la stock buybacks).

The second point is underspecified; how, exactly, is the base fee derived from the preceding blocks? Intuitively, increases and decreases in demand should put upward and downward pressure on the base fee, respectively.[8] But the blockchain records only the confirmed transactions, not the transactions that were priced out. If miners publish a sequence of full (12.5M gas) blocks, how can the protocol distinguish whether the current base fee is too low or exactly right?

2.3.2 Variable-Size Blocks

The next key idea is to relax the constraint that every block has size at most 12.5M gas and instead require only that the *average* block size is at most 12.5M gas.[9] The mechanism in EIP-1559 then uses past block sizes as an on-chain measure of demand, with big blocks (more than 12.5M gas) and small blocks (less than 12.5M gas) signaling increasing and decreasing demand, respectively.[10] Some finite maximum block size is still needed to control network congestion; the current EIP-1559 spec [20] proposes using twice the average block size.

EIP-1559: Key Ideas (continued)

4. Double the maximum block size (e.g., from 12.5M gas to 25M gas), with the old maximum (e.g., 12.5M gas) serving as the *target* block size.
5. Adjust the base fee upward or downward whenever the size of the latest block is bigger or smaller than the target block size, respectively.

The specific adjustment rule proposed in the EIP-1559 spec [20] computes the base fee r_{cur} for the current block from the base fee r_{pred} and size s_{pred} of the predecessor block using the following

[7] Technically, a miner can also include a transaction unwilling to pay the full base fee, but it must then dip into its block reward to make up the difference. We ignore this detail in this report.

[8] In the economics literature, such demand-dependent price adjustment is called "tâtonnement" (French for "groping").

[9] More generally, EIP-1559 is parameterized by a target block size, which is adjusted by miners over time (like the maximum block size is now). For concreteness, throughout this report we assume a target block size of 12.5M gas, the current maximum block size.

[10] The flexibility provided by variable block sizes can also reduce the variance in equilibrium transaction fees and the delays experienced by some users; see Section 3.2.

formula, where s_{target} denotes the target block size:[11]

$$r_{cur} := r_{pred} \cdot \left(1 + \frac{1}{8} \cdot \frac{s_{pred} - s_{target}}{s_{target}}\right). \qquad (2)$$

For example, the base fee increases by 12.5% after a maximum-size block (i.e., double the target size) and decreases by 12.5% after an empty block. A maximum-size block followed by an empty block (or vice versa) leaves the base fee at $\frac{9}{8} \cdot \frac{7}{8} = \frac{63}{64} \approx 98.4\%$ of its prior value.[12]

If the base fee is burned rather than given to miners, why should miners bother to include any transactions in their blocks at all? Also, what happens when there are lots of transactions (more than 25M gas worth) willing to pay the current base fee?

2.3.3 Tips

The transaction fee mechanism proposed in EIP-1559 addresses the preceding two questions by allowing the creator of a transaction to specify a *tip*, to be paid above and beyond the base fee, which is transferred to the miner of the block that includes the transaction (as in a first-price auction). Small tips should be sufficient to incentivize a miner to include a transaction during a period of stable demand, when there is room in the current block for all the outstanding transactions that are willing to pay the base fee. Large tips can be used to encourage special treatment of a transaction, such as a specific positioning within a block, or the immediate inclusion in a block in the midst of a sudden demand spike.

EIP-1559: Key Ideas (continued)

6. Rather than a single gas price, a transaction now includes a *tip* and a *fee cap*. A transaction will be included in a block only if its fee cap is at least the block's base fee.
7. *Who pays what?* If a transaction with tip δ, fee cap c, and gas limit g is included in a block with base fee r, the transaction creator pays $g \cdot \min\{r + \delta, c\}$ ETH.
8. *Who gets the payment?* Revenue from the base fee (that is, $g \cdot r$) is burned and the remainder ($g \cdot \min\{\delta, c - r\}$) is transferred to the miner of the block.

For example, consider a block with base fee 100 (in gwei per unit of gas). If the block's miner includes a transaction with tip 4 and fee cap 200, the creator of that transaction will pay 104 gwei per unit of gas (100 of which is burned, 4 of which goes to the miner). An included transaction with tip 10 and fee cap 105 would pay 105 gwei per unit of gas (100 of which is burned, 5 of which goes to the miner).

A user submitting a transaction with tip δ and fee cap c is sure to pay at most c gwei per unit of gas, and will pay less whenever the current base fee is small (i.e., less than $c - \delta$). A miner who mines a block is sure to receive all the revenue from the tips of the transactions it chooses to include. Accordingly, one might expect a typical miner to include all the transactions with fee cap greater than the base fee. If the total gas consumed by such transactions exceeds the maximum

[11] For simplicity, we ignore numerical details such as rounding the base fee to an integer.

[12] See also Table 1 in Section 3.2.2 for a more complex example of this update rule in action, Monnot [43] for detailed simulations, and Filecoin [4] for a recent deployment.

block size of 25M gas, one might expect the miner to pack its block full, greedily prioritizing the outstanding transactions with the highest tips.

2.4 An Informal Argument for EIP-1559

The number of new ideas in EIP-1559 can be overwhelming. Why so many changes at once? Does one of the changes necessitate the rest? We next outline one narrative of why EIP-1559 might have to look more or less the way that it does, taking as given the goal of making fee estimation far easier for users than in the status quo. The remainder of this report will interrogate this narrative mathematically and explore some alternative designs.

Why EIP-1559 Looks the Way That It Does (Informal Argument)

1. First-price auctions are challenging for users to reason about because a user's optimal gas price depends on the gas prices offered by other users at the same time.

2. Other common auction designs in which the prices charged depend on the set of included transactions, such as second-price (a.k.a. Vickrey) auctions, can be easily manipulated by miners through fake transactions.

3. Simple fee estimation, in which users are not forced to reason about other users' behavior, therefore seems to require a base fee—a price that is set independently of the transactions included in the current block.

4. The ideal base fee would result in blocks filled with the highest-value transactions. Demand changes over time, so the base fee must respond in kind.

5. The base fee revenues of a block must be burned or otherwise withheld from the block's miner, as otherwise the miner could collude with users off-chain to costlessly simulate a first-price auction.

6. Because demand is not recorded on-chain, an on-chain proxy such as variable block sizes must be used to adjust the base fee.

7. Tips are required to disincentivize miners from publishing empty blocks.

8. Tips should be specified by users rather than hard-coded into the protocol so that high-value transactions can be identified and accommodated during a sudden demand spike.

9. Burning any portion of the tips would drive the tip market off-chain, and thus tips may as well be transferred entirely to a block's miner.

3 The Market for Ethereum Transactions

This section steps away from the discussion of specific mechanisms and focuses instead the basic forces of supply and demand at work in the Ethereum blockchain. Section 3.1 defines a "market-clearing outcome" and posits it as the ideal outcome of a transaction fee mechanism. Section 3.2

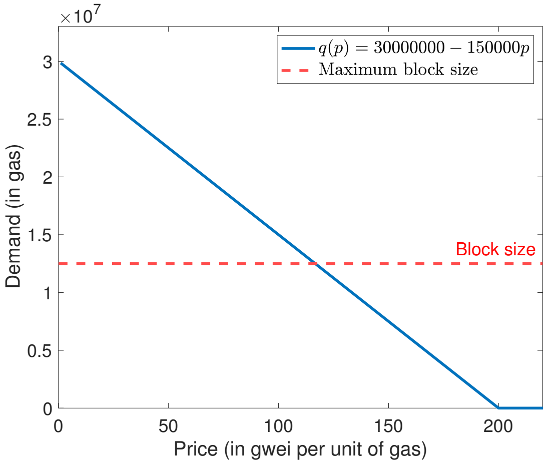


Figure 1: An example linear demand curve, with $b = 30\text{M}$ and $a = 150\text{K}$. For example, there is a demand of 30M gas at a gas price of 0; zero demand at a gas price of 200 gwei; and a demand of 12.5M gas at a gas price of $116\frac{2}{3}$ gwei.

emphasizes that no mechanism can guarantee low transaction fees during periods in which the demand for EVM computation significantly outstrips its supply, and clarifies EIP-1559's likely effect on high transaction fees.

3.1 Market-Clearing Prices and Outcomes

The 12.5M gas available in an Ethereum block is a scarce resource, and in a perfect world it should be allocated to the transactions that derive the most value from it. We can make this idea precise using a *demand curve*, which is a decreasing function that specifies the total amount of gas demanded by users at a given gas price.[13] For example, a *linear* demand curve has the form $D(p) = \max\{0, b - ap\}$, where p denotes the gas price and $a, b \geq 0$ are nonnegative constants (Figure 1).

The *market-clearing price* is then the price at which the total amount of gas demanded equals the available supply (i.e., 12.5M gas). For example, in Figure 1, the market clearing price is $116\frac{2}{3}$ gwei. If the demand at price 0 is less than the supply, we define the market-clearing price as 0.

The market-clearing price is the ideal gas price for a block. For suppose such a price p^* fell magically from the sky and became common knowledge to all users, with the understanding that all confirmed transactions in the current block will pay p^* per unit of gas. In the resulting outcome—the *market-clearing outcome*—users with maximum willingness to pay at least p^* per unit of gas will opt to have their transactions included, while those with a lower willingness to pay opt out. The end result? The supply of 12.5M gas will be fully utilized (because p^* is a market-clearing price), and moreover will be allocated precisely to the highest-value transactions (those willing to

[13] For simplicity of analysis, throughout this report we assume that demand is exogenous and independent of the choice of or actions by a transaction fee mechanism. Houy [34] and Rizun [52] use a similar formalism to reason about blockchain transaction fee markets. Richer models of demand, with pending transactions excluded from one block persisting to the next, are studied by Monnot [43, 45] in the context of EIP-1559 simulations and by Easley et al. [25] and Huberman et al. [35] to carry out an economic analysis of Bitcoin's transaction fee mechanism.

pay a gas price of at least p^*).[14] Put differently, the market-clearing outcome maximizes the value of the current block, subject to the supply constraint of 12.5M gas. For this reason, we adopt the market-clearing outcome as the most desirable one for a transaction fee mechanism.

Ideal Outcome of a Transaction Fee Mechanism

Every block is fully utilized by the highest-value transactions, with all transactions paying a gas price equal to the market-clearing price.

Both the status quo and EIP-1559 transaction fee mechanisms can be viewed as striving for this ideal, market-clearing outcome. In first-price auctions, users are expected to estimate what the current market-clearing price might be and bid accordingly. In the EIP-1559 mechanism, the protocol continually adjusts the base fee in search of the market-clearing price.

Remark 3.1 (Revenue as a Necessary Evil) The purpose of the market-clearing price is to differentiate high-value and low-value transactions, so that the scarce resource that is an Ethereum block can be allocated in the most valuable way. Revenue is generated in the market-clearing outcome (provided the supply constraint is binding), but only as a side effect in the service of economic efficiency. The revenue-maximizing price is generally higher than the market-clearing price, and it plays an important role in the discussion in Section 7 of possible attacks by colluding miners.

Remark 3.2 (Non-Zero Marginal Costs) The preceding definition of a market-clearing outcome assumes that the marginal cost to a miner of including an additional transaction in its block is 0 (or $+\infty$, if including the transaction would violate the cap of 12.5M gas). In reality, every transaction imposes a small marginal cost on the miner; for example, one factor is that the probability that a block is orphaned from the main chain (i.e., the "uncle rate") increases with the block size [24].

If the overall marginal cost to a miner is μ gwei per unit of gas, then μ plays the role of 0 in the more general definitions of market-clearing prices and outcomes.[15] That is, if the demand at price μ is at most the supply of 12.5M gas, the market-clearing price is μ; in the corresponding outcome, all transactions willing to pay a gas price of at least μ are included in the block.

3.2 Will EIP-1559 Lower Transaction Fees?

The Ethereum community is justifiably concerned about overly high transaction fees crowding out all but the most lucrative uses of the Ethereum blockchain (e.g., DeFi arbitrage opportunities). No transaction fee mechanism can be a panacea to this problem. This section clarifies what effects on transaction fees should and should not be expected from the adoption of the transaction fee mechanism proposed in EIP-1559.

3.2.1 The Problem of High Market-Clearing Prices

First, whatever the mechanism, real transaction fees cannot be expected to drop significantly below the market-clearing price during a period of relatively stable demand. With fees below that

[14] Or if the supply constraint is not binding (and hence $p^* = 0$), all transactions are included.

[15] Alternatively, μ is the minimum compensation per unit of gas that a miner is willing to accept for including a transaction.

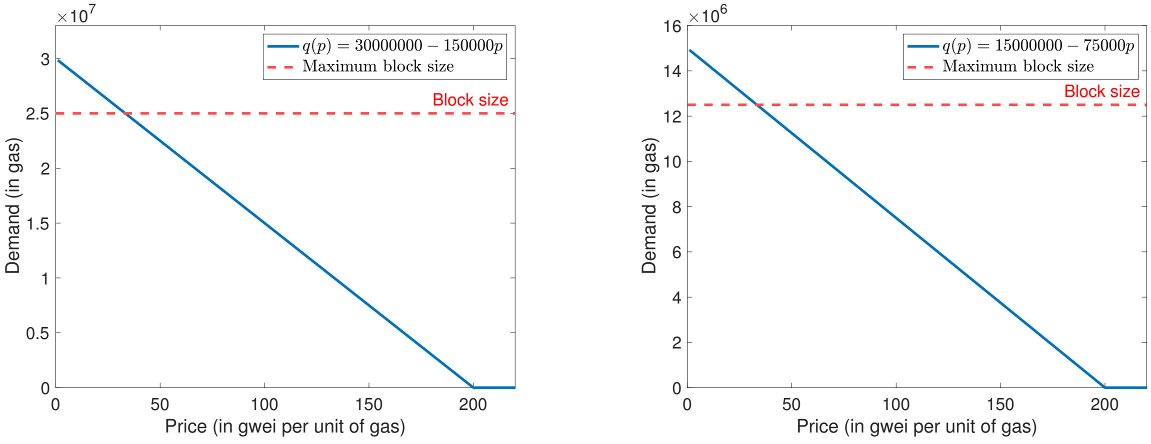
price, demand for gas would exceed supply, resulting in some lower-value transactions replacing higher-value transactions. For example, with the demand curve in Figure 1, if typical fees dropped to 100 gwei per unit of gas, the demand would be 15M gas. The 2.5M gas worth of excluded transactions will inevitably include some for which the creator's willingness to pay is at least the market-clearing price of $116\frac{2}{3}$ gwei. Such users should be expected to push up transaction fees and guarantee inclusion of their transactions, either on-chain through the transaction fee mechanism (e.g., by increasing a transaction's gas price in a first-price auction), or off-chain through a side agreement with a miner.

But what if the market-clearing price is already unacceptably high? The only ways to decrease the market-clearing price are to increase supply or decrease demand (Figure 2)—actions that are generally outside the purview of mechanism design.

Scalability vs. Mechanism Design

Lowering the market-clearing price by increasing supply or decreasing demand is fundamentally a scalability problem, not a mechanism design problem.

For example, typical layer-1 scaling solutions like sharding, in which different parts of the blockchain operate in parallel, increase transaction throughput and therefore decrease the market-clearing price. Typical layer-2 scaling solutions like payment channels and rollups, which effectively move some transactions off-chain, decrease demand for EVM computation and likewise decrease the market-clearing price. Looking toward the near future, good scaling solutions will be crucial for keeping transaction fees in check and more generally for encouraging the growth of the Ethereum network.



(a) Increasing the supply

(b) Decreasing the demand

Figure 2: In the example in Figure 1, doubling the supply (shown in (a)) or halving the demand (shown in (b)) cuts the market-clearing price from $116\frac{2}{3}$ gwei to $33\frac{1}{3}$ gwei.

3.2.2 Two Potential Benefits of EIP-1559

The transaction fee mechanism proposed in EIP-1559 has the potential to partially mitigate high transaction fees in two different ways. First, in a period of relatively stable demand, users can

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8
Demand	Low	High	High	High	High	High	High	Low
M-C Price (12.5M)	33.33	116.67	116.67	116.67	116.67	116.67	116.67	33.33
EIP-1559 Base Fee	33.33	33.33	37.5	41.95	46.65	51.55	56.59	61.69
EIP-1559 Block Size	12.5M	25M	24.38M	23.71M	23M	22.27M	21.51M	10.37M

Table 1: An example of the EIP-1559 base fee adjustment rule in action. "Low" demand means the demand curve $D(p) = 15000000 - 75000p$ shown in Figure 2(b); "high" means the demand curve $D(p) = 30000000 - 150000p$ shown in Figure 1. (Here "demand" means the total gas consumed by all pending transactions that have a fee cap of p or more.) The second row shows the market-clearing price for each demand curve when the supply is fixed at 12.5M gas. The third and fourth rows show the joint evolution of the base fee and block size under the EIP-1559 mechanism, assuming that the base fee matches the market-clearing price in period 1 and that all users submit negligible tips.

adopt the base fee as a good known-in-advance proxy for the market-clearing price; this should lead to less guesswork and consequent overpayment than in today's first-price auctions. See also the discussion in Section 4.1.

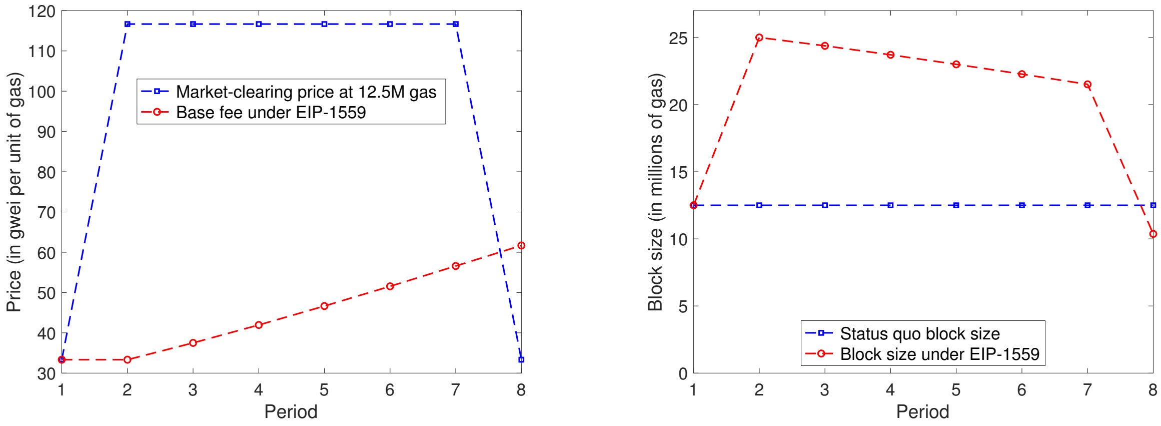
Second, in a period of volatile demand, the mechanism proposed in EIP-1559 can reduce the *variance* in transaction fees experienced by users by exploiting variable block sizes—in effect, borrowing capacity from the near future to use in a time of need. This flexibility in block sizes can reduce the maximum transaction fee paid during the period (as well as the delay experienced by some users).

Example 3.3 (Trajectory of EIP-1559) Consider the trajectory of the EIP-1559 mechanism that is detailed in Table 1 and depicted in Figure 3. For this example, we assume that tips are negligible and that a transaction is included in a block if and only if its fee cap is at least the current base fee. Period 1 represents the end of a long era of stable demand, during which the base fee converged to the market-clearing price for the target block size (12.5M gas). Demand doubles for the next six periods. With a fixed supply of 12.5M gas, the market-clearing price jumps suddenly from $33\frac{1}{3}$ to $116\frac{2}{3}$ after period 1, and back to $33\frac{1}{3}$ after period 7. In the EIP-1559 mechanism, the base fee—the mechanism's guess at the current market-clearing price for the target block size—increases slowly but surely, with larger-than-target blocks absorbing the excess demand along the way. Once demand returns to its original level, blocks will have size smaller than the target as the mechanism's base fee slowly but surely decreases to the new market-clearing price. In this example, the maximum base fee of 61.69 (in period 8) is only about 53% of the maximum market-clearing price with a fixed block size of 12.5M gas ($116\frac{2}{3}$, in periods 2–7).

4 The Purpose of EIP-1559: Easy Fee Estimation

4.1 The Problem of Fee Estimation

With or without EIP-1559, transaction fees will be high whenever the demand for EVM computation far exceeds its supply (Section 3.2). So what's the point of the proposal? To make transaction fees *more predictable* and thereby make the fee estimation problem—the problem of choosing the optimal gas price for a transaction—as straightforward as possible.



(a) Price comparison

(b) Block size comparison

Figure 3: For the example detailed in Table 1, a comparison of the price and block size under the status quo and under EIP-1559. In the subsequent periods, the base fee and block size under EIP-1559 gradually return to $33\frac{1}{3}$ gwei and 12.5M gas, respectively.

Ethereum users appear to overpay regularly for EVM computation, offering gas prices that are significantly larger than the market-clearing price [3]. Part of the problem may be attributable to poor fee estimation algorithms in wallets, which could conceivably improve over time (see e.g. [39, 48], in the similar context of Bitcoin). But part of the problem is fundamental to first-price auctions, and addressing it necessitates a major change in the transaction fee mechanism.[16]

EIP-1559: Improving the User Experience with Easy Fee Estimation

This report assumes that the primary purpose of EIP-1559 is to improve the "user experience (UX)" of Ethereum users, and to do so specifically by making the fee estimation problem as easy as possible.

EIP-1559 also offers a number of other benefits (see Section 9.1), which are treated in this report as happy accidents—byproducts of the proposed UX improvements.[17]

4.2 Auctions vs. Posted-Price Mechanisms

To what extent does EIP-1559 achieve its goal of a "better UX"? "User experience" is a vague term, and it must be defined mathematically before this question can be answered.

Definition 5.21 presents our formalization of "good UX," and the intuition for it is simple: Shopping on Amazon is a lot easier than buying a house in a competitive real estate market. On Amazon, there's no need to be strategic or second-guess yourself; you're either willing to pay the listed price for the listed product, or you're not. The outcome is economically efficient in that every

[16] Bidding in a first-price auction has long been known to be a hard problem; see e.g. [22].

[17] This viewpoint appears consistent with the original motivation for EIP-1559. Buterin [19] writes: "Our goal is to discourage the development of complex miner strategies and complex transaction sender strategies in general, including both complex client-side calculations and economic modeling as well as various forms of collusion."

user who buys a product has a higher willingness to pay for it than every user who doesn't buy the product.

When pursuing a house and competing with other potential buyers, you must think carefully about what price to offer to the seller. And no matter how smart you are, you might regret your offer in hindsight—either because you underbid and were outbid at a price you would have been willing to pay, or because you overbid and paid more than you needed to. The house need not be sold to the potential buyer willing to pay the most (if that buyer shades their bid too aggressively), which is a loss in economic efficiency.

Bidding in Ethereum's first-price auctions is like buying a house. Estimating the optimal gas price for a transaction requires making educated guesses about the gas prices chosen for the competing transactions. From a user's perspective, any bid may end up looking too high or too low in hindsight. From a societal perspective, lower-value transactions that bid aggressively may displace higher-value transactions that do not.

Could we redesign Ethereum's transaction fee mechanism so that setting a transaction's gas price is more like shopping on Amazon? Ideal would be a *posted-price mechanism*, meaning a mechanism that offers each user a take-it-or-leave-it gas price for inclusion in the next block. We'll see in Section 6.3 that the transaction fee mechanism proposed in EIP-1559 acts like a posted-price mechanism except when there is a large and sudden increase in demand (Theorem 6.8).

5 Incentive-Compatible Transaction Fee Mechanisms

This section formalizes three desirable game-theoretic guarantees for a transaction fee mechanism. First, miners should be incentivized to carry out the mechanism as intended, and strongly disincentivized from including fake transactions (Section 5.3). Second, the optimal gas price to specify should be obvious to the creator of a transaction (Section 5.4). Finally, there should be no way for miners and users to collude and strictly increase their utility by moving payments off-chain (Section 5.5). Sections 5.1 and 5.2 set up the notation and language necessary to formally state these three definitions.

This and the next section focus on incentives for miners and users at the time scale of a single block, and on two important types of attacks that can be carried out at this time scale (the insertion of fake transactions, and off-chain agreements between miners and users). Section 7 treats incentive issues and attacks that manifest over longer time scales.

5.1 The Basic Model

On the supply side, let G denote the maximum size of a block in gas (e.g., 12.5M gas in the status quo or 25M gas under EIP-1559), and $\mu \geq 0$ the marginal cost of gas to a miner (as in Remark 3.2).[18] For simplicity, we assume that μ is the same for all miners and common knowledge among users.[19] On the demand side, let M denote the set of transactions in the mempool at the time of the current block's creation.

We associate three parameters with each transaction $t \in M$:

[18] Equivalently, μ is the minimum gas price that a profit-maximizing miner is willing to accept in exchange for transaction inclusion when the maximum block size is not a binding constraint. The formal definition of a "profit-maximizing miner" is given in Definition 5.13.

[19] Calculations by Buterin [1] suggest that μ is, at this time of writing, on the order of 0.4–3.3 gwei. In a proof-of-stake blockchain such as ETH 2.0, the parameter μ is likely to be even smaller.

- a *gas limit* g_t in gas;
- a *value* v_t in gwei per unit of gas;
- a *bid* b_t in gwei per unit of gas.

The gas limit is the amount of gas required to carry out the transaction. The value is the maximum gas price the transaction's creator would be willing to pay for its execution in the current block.[20] The bid corresponds to the gas price that the creator actually offers to pay, which in general can be less (or more) than the value. With a first-price auction, the bid corresponds to the gas price specified for a transaction. In the transaction fee mechanism proposed in EIP-1559, the bid corresponds to the minimum of the fee cap and the sum of the base fee and the tip ($\min\{r + \delta, c\}$ in the notation of Section 2.3). We view the gas limit and value as immutable properties of a transaction; the bid, by contrast, is under control of the transaction's creator. The gas limit and bid of a confirmed transaction are recorded on-chain; the value of a transaction is known solely to its creator.

5.2 Allocation, Payment, and Burning Rules

A transaction fee mechanism decides which transactions should be included in the current block, how much the creators of those transaction have to pay, and to whom their payment is directed. These decisions are formalized by three functions: an *allocation rule*, a *payment rule*, and a *burning rule*.

5.2.1 Allocation Rules

We use $B_1, B_2, \ldots, B_{k-1}$ to denote the sequence of blocks in the current longest chain (with B_1 the genesis block and B_{k-1} the most recent block) and M the pending transactions in the mempool. Generally, bold type (like $\mathbf{x}$) will indicate a vector and regular type (like x_t) one of its components.

Definition 5.1 (Allocation Rule) An *allocation rule* is a vector-valued function $\mathbf{x}$ from the on-chain history $B_1, B_2, \ldots, B_{k-1}$ and mempool M to a 0-1 value $x_t(B_1, B_2, \ldots, B_{k-1}, M)$ for each pending transaction $t \in M$.

A value of 1 for $x_t(B_1, B_2, \ldots, B_{k-1}, M)$ indicates transaction t's inclusion in the current block B_k; a value of 0 indicates its exclusion. We sometimes write $B_k = \mathbf{x}(B_1, B_2, \ldots, B_{k-1}, M)$, with the understanding that B_k is the set of transactions t for which $x_t(B_1, B_2, \ldots, B_{k-1}, M) = 1$.

We consider only feasible allocation rules, meaning allocation rules that respect the maximum block size G.

Definition 5.2 (Feasible Allocation Rule) An allocation rule $\mathbf{x}$ is *feasible* if, for every possible history $B_1, B_2, \ldots, B_{k-1}$ and mempool M,

$$\sum_{t \in M} g_t \cdot x_t(B_1, B_2, \ldots, B_{k-1}, M) \leq G. \tag{3}$$

[20] We assume that the value is independent of the position in the block, ignoring e.g. front-running bots aiming to secure the first position in a block (see [23, 53]).

We call a set T of transactions *feasible* if they can all be packed in a single block: $\sum_{t \in T} g_t \leq G$.

Remark 5.3 (Miners Control Allocations) While a transaction fee mechanism is generally designed with a specific allocation rule in mind, it is important to remember that a miner ultimately has dictatorial control over the block it creates.

Example 5.4 (First-Price Auction Allocation Rule) The (intended) allocation rule $\mathbf{x}^f$ in a first-price auction is to include a feasible subset of outstanding transactions that maximizes the sum of the gas-weighted bids, less the gas costs. That is, the x_t^f's are assigned 0-1 values to maximize

$$\sum_{t \in M} x_t^f(B_1, B_2, \ldots, B_{k-1}, M) \cdot (b_t - \mu) \cdot g_t, \tag{4}$$

subject to (3).

5.2.2 Payment and Burning Rules

The payment rule specifies the revenue earned by the miner from included transactions.

Definition 5.5 (Payment Rule) A *payment rule* is a function $\mathbf{p}$ from the current on-chain history $B_1, B_2, \ldots, B_{k-1}$ and transactions B_k included in the current block to a nonnegative number $p_t(B_1, B_2, \ldots, B_{k-1}, B_k)$ for each included transaction $t \in B_k$.

The value of $p_t(B_1, B_2, \ldots, B_{k-1}, B_k)$ indicates the payment from the creator of an included transaction $t \in B_k$ to the miner of the block B_k (in ETH, per unit of gas).

For example, in a first-price auction, a winner always pays its bid (per unit of gas), no matter what the blockchain history and other included transactions.

Example 5.6 (First-Price Auction Payment Rule) In a first-price auction,

$$p_t^f(B_1, B_2, \ldots, B_{k-1}, B_k) = b_t$$

for all $B_1, B_2, \ldots, B_k$ and $t \in B_k$.

Finally, the burning rule specifies the amount of ETH burned—or equivalently, refunded to ETH holders—for each of the included transactions.

Definition 5.7 (Burning Rule) A *burning rule* is a function $\mathbf{q}$ from the current on-chain history $B_1, B_2, \ldots, B_{k-1}$ and transactions B_k included in the current block to a nonnegative number $q_t(B_1, B_2, \ldots, B_{k-1}, B_k)$ for each included transaction $t \in B_k$.

The value of $q_t(B_1, B_2, \ldots, B_{k-1}, B_k)$ indicates the amount of ETH burned (per unit of gas) by the creator of an included transaction $t \in B_k$.

Example 5.8 (First-Price Auction Burning Rule) Status quo first-price auctions burn no fees, so

$$q_t^f(B_1, B_2, \ldots, B_{k-1}, B_k) = 0$$

for all $B_1, B_2, \ldots, B_k$ and $t \in B_k$.

Remark 5.9 (The Protocol Controls Payments and Burns) A miner does not control the payment or burning rule, except inasmuch as it controls the allocation, meaning the transactions included in B_k. Given a choice of allocation, the on-chain payments and fee burns are completely specified by the protocol. (Miners might seek out off-chain payments, however; see Section 5.5.)

Remark 5.10 (Mempool-Dependence) The allocation rule $\mathbf{x}$ depends on the mempool M because a miner can base its allocation decision on the entire set of outstanding transactions. Payment and burning rules must be computable from the on-chain information $B_1, B_2, \dots, B_k$, and in particular cannot depend on outstanding transactions of M excluded from the current block B_k.

5.2.3 Transaction Fee Mechanisms

Formally, a transaction fee mechanism is specified by its allocation, payment, and burning rules.

Definition 5.11 (Transaction Fee Mechanism (TFM)) A *transaction fee mechanism (TFM)* is a triple $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ in which $\mathbf{x}$ is a feasible allocation rule, $\mathbf{p}$ is a payment rule, and $\mathbf{q}$ is a burning rule.

For example, a first-price auction is mathematically encoded by the triple $(\mathbf{x}^f, \mathbf{p}^f, \mathbf{q}^f)$ in which $\mathbf{x}^f$ is the revenue-maximizing allocation rule (Example 5.4), $\mathbf{p}^f$ is the pay-as-bid payment rule (Example 5.6), and $\mathbf{q}^f$ is the all-zero burning rule (Example 5.8).

Finally, we consider only individually rational mechanisms, meaning TFMs that cannot force users to pay more than their declared willingness to pay.

Definition 5.12 (Individual Rationality) A TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ is *individually rational* if, for every history $B_1, B_2, \dots, B_k$,

$$\underbrace{p_t(B_1, B_2, \dots, , B_k) + q_t(B_1, B_2, \dots, , B_k)}_{\text{total gas price paid by } t\text{'s creator}} \leq b_t$$

for every transaction $t \in B_k$.

5.3 Incentive Compatibility (Myopic Miners)

This section formalizes what it means for a TFM to be game-theoretically sound from the perspective of miners—intuitively, that a miner is incentivized to implement the intended allocation rule and disincentivized from including fake transactions. As a reminder, our current focus is on incentives at the time scale of a single block, with longer time scales discussed in Section 7.

5.3.1 Myopic Miner Utility Function

In addition to choosing an allocation (Remark 5.3), we assume that miners can costlessly add any number of fake transactions to the mempool (with arbitrary gas limits and bids). We call a miner *myopic* if its *utility*—meaning the quantity that it acts to maximize—equals its net revenue from the current block.[21]

[21] We ignore the block reward (currently 2 ETH), as it is independent of the miner's actions and therefore irrelevant for the single-block game-theoretic analysis in this and the next section. The block reward does, of course, affect the security of the Ethereum blockchain (e.g. [10, 14]).

Definition 5.13 (Myopic Miner Utility Function) For a TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$, on-chain history B_1, $B_2, \ldots, B_{k-1}$, mempool M, fake transactions F, and choice $B_k \subseteq M \cup F$ of included transactions (real and fake), the utility of a *myopic miner* is

$$u(F, B_k) := \underbrace{\sum_{t \in B_k \cap M} p_t(B_1, B_2, \ldots, , B_k) \cdot g_t}_{\text{miner's revenue}} - \underbrace{\sum_{t \in B_k \cap F} q_t(B_1, B_2, \ldots, , B_k) \cdot g_t}_{\text{fee burn for miner's fake transactions}} - \underbrace{\mu \sum_{t \in B_k} g_t}_{\text{gas costs}} . \quad (5)$$

The first term sums over only the real included transactions, as for fake transactions the payment goes from the miner to itself. The second term sums over only the fake transactions, as for real transactions the burn is paid by their creators (not the miner). In (5), we highlight the dependence of the utility function on the two arguments that are under a miner's direct control, the choices of the fake transactions F and included (real and fake) transactions B_k.[22]

5.3.2 Incentive-Compatibility for Myopic Miners

A transaction fee mechanism is generally designed with a specific allocation rule in mind (Remark 5.3), but will miners actually implement it?

Definition 5.14 (Incentive-Compatibility for Myopic Miners (MMIC)) A TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ is *incentive-compatible for myopic miners (MMIC)* if, for every on-chain history $B_1, B_2, \ldots, B_{k-1}$ and mempool M, a myopic miner maximizes its utility (5) by creating no fake transactions (i.e., setting $F = \emptyset$) and following the suggestion of the allocation rule $\mathbf{x}$ (i.e., setting $B_k = \mathbf{x}(B_1, B_2, \ldots, B_{k-1}, M)$).

Example 5.15 (First-Price Auctions Are MMIC) A status quo first-price auction $(\mathbf{x}^f, \mathbf{p}^f, \mathbf{q}^f)$ is MMIC. Because $\mathbf{q}^f$ is the all-zero function (Example 5.8), the second term in (5) is zero. Because payments equal bids (Example 5.6), miner utility equals the exact same quantity (4) maximized by the allocation rule $\mathbf{x}^f$ (Example 5.4). Thus, myopic miner utility is maximized by following the allocation rule and setting $B_k = \mathbf{x}^f(B_1, B_2, \ldots, B_{k-1}, M)$.

Example 5.16 (Vickrey (Second-Price) Auctions Are Not MMIC) *Vickrey* (a.k.a. *second-price*) auctions play as central a role in traditional auction theory as first-price auctions. Their claim to fame is that, assuming the auction is implemented by a trusted third party, truthful bidding (i.e., setting one's bid b_t equal to one's value v_t) is a dominant strategy, meaning it maximizes a bidder's utility no matter what the other bidders do. This property sure sounds like "easy fee estimation," so why not use it as a TFM?

Unfortunately, Vickrey auctions can be manipulated via fake transactions and thus fail to be MMIC. For example, consider a set of transactions that all have the same gas limit and a block that has room for three of them. In this setting, a Vickrey auction would prescribe including the three transactions with the highest bids and charging each of them (per unit of gas) the lowest of these three bids.[23] Now imagine that the top three bids are 10, 8, and 3. If a miner honestly executes a Vickrey auction, its revenue will be $3 \times 3 = 9$. If the miner instead submits a fake transaction with

[22] We can assume that $F \subseteq B_k$, as there's no point to creating and then excluding a fake transaction.

[23] Actually, a Vickrey auction would prescribe charging the highest losing bid rather than the lowest winning bid. The former is off-chain and thus unimplementable in a blockchain context, while the latter is on-chain and typically close enough.

bid 8 and executes a Vickrey auction (with the top two real transactions included along with the fake transaction), its net revenue jumps to $2 \times 8 = 16$.

Remark 5.17 (Credible Mechanisms) The definition of MMIC (Definition 5.14) is closely related to Akbarpour and Li's notion of a *credible mechanism* [9]. Intuitively, a mechanism is credible if the agent tasked with carrying it out has no plausibly deniable utility-improving deviation. For instance, Example 5.16 is a proof that the Vickrey auction is not credible in this sense. Akbarpour and Li [9] study both single-shot (a.k.a. "static") mechanisms and mechanisms that require many rounds (such as ascending auctions); the former type are much more practical for blockchain transaction fee mechanisms. Interestingly, one of the main results in [9, Theorem 3.7] is that first-price auctions with an exogenously restricted bid space are the only static credible mechanisms.[24] All of the MMIC mechanisms appearing in this report—first-price auctions (Example 5.15), the 1559 mechanism (Theorem 6.4), and the tipless mechanism of Section 8.5 (Theorem 8.8)—can be viewed as first-price auctions with different restricted bid spaces.[25]

Returning to status quo first-price auctions, the argument in Example 5.15 highlights two of their properties:

(i) excluding real transactions suggested by the allocation rule strictly decreases myopic miner utility;

(ii) including fake transactions does not increase myopic miner utility.

We next pursue a stronger version of property (ii).

5.3.3 γ-Costly Transaction Fee Mechanisms

A stronger version of property (ii) would state that, as with excluding real transactions, fake transactions significantly decrease myopic miner utility. First-price auctions possess this stronger property when the maximum block size constraint is binding (as fake transactions then displace real ones) or when the marginal cost μ is large. Otherwise, a miner can devote any extra room in a block to fake transactions without suffering a significant cost.

The next definition formalizes this stronger version of property (ii).

Definition 5.18 (γ-Costly Transaction Fee Mechanism) A TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ is γ*-costly* if, for every on-chain history $B_1, B_2, \ldots, B_{k-1}$, mempool M, fake transactions F, and block $B_k \subseteq M \cup F$ chosen by a miner, the fake transactions of B_k decrease myopic miner utility (5) by at least γ per unit of gas:

$$u(F, B_k) \leq \underbrace{u(\emptyset, B_k \cap M)}_{\text{utility w/out fake txs}} - \underbrace{\gamma \cdot \sum_{t \in F} g_t}_{\text{cost of fake txs}} .$$

[24] The results in [9] assume a computationally unbounded auctioneer. Ferreira and Weinberg [26] explore what other credible mechanisms are possible assuming a computationally bounded auctioneer and the existence of cryptographically secure hash functions.

[25] First-price auctions correspond to the bid space $[0, \infty)$; the 1559 mechanism to the bid space $\{\text{"no bid"}\} \cup [r, \infty)$, where r is the block's base fee; and the tipless mechanism to the bid space $\{\text{"no bid"}, r + \delta\}$, where r is the block's base fee and δ is a protocol-defined hard-coded tip.

For example, first-price auctions are μ-costly, where μ is the marginal cost of gas to a miner, and are not γ-costly for any $\gamma > \mu$. We'll see later (Corollary 6.5) that the transaction fee mechanism proposed in EIP-1559 is generally γ-costly for larger values of γ, and in this sense more aggressively punishes fake transactions.

5.4 Incentive Compatibility (Users)

Next we formalize what it means for a TFM to be game-theoretically sound from the perspective of users—intuitively, that there is an "obvious' optimal bid" when creating a new transaction. This is also our definition of a "good user experience" is the sense of easy fee estimation (see Section 4).

5.4.1 User Utility Function

Recall from Section 5.1 that the value v_t of a transaction t is the maximum gas price the transaction's creator would be willing to pay for its inclusion in the current block. We assume that a user bids in order to maximize its net gain (i.e., the value for inclusion minus the cost for inclusion). To reason about the different possible bids for a transaction t submitted to a mempool M, we use $M(b_t)$ to denote the result of adding the transaction t with bid b_t to M. For simplicity, we assume that each transaction in the current mempool has a distinct creator.

Definition 5.19 (User Utility Function) For a TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$, on-chain history $B_1, B_2, \ldots, B_{k-1}$, and mempool M, the utility of the originator of a transaction $t \notin M$ with value v_t and bid b_t is

$$u_t(b_t) := \left(v_t - \underbrace{p_t(B_1, \ldots, B_{k-1}, B_k)}_{\text{payment to miner (per-gas-unit)}} - \underbrace{q_t(B_1, \ldots, B_{k-1}, B_k)}_{\text{fee burn (per-gas-unit)}} \right) \cdot g_t \qquad (6)$$

if t is included in $B_k = \mathbf{x}(B_1, \ldots, B_{k-1}, M(b_t))$ and 0 otherwise.

In (6), we highlight the dependence of the utility function on the argument that is directly under a user's control, the bid b_t submitted with the transaction. We assume that a transaction creator bids to maximize the utility function in (6).[26]

5.4.2 Bidding Strategies and Ex Post Nash Equilibrium

Intuitively, "easy fee estimation" should mean that the "obvious" bidding strategy is optimal. Formally, a *bidding strategy* is a function b^* that specifies a bid $b^*(v_t)$ for a transaction t as a function of the value v_t of that transaction. A bidding strategy is a function of the value v_t only (which is known to the transaction creator) and not, for example, bids submitted by competing transactions (which are not).[27] For example, a plausible bidding strategy in a first-price auction is to shade one's bid, but not by too much, perhaps by setting $b^*(v_t) = .75v_t$ for all v_t.

[26] While the creator of a transaction t has no direct control over $\mathbf{x}$, $\mathbf{p}$, or $\mathbf{q}$, its bid b_t is embedded in $M(b_t)$ and therefore can affect $B_k = \mathbf{x}(B_1, \ldots, B_{k-1}, M(b_t))$. This, in turn, can affect $p_t(B_1, \ldots, B_{k-1}, B_k)$ and $q_t(B_1, \ldots, B_{k-1}, B_k)$. For example, whether or not $x_t(B_1, \ldots, B_{k-1}, M(b_t)) = 1$ generally depends on whether or not b_t is large relative to the bids of competing transactions in M.

[27] A bidding strategy can depend also on the blockchain history (e.g., with EIP-1559, on the current base fee). For the purposes of a single-block game-theoretic analysis, we can take the history as fixed and suppress this dependence in the notation.

Suppose we have in mind an "obvious" bidding strategy $b^*(\cdot)$ for users to employ. What does it mean that bidding in this obvious way is "always optimal"? The answer is formalized by the concept of a symmetric ex post Nash equilibrium (symmetric EPNE). Intuitively, obvious bidding should maximize a user's utility as long as all the other users are also bidding in the obvious way.[28]

Definition 5.20 (Symmetric Ex Post Nash Equilibrium (Symmetric EPNE)) Fix a TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ and the on-chain history $B_1, B_2, \ldots, B_{k-1}$. A bidding strategy $b^*(\cdot)$ is a *symmetric ex post Nash equilibrium (symmetric EPNE)* if, for every mempool M in which all transactions' bids were set according to this strategy, and for every transaction $t \notin M$ with value v_t, bidding $b^*(v_t)$ maximizes the utility (6) of t's creator.

Crucially, following the bid recommendation $b^*(v_t)$ of a symmetric EPNE does not require reasoning about competing transactions in M, other than keeping the faith that their bids were set according to the bid recommendations of the symmetric EPNE.[29]

We can now define a TFM to be incentive-compatible from the user perspective if there's always an obvious bidding strategy in the form of a symmetric EPNE.

Definition 5.21 (Incentive-Compatibility for Users (UIC)) A TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ is *incentive-compatible for users (UIC)* if, for every on-chain history $B_1, B_2, \ldots, B_{k-1}$, there is a symmetric EPNE.

In this report, we identify "mechanisms with easy fee estimation" and "mechanisms with good UX" with the UIC condition of Definition 5.21.

Example 5.22 (First-Price Auctions Are Not UIC) First-price auctions are not easy to reason about, in the sense that they are not UIC. Intuitively, the utility-maximizing bid depends on the precise numerical values of others' bids, and not merely on the qualitative knowledge that they are following a particular bidding strategy.

For example, consider a block with room for one transaction, a transaction t with value $v_t = 10$, and suppose that all transactions other than t use the same bidding strategy $b^*(v_s) = .75 \cdot v_s$. If the highest value of v_s of any transaction $s \neq t$ is 10, then the highest bid by any such transaction will be 7.5, and the utility-maximizing bid for t's creator will be 7.51. If the highest other value is 8, the optimal bid is 6.01; and so on. The key point is that the optimal bid to include with the transaction is a function not only of that transaction's value, but also of the values of the competing transactions (even after assuming that all their bids are set using a known bidding strategy $b^*(\cdot)$).

Thus, in a precise sense, first-price auctions do not offer "good UX" in the form of an easy-to-follow optimal bid recommendation. We'll see later (Theorem 6.8) that the transaction fee mechanism proposed in EIP-1559 is UIC except during periods of rapidly increasing demand.

[28] "Symmetric" refers to the fact that the obvious bidding strategy $b^*(\cdot)$ is the same for every transaction t.

[29] An even stronger notion is a *dominant-strategy equilibrium*, in which $b^*(v_t)$ is optimal for t's creator no matter what the other users do. "Obvious bidding" is not a dominant-strategy equilibrium in the transaction fee mechanism proposed in EIP-1559 (see Remark 6.10), but it is in a variant with hard-coded tips (see Theorem 8.9 and footnote 56).

5.5 Off-Chain Agreements

The game-theoretic guarantees in Section 5.3 concern attacks that manipulate the contents of a block (by including fake transactions, or more generally deviating from the allocation intended by the transaction fee mechanism). This section treats a different type of attack that is also implementable at the time scale of a single block, namely collusive agreements between miners and users. Recall that a set T of transactions is *feasible* if the total gas $\sum_{t \in T} g_t$ is at most the maximum block size G.

Definition 5.23 (Off-Chain Agreement (OCA)) For a feasible set T of transactions and a miner m, an *off-chain agreement (OCA)* between T's creators and m specifies:

(i) a bid vector $\mathbf{b}$, with b_t indicating the bid to be submitted with the transaction $t \in T$;

(ii) a per-gas-unit ETH transfer τ_t from the creator of each transaction $t \in T$ to the miner m.

In an OCA, each creator of a transaction t agrees to submit t on-chain with a bid of b_t while transferring τ_t per unit of gas to the miner m off-chain; the miner, in turn, agrees to mine a block $B(\mathbf{b})$ comprising the transactions in T (with on-chain bids $\mathbf{b}$).

Example 5.24 (Moving Payments Off-Chain) To get a feel for OCAs, imagine a first-price auction in which 50% of the revenue is burned and the other 50% is transferred to the miner. (See also Section 8.2.) Miners and users could then collude as follows:

1. Users bid zero on-chain and communicate off-chain what they would have bid in a standard first-price auction.
2. Miners keep 75% of the (off-chain) bids of the transactions they include, with the other 25% refunded to those transactions' creators.

In the notation of Definition 5.23, this is the OCA $(\mathbf{b}, \boldsymbol{\tau})$ in which $\mathbf{b} = \mathbf{0}$ and $\tau_t = .75b'_t$, where b'_t denotes what t's creator would have bid in a first-price auction without fee-burning. Compared to the "honest" on-chain outcome with bids $\mathbf{b}'$, miners earn 50% more revenue and users enjoy a 25% discount, both at the expense of the network.

Given a TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ and on-chain history $B_1, B_2, \ldots, B_{k-1}$, the utility of t's creator from such an OCA $(\mathbf{b}, \boldsymbol{\tau})$ is given by the right-hand side of (6), less its transfer to the miner:

$$(v_t - p_t(B_1, \ldots, B_{k-1}, B(\mathbf{b})) - q_t(B_1, \ldots, B_{k-1}, B(\mathbf{b})) - \tau_t) \cdot g_t. \tag{7}$$

(Users not part of T receive zero utility.) The miner's utility is given by the sum of on-chain and off-chain payments received, less the costs incurred:

$$\sum_{t \in T} (p_t(B_1, B_2, \ldots, B_{k-1}, B(\mathbf{b})) + \tau_t - \mu) \cdot g_t. \tag{8}$$

Adding up these utility functions—one per transaction $t \in T$, plus one for the miner—results in the joint utility enjoyed by all parties in an OCA $(\mathbf{b}, \boldsymbol{\tau})$:

$$u_{T,m}(\mathbf{b}, \boldsymbol{\tau}) := \sum_{t \in T} (v_t - q_t(B_1, \ldots, B_{k-1}, B(\mathbf{b})) - \mu) \cdot g_t.$$

From the coalition's perspective, on-chain and off-chain payments from the users to the miner (the p_t's and τ_t's) remain within the coalition and thus cancel out; the fee burn (the q_t's) is transferred outside the coalition (to the network) and is therefore a loss. Thus, the point of an OCA is to maximize the joint utility—the amount of transaction value that is not lost to the protocol or to the miner's costs.

Definition 5.25 (Joint Utility) For an on-chain history $B_1, B_2, \ldots, B_{k-1}$, the *joint utility* of the miner and users for the block B_k is

$$\sum_{t \in B_k} (v_t - q_t(B_1, B_2, \ldots, B_{k-1}, B_k) - \mu) \cdot g_t. \tag{9}$$

We assume that miners and users act to maximize their joint utility. Using transfers, a miner and users can then split this joint utility among themselves in an arbitrary way.[30] For this reason, when analyzing OCAs, we can focus on the joint utility (9) of the miner and the creators of the included transactions, without concern about how it might be split among them and the creators of the excluded transactions.

A TFM is then *OCA-proof* if, for every OCA, there is an equally good on-chain outcome. For a set of transactions U and bids $\mathbf{b}$ for those transactions, we denote by $U(\mathbf{b})$ the corresponding mempool.

Definition 5.26 (OCA-Proof) A TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ is *OCA-proof* if, for every on-chain history B_1, $B_2, \ldots, B_{k-1}$ and set U of outstanding transactions, there exists bids $\mathbf{b}^*$ for the transactions of U such that, for the resulting on-chain outcome $B_k = \mathbf{x}(B_1, B_2, \ldots, B_{k-1}, U(\mathbf{b}^*))$,

$$\underbrace{\sum_{t \in B_k} (v_t - q_t(B_1, \ldots, B_{k-1}, B_k) - \mu) \cdot g_t}_{\text{joint utility of on-chain outcome}} \geq u_{T,m}(\mathbf{b}, \boldsymbol{\tau}) \tag{10}$$

for every feasible subset $T \subseteq U$ of transactions and OCA $(\mathbf{b}, \boldsymbol{\tau})$ between their creators and the miner m.

In other words, if a TFM is *not* OCA-proof, there are scenarios in which a miner and users can collude to achieve higher joint utility—and, after defining appropriate transfers, higher individual utilities—than in any on-chain outcome.

Intuitively, first-price auctions are OCA-proof because off-chain payments can be costlessly replaced by on-chain bids. The next example formally verifies Definition 5.26.

Example 5.27 (First-Price Auctions Are OCA-Proof) Consider a set U of transactions and set $b_t^* = v_t$ for every $t \in U$. Then, because $\mathbf{q}^f$ is the all-zero function (Example 5.8), the objective (4) maximized by the allocation rule $\mathbf{x}^f$ is identical to the joint utility (9). Thus, the joint utility of the on-chain outcome with bids $\mathbf{b}^*$ cannot be improved upon by any OCA.

[30] For example, suppose an OCA increases the joint utility of a coalition by increasing the utility of six users by 1 ETH each while decreasing the miner's utility by 5 ETH. The OCA transfers can then be adjusted so that all parties enjoy strictly higher individual utility, for example by sending an extra $\frac{11}{12}$ ETH from each of these users to the miner. Additional transfers can be used to also strictly increase the utility of the creators of the transactions excluded from the block B_k.

Remark 5.28 (OCA-Proofness and Fee Burning) OCAs are the biggest game-theoretic driver for the why and the how of the fee burn in the transaction fee mechanism proposed in EIP-1559. For example, adding a fee burn to a first-price auction destroys its OCA-proofness (Section 8.2). Meanwhile, because of OCAs, a history-dependent base fee has no teeth unless revenue from it is burned or otherwise withheld from the miner (Section 8.1).

6 Formal Analysis of the 1559 Mechanism with Myopic Miners

This section investigates to what extent the transaction fee mechanism proposed in EIP-1559—henceforth, the *1559 mechanism*—satisfies the three game-theoretic guarantees identified in Section 5 (MMIC, UIC, and OCA-proofness). Section 6.1 translates the description of the mechanism in Section 2.3 into the formalism introduced in Section 5. Sections 6.2–6.4 prove that the mechanism is always MMIC and OCA-proof, and is UIC except during periods of rapidly increasing demand.

Game-Theoretic Guarantees for the 1559 Mechanism

1. Myopic miners are incentivized to follow the intended allocation rule, and are strictly disincentivized from including fake transactions in a block.
2. Except in periods of a large and sudden demand spike, there are "obvious" optimal bids for users: set a transaction's fee cap to its value and its tip to cover the marginal cost of gas to the miner.
3. Miners and users can never improve their joint utility through an off-chain agreement.

6.1 The 1559 Mechanism

Recall from Section 2.3 that, in the 1559 mechanism, each block is associated with a base fee that is fixed by the history of past blocks and independent of the contents of the current block; we denote by $\alpha(B_1, B_2, \ldots, B_{k-1})$ the base fee for the next block that is determined by a particular history $B_1, B_2, \ldots, B_{k-1}$. The specific function α proposed in EIP-1559 is the iteration of the base fee update rule in (2), although these details will not be important for the single-block game-theoretic analysis carried out in this section.

Recall also that, in EIP-1559, each transaction specifies a tip δ_t and a fee cap c_t. These two parameters induce a bid b_t for the transaction with respect to any given base fee r, namely

$$b_t = \min\{r + \delta_t, c_t\}. \tag{11}$$

Definition 6.1 (1559 Allocation Rule) For each history $B_1, B_2, \ldots, B_{k-1}$ and corresponding base fee $r = \alpha(B_1, B_2, \ldots, B_{k-1})$, the (intended) allocation rule $\mathbf{x}^*$ of the 1559 mechanism is to include a feasible subset of outstanding transactions that maximizes the sum of the gas-weighted bids, less the gas costs and total base fee paid. That is, the x_t^*'s are assigned 0-1 values to maximize

$$\sum_{t \in M} x_t^*(B_1, B_2, \ldots, B_{k-1}, M) \cdot (b_t - r - \mu) \cdot g_t, \tag{12}$$

subject to the block size constraint (3).

The payment rule transfers the difference between the bid and the base fee to the miner.

Definition 6.2 (1559 Payment Rule) In the 1559 mechanism, letting $r = \alpha(B_1, B_2, \ldots, B_{k-1})$,

$$p^*_t(B_1, B_2, \ldots, B_{k-1}, B_k) = b_t - r$$

for all $B_1, B_2, \ldots, B_k$ and $t \in B_k$.

The burning rule burns the base fee.

Definition 6.3 (1559 Burning Rule) In the 1559 mechanism, letting $r = \alpha(B_1, B_2, \ldots, B_{k-1})$,

$$q^*_t(B_1, B_2, \ldots, B_{k-1}, B_k) = r$$

for all $B_1, B_2, \ldots, B_k$ and $t \in B_k$.

Formally, the *1559 mechanism* is the TFM mathematically encoded by the triple of rules $(\mathbf{x}^*, \mathbf{p}^*, \mathbf{q}^*)$ described in Definitions 6.1–6.3.

6.2 The 1559 Mechanism Is Incentive Compatible for Myopic Miners

This section evaluates the 1559 mechanism from the perspective of myopic miners, and specifically the MMIC property (Definition 5.14) and γ-costliness (Definition 5.18).

Theorem 6.4 (The 1559 Mechanism is MMIC) *The 1559 mechanism* $(\mathbf{x}^*, \mathbf{p}^*, \mathbf{q}^*)$ *is MMIC.*

Proof: Fix an on-chain history $B_1, B_2, \ldots, B_{k-1}$, a mempool M, and a marginal cost of gas $\mu \geq 0$ (as in Remark 3.2). Let r denote the corresponding base fee $\alpha(B_1, B_2, \ldots, B_{k-1})$ for the current block. Substituting in Definitions 6.2 and 6.3, myopic miner utility (5) equals

$$u(F, B_k) = \underbrace{\sum_{t \in B_k \cap M} (b_t - r - \mu) \cdot g_t}_{\text{net revenue from } B_k} - \underbrace{\sum_{t \in B_k \cap F} (r + \mu) \cdot g_t}_{\text{cost of fake txs}}, \tag{13}$$

where B_k denotes the transactions included by the miner and F the fake transactions that it creates. Included fake transactions strictly increase the second term (by $r + \mu$ per unit of gas) while leaving the first unaffected, so a myopic miner will only include real transactions in B_k. In this case, myopic miner utility equals

$$\sum_{t \in B_k} (b_t - r - \mu) \cdot g_t,$$

which is identical to the quantity (12) maximized by the allocation rule $\mathbf{x}^*$ (Definition 6.1). Thus, myopic miner utility is maximized by following the allocation rule and setting B_k equal to $\mathbf{x}^*(B_1, B_2, \ldots, B_{k-1}, M)$. ■

From the expression (13) for myopic miner utility in the 1559 mechanism, we can see immediately that it is γ-costly (Definition 5.18) for $\gamma = r + \mu$.

Corollary 6.5 (The 1559 Mechanism is $(r+\mu)$-Costly) *Fix an on-chain history $B_1, B_2, \ldots, B_{k-1}$ and corresponding base fee $r = \alpha(B_1, B_2, \ldots, B_{k-1})$ for the current block, a mempool M, and a marginal cost of gas $\mu \geq 0$. The 1559 mechanism is $(r+\mu)$-costly.*

Remark 6.6 (Role of the Fee Burn) If the base fee was paid to miners rather than burned, the 1559 mechanism would only be μ-costly and fake transactions would be only mildly disincentivized. The primary motivation for the fee burn, however, is to rule out its evasion by off-chain agreements (see Section 8.1).

6.3 The 1559 Mechanism Is Typically Incentive Compatible for Users

The 1559 mechanism is always incentive compatible for myopic miners, no matter what the current base fee and demand for block space (Theorem 6.4). We next show that the mechanism is also incentive compatible for users, except in periods of rapidly increasing demand.

6.3.1 Excessively Low Base Fees

The next definition is a proxy for a period of rapidly increasing demand.

Definition 6.7 (Excessively Low Base Fee) Let μ denote the marginal cost per unit of gas. A base fee r is *excessively low* for a mempool M of transactions if the demand at price $r+\mu$ exceeds the maximum block size G:

$$\underbrace{\sum_{t \in M \,:\, v_t \geq r+\mu} g_t}_{\text{demand at price } r+\mu} > G. \tag{14}$$

Excessively low base fees arise from large and sudden demand spikes. In Example 3.3 in Section 3.2, for instance, none of the eight periods suffer from an excessively low base fee, despite the sudden doubling of demand. Modifying that example so that demand more than doubles in period 2, there is a sequence of periods with excessively low base fees, ending once the base fee has increased enough to bring demand back down below 25M gas (Table 2).

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8
Demand	Low	High	High	High	High	High	High	Low
EIP-1559 Base Fee	33.33	33.33	37.5	42.18	47.46	53.39	60.06	66.19
EIP-1559 Block Size	12.5M	25M	25M	25M	25M	25M	24.49M	10.04M
Excessively low?	No	Yes	Yes	Yes	Yes	Yes	No	No

Table 2: An example of excessively low base fees due to a large and sudden jump in demand. The marginal cost μ of gas is 0. "Low" demand means the demand curve $D(p) = 15000000 - 75000p$; "high" means the demand curve $D(p) = 35000000 - 175000p$. (Here "demand" means the total gas consumed by all pending transactions with a value of p or more.) The second and third rows show the joint evolution of the base fee and block size under the EIP-1559 mechanism, assuming that the base fee matches the market-clearing price in period 1 and that all users submit a bid equal to the minimum of their value and the base fee. Periods 2–6 suffer from excessively low base fees.

6.3.2 The 1559 Mechanism Is UIC Except with Excessively Low Base Fees

When the base fee is excessively low, users must compete for scarce block space through their tips, and the 1559 mechanism effectively reverts back to a first-price auction. As first-price auctions are essentially never UIC (see Example 5.22), the 1559 mechanism is not UIC when the base fee is excessively low. The good news is that an excessively low base fee is the only reason why the 1559 mechanism might fail to be UIC. That is, whenever the base fee is not excessively low, there is an "obvious optimal bid" in the form of a symmetric EPNE (Definition 5.20). This optimal bid corresponds to setting a transaction's fee cap equal to its creator's value (i.e., $c_t = v_t$), and a transaction's tip equal to the marginal cost of gas to a miner (i.e., $\delta_t = \mu$).

Theorem 6.8 (The 1559 Mechanism Is Typically UIC) *Fix an on-chain history $B_1, B_2, \ldots, B_{k-1}$ and corresponding base fee $r = \alpha(B_1, B_2, \ldots, B_{k-1})$, a marginal cost μ of gas to miners, and a mempool M of transactions for which r is not excessively low. The bidding strategy*

$$b^*(v_t) = \min\{r + \mu, v_t\} \tag{15}$$

constitutes a symmetric EPNE under the 1559 mechanism.

Proof: Suppose each creator of a transaction $t \in M$ sets its bid according to the strategy $b^*(\cdot)$ in (15); we need to show that no creator could increase its expected utility (6) by changing its bid (holding the bids of other transactions fixed).

The objective (12) of the 1559 allocation rule prescribes including precisely the transactions $t \in M$ with $b_t \geq r + \mu$. Because $b^*(v_t) = \min\{r + \mu, v_t\}$ for all $t \in M$, these are precisely the transactions $t \in M$ with $v_t \geq r + \mu$. In particular, because r is not excessively low for M, this allocation is feasible:

$$\underbrace{\sum_{t \in M \,:\, b^*(v_t) \geq r+\mu} g_t}_{\text{gas of included txs}} \quad = \quad \underbrace{\sum_{t \in M \,:\, v_t \geq r+\mu} g_t}_{\text{demand at price } r+\mu} \quad \leq \quad G. \tag{16}$$

There are two types of transactions t to consider, high-value ($v_t \geq r + \mu$) and low-value ($v_t < r + \mu$); see also Table 3. When all bids are set according the strategy $b^*(\cdot)$ in (15), the former transactions are included (and pay $b^*(v_t) = r + \mu$ per unit of gas) while the latter are excluded (and pay nothing). The utility (6) of t's creator is $(v_t - r - \mu) \cdot g_t \geq 0$ if t is a high-value transaction and 0 otherwise. Every alternative bid $\hat{b}_t$ for a high-value transaction either has no effect on its creator's utility (if $\hat{b}_t \geq r + \mu$) or leads to t's exclusion from the block (if $\hat{b}_t < r + \mu$) and reduces this utility from $(v_t - r - \mu) \cdot g_t$ to 0. Every alternative bid $\hat{b}_t$ for a low-value transaction either has no effect on its creator's utility or leads to t's inclusion in the block; the latter can only occur when $\hat{b}_t \geq r + \mu$, in which case the creator's utility drops from 0 to $(v_t - \hat{b}_t) \cdot g_t < 0$. We conclude that there is no alternative bid for any transaction of M that increases its creator's utility. ■

Theorem 6.8 and its proof show that, at its symmetric EPNE, the 1559 mechanism acts a posted-price mechanism (Section 4.2) except when the base fee is excessively low.

The 1559 Mechanism Is Typically a Posted-Price Mechanism

The 1559 mechanism acts as a posted-price mechanism at the price $r + \mu$, where r

	Low-Value ($v_t < r + \mu$)	High-Value ($v_t \geq r + \mu$)
Bid at EPNE	v_t	$r + \mu$
Utility at EPNE	0	$(v_t - r - \mu) \cdot g_t \geq 0$
Utility of Alternative	$\leq (v_t - r - \mu) \cdot g_t < 0$	0

Table 3: Proof of Theorem 6.8. For both low- and high-value transactions, no unilateral deviation from the symmetric EPNE bid can increase a user's utility.

is the base fee and μ is the marginal cost of gas, except during periods of rapidly increasing demand.

Remark 6.9 (Welfare Properties of the 1559 Mechanism) An attractive property of the symmetric EPNE in (15) is that the outcome perfectly differentiates between high-value ($v_t \geq r + \mu$) and low-value ($v_t < r + \mu$) transactions, including the former while excluding the latter. This outcome can be viewed as a market-clearing outcome (Section 3.1) with respect to a supply of G^* gas, where G^* denotes the demand at price $r + \mu$.

Remark 6.10 (The Obvious Bid Is Not a Dominant Strategy) The symmetric EPNE (15) in the proof of Theorem 6.8 is not a dominant-strategy equilibrium in the sense of footnote 29. The issue arises when the creators of other transactions overstate their fee caps, in which case the base fee could become excessively low with respect to the stated demand (even though it is not with respect to the true demand). In particular, the equality in (16) need not hold if other transactions' bids are set arbitrarily.

Remark 6.11 (Expected Frequency of Excessively Low Base Fees) Demand for EVM computation has generally been volatile, at both short and long time scales. For this reason, one would expect at least occasional excessively low base fees. It would be interesting to predict, perhaps based on experiments using historical demand data, the likely frequency of excessively low base fees in a post-EIP-1559 world.

6.4 The 1559 Mechanism Is OCA-Proof

Finally, we show that, under the 1559 mechanism, miners and users cannot improve their joint utility through off-chain agreements. A key driver of this result is that the fee burn (per unit of gas) does not depend on the current actions of the miner or users (cf., Section 8.2).

Theorem 6.12 (The 1559 Mechanism is OCA-Proof) *The 1559 mechanism* $(\mathbf{x}^*, \mathbf{p}^*, \mathbf{q}^*)$ *is OCA-proof.*

Proof: Fix an on-chain history $B_1, B_2, \ldots, B_{k-1}$ and corresponding base fee $r = \alpha(B_1, B_2, \ldots, B_{k-1})$. Consider a set U of transactions and set $b_t^* = v_t$ for every $t \in U$. Then, because $\mathbf{q}^*$ is the constant function always equal to r (Definition 6.3), the objective (12) maximized by the allocation rule $\mathbf{x}^*$ is identical to the joint utility (9). Thus, the joint utility of the on-chain outcome with bids $\mathbf{b}^*$ cannot be improved upon by any OCA. ■

7 Miner Collusion at Longer Time Scales

Section 6 demonstrates that the 1559 mechanism enjoys several game-theoretic guarantees at the time scale of a single block. But what about longer time scales? For example, to achieve the typically-UIC guarantee in Theorem 6.8, the mechanism introduces a history-dependent base fee that is burned; a natural worry is that miners may be incentivized to manipulate and artificially decrease this base fee over time.

This section investigates the incentives for miner collusion, both under the status quo and under EIP-1559. Section 7.1 formalizes "extreme miner collusion" through a thought experiment in which a single miner controls 100% of Ethereum's hashrate. Section 7.2 identifies the revenue-maximizing strategy for such a miner in a first-price auction; in some cases, the miner is incentivized to artificially restrict the supply of EVM computation in order to boost the bids submitted by creators of high-value transactions. Section 7.3 repeats the exercise for the 1559 mechanism and determines that the outcome of extreme collusion would be similar to that with today's first-price auctions. Section 7.4 classifies different types of miner collusion and reviews to what extent each type appears to occur in Ethereum at present. Section 7.5 argues that the game-theoretic impediments to double-spend, censorship, denial-of-service, and revenue-maximizing 100% miner strategies (including base fee manipulation) appear as strong under EIP-1559 as under the status quo. Finally, Section 7.6 brainstorms possible reasons for why miner collusion might nevertheless be more likely under EIP-1559 than it is today.

7.1 Extreme Collusion: The 100% Miner Thought Experiment

The fidelity of the myopic miner model of Sections 5–6 depends on the degree of decentralization in Ethereum mining. For example, with extreme decentralization, such as the hashrate being spread equally across millions of non-colluding miners, any given miner mines a block so rarely that there is no point to non-myopic strategies (i.e., strategies that forego immediate rewards in favor of future rewards). In particular, in the 1559 mechanism, because the base fee is set by past history and independent of the current block, no such miner will be interested in manipulating it.

To meaningfully study miner deviations such as base fee manipulation, we must therefore consider miners (or tightly coordinated mining pools) that possess a significant fraction of the total hashrate and strategize at time scales longer than a single block.[31] To get the lay of the land, we next investigate both first-price auctions and the 1559 mechanism in the opposite extreme scenario in which *all* of the hashrate is controlled by a single miner or, equivalently, a perfectly coordinated cartel comprising all of the miners.[32]

The 100% Miner Thought Experiment

1. A single miner controls 100% of the hashrate.
2. The miner acts to maximize its net revenue received from transaction fees over a significant period of time (e.g., thousands of blocks).

[31] We continue to assume that users are myopic, and bid to maximize their utility in the current block (Definition 5.19). Simulations by Monnot [45] suggest that more complex user strategies do not significantly change the behavior of the mechanism proposed in EIP-1559.

[32] A similar approach is taken by Hasu et al. [32] in the context of Bitcoin and Zoltu [59] in the context of EIP-1559.

3. The demand curve (see Section 3) is the same for every block, independent of the miner's actions, and known to the miner.

The second assumption clarifies that the thought experiments in Sections 7.2 and 7.3 will not consider off-chain rewards, for example from a double-spend attack, in order to isolate incentive issues specific to the transaction fee mechanism. The point of the third assumption is to stack the deck against a protocol by making it as easy as possible for a miner or cartel of miners to identify and carry out optimal deviations from the protocol's prescriptions.

7.2 First-Price Auctions with a 100% Miner

What would a 100% miner do under the status quo of first-price auctions? Let $D(p)$ denote the demand curve—the total gas demanded at a gas price of p gwei. We assume that $D(p)$ is a continuous and strictly decreasing function, and that $D(p) = 0$ once p is sufficiently large. We continue to assume that the demand curve is exogenous, the same for every block, and known to the miner.

We consider strategies of the following form:

Strategies for a 100% Miner

1. **Price-setting:** for a gas price p with $D(p) \leq G$, include a transaction in the block if and only if its gas price is at least p. (As usual, G denotes the maximum block size.)[33]
2. **Quantity-setting:** for a quantity $q \leq G$, include the transactions with the highest gas prices, up to a limit of q on the total gas.[34]

In our model, these two types of strategies are equivalent—a price-setting strategy at the price p has the same effect as a quantity-setting strategy at the quantity $q = D(p)$. In either case, a creator of a transaction t with $v_t \geq p$ should be expected to respond by bidding the fixed price p (enough for inclusion in the block), and one with $v_t < p$ to bid something between 0 and v_t (in any case, being excluded from the block).

Because there are no dependencies between first-price auctions in different blocks and no fee burn, a 100% miner maximizes its net revenue by maximizing its revenue from each block separately. For a single block, the revenue earned by a miner using a price-setting strategy with price p (or the equivalent quantity-setting strategy) is the price times the quantity willing to pay it:[35]

$$R(p) := p \cdot D(p). \tag{17}$$

For ease of exposition, in this section we focus on demand curves for which the revenue (17) is a strictly concave function (as is the case with, for example, a linear demand curve).

Because a 100% miner can be thought of as a monopoly on EVM computation, there is an obvious price and quantity to focus on:

[33] For an analogy, think of a consultant with a unique skill set committing to an hourly rate.

[34] For an analogy, think of a restriction on oil production set by the Organization of the Petroleum Exporting Countries (OPEC).

[35] For simplicity, we assume in this section that the marginal cost of gas to a miner—the parameter μ in Sections 5–6—is zero. The conclusions of this section remain the same for a positive marginal cost.

Definition 7.1 (Monopoly Price and Quantity) Consider a maximum block size G and a demand curve $D(\cdot)$ for which the revenue (17) is a strictly concave function of price. If $\bar{p}$ attains the maximum in (17), then:

(a) the *monopoly price* is the revenue-maximizing price or the market-clearing price, whichever is larger:[36]

$$p^* := \max\{\bar{p}, D^{-1}(G)\}; \tag{18}$$

(b) the *monopoly quantity* is the revenue-maximizing quantity or the maximum block size, whichever is smaller:

$$q^* := D(p^*) = \min\{D(\bar{p}), G\}. \tag{19}$$

Example 7.2 (Monopoly Prices and Quantites) Suppose the maximum block size G is 12.5M gas and the demand curve is $D(p) = 30000000 - 150000p$ (as in Figure 1). The revenue (17) as a function of price is $30000000p - 150000p^2$. This function is differentiable and strictly concave, so its unique maximum $\bar{p}$ is the point at which the derivative $30000000 - 300000p$ equals 0. Thus, $\bar{p} = 100$ gwei and $D(\bar{p}) = 15$M gas. This exceeds the maximum block size of 12.5M gas, and hence the monopoly price is the market-clearing price $D^{-1}(12.5\text{M}) = 116\frac{2}{3}$ gwei.

If instead the demand curve was $D(p) = 20000000 - 150000p$, $\bar{p}$ would be $66\frac{2}{3}$ gwei and $D(\bar{p})$ would be 10M gas. In this case, the monopoly price is strictly higher than the market-clearing price (of 50 gwei) and the monopoly quantity is strictly smaller than the maximum block size.

Price-setting at the monopoly price or quantity-setting at the monopoly quantity both have the effect of maximizing revenue (17) subject to the maximum block size. That is, these are precisely the optimal strategies for a 100% miner:

Optimal Strategies for a 100% Miner (First-Price Auctions)

- A 100% miner would price-set at the monopoly price or quantity-set at the monopoly quantity.
- If the monopoly quantity is the maximum gas size (equivalently, the monopoly price is the market-clearing price), a 100% miner would not deviate from the intended allocation rule of a first-price auction (Example 5.4).

We conclude that, with a first-price auction, extreme miner collusion can increase transaction fee revenue if and only if the monopoly quantity is less than the maximum block size; in this case, miners can boost revenues by artificially restricting the supply of EVM computation, thereby forcing the creators of high-value transactions to submit higher bids for their inclusion.

Remark 7.3 (Detecting a Price- or Quantity-Setting Strategy) Suppose a cartel of miners implemented a quantity-setting strategy (with quantity less than 12.5M gas), or the corresponding price-setting strategy. Would anyone notice? Naively executed, persistent underfull blocks would be a dead giveaway. But if miners include fake transactions to keep all the blocks full, such a strategy could be difficult to conclusively detect.

[36] See Section 3.1 for the definition of a market-clearing price.

7.3 EIP-1559 with a 100% Miner

EIP-1559, described in Section 2.3, would have two immediate consequences for the 100% miner thought experiment. First, a miner would strive to simultaneously maximize the transaction fee revenue (as in a first-price auction) and minimize the amount of these fees lost to the fee burn. Second, a miner's allocation decision in one block would affect the base fee (and hence net revenue earned) in future blocks. Now what's the miner's optimal strategy?

First suppose that the monopoly quantity (Definition 7.1) is at most the target block size of 12.5M gas. (Recall that the maximum block size G is double this amount under EIP-1559.) In this case, the maximum block size may as well be 12.5M gas—a 100% miner will never use more than this, as doing so would decrease its net revenue both in the current block (for including more gas than the monopoly quantity) and in future blocks (because of the increased base fee, as per (2)). Thus, the best-case scenario for a 100% miner is to match the revenue of a 100% miner under the status quo (Section 7.2) while simultaneously paying no fee burn. A 100% miner can closely approximate this best-case scenario:

Optimal Strategy for a 100% Miner (Monopoly Quantity $\leq$ 12.5M gas)

1. Drive the base fee to zero (or its minimum amount) from its initial value, for example by publishing a sequence of empty blocks.
2. For all future blocks, proceed as a 100% miner would with first-price auctions, by using the monopoly quantity-setting strategy.

Because the monopoly quantity is at most the target block size, the base fee will remain at its minimum value forevermore. Notably, in this case, the outcome of extreme miner collusion is essentially the same under EIP-1559 as under the status quo!

When the monopoly quantity is more than the target block size, a 100% miner faces the non-trivial optimization problem of optimally trading off the short-term revenue gain from including more than 12.5M gas in a block and the long-term revenue decrease due to higher future base fees. The optimal solution to this problem depends in an intricate way on the assumed demand curve $D(\cdot)$; a detailed discussion of it is outside the scope of this report. Qualitatively, we can view this optimal strategy as an optimized version of the quantity-setting strategy with quantity 12.5M gas, in which the variable block size of EIP-1559 is exploited to mix underfull and overfull blocks so as to boost net revenue (even after accounting for the nonzero fee burn).

Remark 7.4 (51% Miner = 100% Miner) A miner or perfectly coordinated cartel of miners controlling 51% of the overall hashrate can control 100% of the blocks on the longest chain by refusing to extend any block mined by a miner outside the cartel. Thus, the optimal 100% miner strategies identified in this and the previous section are equally well available to a 51% miner[37],[38]

[37] Or at least, to a 51% miner unconcerned with detectable coordinated strategies; see Section 7.4.4 for further discussion.

[38] Even a medium-size miner or mining pool—over 20% of the total hashrate, say, as with the two biggest Ethereum mining pools [6]—could conceivably benefit from a monopoly price-setting or quantity-setting strategy, if enough users are willing to pay a premium to avoid a 20% chance of a transaction being delayed by one block.

7.4 First-Price Auctions: Do Miners Collude?

We offer no prediction on whether miners would collude under EIP-1559, for example to implement some form of the 100% miner optimal strategy identified in Section 7.3. We can, however, speculate in a principled way via analogy with observed miner behavior under the status quo.

7.4.1 Types of Miner Coordination

Miners can coordinate their actions in a number of ways. Next we single out three factors that may influence the likelihood of a cartel of miners carrying out a particular coordinated strategy.

Classifying Coordinated Strategies

1. Is the coordinated strategy plausibly for the good of the entire Ethereum network, or clearly for the good of the miners?
2. Is the coordinated strategy easily detectable?
3. Is the cartel of miners game-theoretically robust (with an incentive for cartel members to remain) or game-theoretically fragile (with an incentive for members to secede, for example by switching to solo mining or joining a competing mining pool)?

7.4.2 Coordination for the Greater Good

Ethereum miners do appear to coordinate their actions at times, for example when resolving hard forks or increasing the maximum block size (which in Ethereum is voted on by miners). In these cases, the goal is plausibly to maximize the health of the Ethereum network. For example, increases in the maximum block size over time may have been a balancing act between minimizing transaction fees and minimizing the centralization risk due to the computation and communication necessary to process blocks.[39]

Miners appear to coordinate when the goal is plausibly to maximize the health of the Ethereum network.

7.4.3 The Risk of Undetectable Coordination

The key question is then whether miners will use this coordination ability to pursue goals that are primarily in their own interest, rather than in the interest of the network. The risk is greatest from undetectable strategies.

Miners should not be expected to eschew undetectable coordinated strategies that are in their own self-interest.

Protocols should therefore be designed to avoid such undetectable strategies whenever possible, or at least to render them game-theoretically fragile (see Section 7.4.5).

[39] A plausible alternative narrative is that miners are maximizing their rewards from transaction fees subject to acceptance of the maximum block size by the network.

Example 7.5 (Fake Transactions in Vickrey Auctions Are Undetectable) Example 5.16 notes that Vickrey (a.k.a. second-price) auctions can be manipulated via fake transactions to boost a miner's revenue. Such manipulation could be difficult to detect, and a miner (myopic or otherwise) would have a strong incentive to do it. This reinforces the argument against Vickrey auctions in permissionless blockchains.

7.4.4 The Lack of Detectable Coordinated Strategies in the Wild

What about detectable coordinated strategies that favor the miners over the network? For example:

Three Types of Detectable Attacks by a 51% Cartel

1. A double-spend attack via a significant blockchain reorganization.
2. A censorship attack in which every block referencing a blacklisted address is deliberately orphaned by the cartel.
3. A denial-of-service attack in which every non-empty block is deliberately orphaned by the cartel.

All three of these attacks have been rare to non-existent in Ethereum.[40] Why? We can only speculate on the reasons:

Possible Reasons Miners Avoid Detectable Attacks

1. Enough miners (or mining pools) are fundamentally opposed to deliberate attacks that might harm the Ethereum network, due to altruism or blind loyalty, that a 51% cartel cannot form.
2. Many miners are ETH holders and believe that a detectable attack would significantly decrease the price of ETH.[41]
3. Many miners have some other form of vested interest in the health of the Ethereum network and believe it would be significantly damaged by a detectable attack. For example, an ASIC is effectively a call option on ETH [28, 57].
4. Miners fear that they would be punished for a significant detectable attack through a hard fork.[42]

7.4.5 Game-Theoretic Fragility

Perhaps miners will carry out a self-interested coordinated strategy if and only if it is undetectable? The monopoly price- and quantity-setting strategies identified in Section 7.2 indicate that reality

[40] Less secure blockchains, including Ethereum Classic, have suffered from such attacks [40].

[41] Though three recent double-spend attacks on the Ethereum Classic blockchain have not significantly harmed the price of ETC; perhaps that blockchain's relatively low level of security has been priced in all along. See also Moroz et al. [46] for further discussion.

[42] This fear is perhaps more relevant for ASIC miners than for GPU miners.

is more complex. Appropriately disguised versions of these strategies can be difficult to detect (Remark 7.3), and yet there is little to no anecdotal evidence suggesting that Ethereum miners have ever coordinated to implement such strategies. Again, there are many possible explanations:

Possible Reasons for the Lack of Price- and Quantity-Setting Strategies

1. Miners would implement such strategies if they could, but there are too many obstacles (e.g., rapidly changing and hard-to-predict demand) to coordinating on a price or quantity for a significant length of time.
2. Miners would implement such strategies if it was in their self-interest, but typically the monopoly quantity for the current demand curve (Definition 7.1) equals the maximum block size and no deviation from the protocol's prescribed behavior is necessary.
3. Disguising such strategies is too difficult, so all the arguments against detectable strategies apply.
4. Implementing such strategies could hurt the throughput and therefore health of the Ethereum network, which many miners have a vested interest in.
5. A large cartel of miners would be game-theoretically fragile, with cartel members incentivized to secede.

To illustrate the last point, we proceed as in Houy [34]. Imagine that all miners belong to the cartel and implement an optimal 100% miner strategy, such as price-setting at a monopoly price p^* that is larger than the market-clearing price $\bar{p}$. Thus, all transactions with bid at least p^* get included in a block while all the other transactions languish in the mempool; because $p^* > \bar{p}$, blocks will not be full.

The optimal myopic miner strategy (Section 5.3), meanwhile, would be to ignore the cutoff p^* and pack the current block as full as possible with the transactions with the highest bids. This strategy maximizes the miner's short-term revenue rather than leaving money on the table to sustain the upward price pressure on the creators of high-value transactions. In effect, such a myopic miner would be free riding on the sacrifices of the other miners which prop up the bids of high-value transactions. A small miner is well approximated by a myopic miner, so one might well expect such a miner to secede from the cartel to implement the optimal strategy for a myopic miner rather than for a 100% miner.

A cartel of miners could discourage secession by its members through the threat of punishment. For example, if the rest of the cartel controls at least 51% of the overall hashrate, the remaining miners could refuse to extend any block that does not conform to its rules, such as blocks that are packed with more than the monopoly quantity worth of gas.[43] However, there is little evidence of any Ethereum miners employing such punishment strategies. Why not? Perhaps they have not been needed. Perhaps carrying them out would be too logistically complex. Or perhaps punishment strategies are inevitably detectable and therefore run into all the impediments listed in Section 7.4.4 for detectable coordinated strategies.

[43] The "feather forking" variant of this strategy has the potential to succeed even with less than 50% of the overall hashrate [41].

We hypothesize that coordinated strategies that harm the Ethereum network and require a credible threat of punishment to sustain might pose little risk.

Hypothesis: Game-Theoretic Fragility Is a Dealbreaker

Coordinated miner strategies that:

(i) favor miners at the expense of the network;

(ii) require short-term sacrifices from each member for the good of the cartel; and

(iii) are costly or difficult to sustain through punishment

may be rare in a well-secured blockchain such as Ethereum.

7.4.6 The Upshot

The discussion in this section clarifies the most worrisome type of miner coordination, which deserves special attention when designing or modifying a protocol:

The most concerning type of coordinated miner strategy is one that:

1. is in the interest of miners, rather than the network;
2. is undetectable; and
3. is game-theoretically robust, with cartel members incentivized to remain.

Revisiting the four types of coordinated strategies discussed in this section—double-spend attacks, censorship attacks, denial-of-service attacks, and monopoly price- or quantity-setting—we see that the first three attacks fail the second criterion while the fourth strategy fails the third.

7.5 EIP-1559: Will Miners Collude?

We now speculate on the likelihood of different forms of miner coordination (Section 7.4) in a post-EIP-1559 world.

First, all of the arguments in Section 7.4.4 against detectable attacks remain equally valid under EIP-1559, and the specific attacks discussed (double-spend, censorship, denial-of-service) remain equally detectable.

Second, while the optimal 100% miner strategy is generally different under EIP-1559 (Section 7.3) than under the status quo (Section 7.2) due to base fee manipulation, four of the five potential impediments to implementing the latter (identified in Section 7.4.5) apply also to the former.[44] In particular, a cartel simulating the optimal 100% miner strategy under EIP-1559 is game-theoretically fragile. Analogous to a first-price auction, a myopic miner is incentivized to pack its block as full as possible with the transactions with the highest tips (up to 25M gas, which is double the target block size). This strategy maximizes short-term miner revenue rather than leaving money on the table in order to keep future base fees low. In effect, such a myopic miner

[44] The exception is the second point. Under EIP-1559, the 100% miner strategy is generally different from the honest mining strategy even when the monopoly quantity exceeds the target block size (see Section 7.3).

would be free riding on the sacrifices of the other miners that respect the target block size and thereby keep the base fee low.

Third, because of the 1559 mechanism's fee burn, fake transactions can no longer be costlessly used to disguise an attack that simulates the optimal 100% miner strategy (cf., Remark 7.3 and first-price auctions). Because this strategy is now either costly or detectable, it is arguably even less likely to be used under EIP-1559 than under the status quo.

The game-theoretic impediments to double-spend attacks, censorship attacks, denial-of-service attacks, and revenue-maximizing 100% miner strategies (including base fee manipulation) appear as strong under EIP-1559 as under the status quo.[45]

7.6 Caveats

Sections 7.4–7.5 show that, with both first-price auctions and the 1559 mechanism, all of the most concerning forms of miner collusion (double-spending, censorship, denial-of-service, revenue-maximization) are detectable, game-theoretically fragile, or both. But is this really enough evidence to conclude that the harmful effects of miner collusion will be no worse under EIP-1559 than they are now? This section plays devil's advocate and suggests some possible complications.

Because of the burned base fee revenues, many miners appear to view EIP-1559 as taking away some of their profits and handing them over to ETH holders.[46] For example, of the nine miners responding to a questionnaire by Beiko [12], six wrote that "they would not implement it under any circumstances." This strong negative reaction suggests that EIP-1559 may galvanize miners to sustain collusion to a degree not yet seen under the status quo.

An immediate issue is miner adoption, and the plan for the deployment and acceptance of EIP-1559 should be explicitly discussed. For example, can the Ethereum Foundation effectively dictate its use? Or is the plan to first secure support from major projects built on top of Ethereum (e.g., the USDC stable coin), thereby forcing miners' hands? Or should further support from miners be sought out directly, and perhaps explicitly incentivized?

A second concern is that the 1559 mechanism's fee burn could change the norms around what types of miner collusion are culturally acceptable (e.g., coordinating on a new maximum block size) versus unacceptable (e.g., a censorship attack). For example, imagine that miners coordinated their actions to avoid the base fee but otherwise acted as in a first-price auction with a maximum

[45] One counterpoint is that the reduction of miner revenue on account of the fee burn would likely reduce the overall hashrate, lowering the cost to a saboteur of launching these attacks (e.g., by renting sufficient hashrate [7, 13]). Under EIP-1559, the block reward alone must be sufficiently high to incentivize an adequate amount of hashrate and consequent security.

[46] There is merit to this argument but also some counterbalancing factors. First, because Ethereum mining has a relatively low barrier to entry and exit, a decrease in aggregate miner rewards should lead to a decrease in the overall hashrate, with the least profitable miners exiting (see e.g. [25, 35]). This, in turn, increases the relative hashrate (and corresponding fraction of miners' rewards) of the miners who remain. (Ethereum security suffers as a result but remains propped up by the block reward [10, 14], which EIP-1559 leaves untouched.)

Second, there is evidence that an increasing share of Ethereum transaction fees are paid by transactions vying for special treatment within a block (e.g., being placed first so as to execute prior to all other transactions in the same block) [33]. Provided the willingness to pay of such transactions is significantly higher than the market-clearing price at the target block size of 12.5M gas—the quantity that the base fee proxies for—miners should continue to collect significant fees from them through their tips in the 1559 mechanism.

block size of 12.5M gas.[47] Like the optimal 100% miner strategy in Section 7.5, such coordination is game-theoretically fragile and requires each miner to leave immediate revenue on the table that could otherwise be collected by including more than 12.5M gas worth of transactions in a block. On the other hand, this coordinated strategy could be much less damaging to the Ethereum network than something like a censorship attack or throttling the transaction rate to boost miner revenues. If such a strategy was widely perceived as mostly harmless—perhaps unlikely in this instance, given the Ethereum community's general enthusiasm for counteracting inflation with a fee burn—it could conceivably find more purchase among miners.

8 Alternative Designs

Does EIP-1559 need to work the way that it does? Are there alternative designs that accomplish the same goals in a better or simpler way?

Sections 8.1 and 8.2 argue that the seemingly orthogonal goals of easy fee estimation and fee burning are in fact inextricably linked through the threat of off-chain agreements. Section 8.3 investigates a design that pays revenue from transaction fees forward to miners of future blocks, an alternative to fee burning with similar game-theoretic properties. Section 8.4 recaps a recent transaction fee mechanism proposal by Basu et al. [11]. Section 8.5 discusses an alternative design that, relative to the 1559 mechanism, favors UIC over OCA-proofness. Section 8.6 explores the possibilities for alternative base fee update rules.

8.1 Paying the Base Fee to the Miner

The 1559 mechanism achieves a "good user experience," in the form of a typically-UIC guarantee (Theorem 6.8). At first glance, the proof of this guarantee seems to hinge on two assumptions: (i) the base fee is determined only by past history and independent of the current block; and (ii) the base fee is high enough that the demand for gas is at most the maximum block size. Where does fee burning come in?

Specifically, consider the following alternative design in which base fee revenues are passed on to the miner of the block; we call this the *1559-R mechanism.* (Here "R" stands for "refund.") The allocation rule $\mathbf{x}^R$ is identical to that of the 1559 mechanism (the rule $\mathbf{x}^*$ in Definition 6.1). Miners can no longer be counted on to exclude transactions with bid less than the base fee, so assume that the protocol automatically treats as invalid any transaction t in a block with a bid b_t that is less than that block's base fee r. The new payment rule $\mathbf{p}^R$ is identical to the payment rule $\mathbf{p}^f$ of a first-price auction (Example 5.6), with the miner collecting the entire bid (i.e., the minimum of the fee cap and the sum of the base fee and tip) as revenue. The new burning rule $\mathbf{q}^R$ is also the same as in a first-price auction—the all-zero rule $\mathbf{q}^f$.

Unfortunately, with a simple off-chain agreement, the 1559-R mechanism devolves into a first-price auction (with block size 25M gas, double the target):[48]

1. Users bid r on-chain and communicate off-chain what they would have bid in a standard first-price auction.

[47] Such coordination can be implemented with a variation of the strategy in Section 7.5: first drive the base fee to zero, for example by publishing a sequence of empty blocks, and then use the quantity-setting strategy with quantity 12.5M gas for all future blocks (thereby keeping the base fee at zero forevermore).

[48] The base fee of the 1559-R mechanism would therefore eventually increase to its maximum level.

2. If a miner includes a transaction t with off-chain bid b_t, t's creator transfers $b_t - r$ gwei per unit of gas to the miner. (When $b_t < r$, this should be interpreted as a refund of $r - b_t$ per unit of gas from the miner to t's creator.)

In the notation of Definition 5.23, this is the OCA $(\mathbf{b}, \boldsymbol{\tau})$ in which $\mathbf{b} = \mathbf{r}$ and $\tau_t = (b'_t - r)$, where b'_t denotes what t's creator would have bid in a standard first-price auction.

Proposition 8.1 (The 1559-R Mechanism Is Equivalent to a First-Price Auction) *For every set of transactions and base fee, there is a one-to-one correspondence between the outcomes possible in a first-price auction and the outcomes possible in the 1559-R mechanism with an OCA (with the same maximum block size).*

Proof: As noted above, the outcome of the bids $\mathbf{b}'$ in a first-price auction is equivalent to the outcome of the on-chain bids $\mathbf{r}$ under the 1559-R mechanism with off-chain transfers $\mathbf{b}' - \mathbf{r}$. In the other direction, for an outcome of the 1559-R mechanism and an OCA in which the net payment from the creator of an included transaction t to the miner is a_t, the outcome of a first-price auction in which the (on-chain) bids $\mathbf{b}'$ are the same as $\mathbf{a}$ (for included transactions) or 0 (for excluded transactions) is equivalent. ■

> Transferring the revenue from the base fee of a block to the miner of that block is economically equivalent to having no base fee. In this sense, a base fee provides UX improvements only if it is burned (or otherwise withheld from the miner).

Remark 8.2 (Partial Refund of the Base Fee) Proposition 8.1 shows that, because of the possibility of off-chain agreements, burning 0% of the base fee is economically equivalent to having no base fee at all. More generally, burning an α fraction of the base fee is economically equivalent to having a fully burned base fee that is α times as large. For example, consider a scenario in which the 1559 mechanism's base fee would stabilize at r^*. If instead half of the base fee was burned, one would expect the new mechanism to stabilize at a base fee of $2r^*$, with r^* of it burned and the rest divvied up between the miner and users via an OCA.[49]

Remark 8.3 (Implementing the Monopoly Price Under the 1559-R Mechanism) A second (if less important) issue with the 1559-R mechanism is that, unlike with the 1559 mechanism, there are scenarios in which a coordinated miner strategy meets all three of the criteria in Section 7.4.6—favoring the miners at the expense of the network, undetectable, and game-theoretically robust.

In more detail, suppose the demand curve $D(\cdot)$ is the same for every block and, at the monopoly price p^*, the demand (i.e., monopoly quantity) $D(p^*)$ is less than the target block size of 12.5M gas. Suppose also that the marginal cost μ of gas to a miner is negligible. Miners can now simulate the revenue-maximizing p^*-price-setting strategy (Section 7.2) simply by keeping the base fee at r^* at all times: (i) increase the base fee to r^*, using fake transactions as necessary; (ii) keep the base fee at r^* forevermore by publishing 12.5M gas blocks, each including $D(p^*)$ total gas of real transactions and the balance in fake transactions. This strategy is game-theoretically robust because, given that

[49] The designers of the NEAR blockchain, possibly unaware of this point, recently deployed a version of the 1559 mechanism in which 70% of the base fee is burned and the remaining 30% is transferred to smart contracts that were used in the previous epoch [5]. See Hasu [29] for further discussion.

past blocks have resulted in a base fee of p^*, a myopic miner maximizes its immediate revenue by including all eligible transactions (i.e., those with bid at least p^*) and is not harmed by the inclusion of additional fake transactions.

8.2 Fee-Burning First-Price Auctions

Alternatively, suppose we wanted a transaction fee mechanism with a fee burn but didn't care about easy fee estimation. Why not stick with first-price auctions, but burn all (or part) of the fees? Formally, this is the TFM $(\mathbf{x}, \mathbf{p}, \mathbf{q})$ with $\mathbf{x} = \mathbf{x}^f, \mathbf{p} = \mathbf{q}^f$, and $\mathbf{q} = \mathbf{p}^f$.

The problem is again the threat of off-chain agreements. Intuitively, first-price auctions in which all payments are burned are not OCA-proof because miners and users would be incentivized to move all their payments off-chain. The next proposition formally shows that this mechanism fails to satisfy Definition 5.26.

Proposition 8.4 (Fee-Burning First-Price Auctions Are Not OCA-Proof) *The fee-burning first-price auction $(\mathbf{x}^f, \mathbf{q}^f, \mathbf{p}^f)$ is not OCA-proof.*

Proof: Consider a non-empty set U of transactions with $v_t > 0$ for every $t \in U$, and assume that the marginal cost μ of gas to a miner is negligible. Assume also that there is a unique feasible subset $T \subseteq U$ of transactions maximizing the total value

$$\sum_{t \in T} v_t \cdot g_t,$$

and denote this maximum-possible total value by $V > 0$. The miner m and users can obtain joint utility V through an OCA $(\mathbf{b}, \boldsymbol{\tau})$ between the creators of T and m in which $b_t = 0$ and (for example) $\tau_t = v_t/2$ for every $t \in T$.

The joint utility (9) of an on-chain outcome is V if and only if the included transactions are precisely T and there is zero fee burn. Every bid vector $\mathbf{b}^*$ in which $b_t^* > 0$ for at least one transaction t leads to a non-zero fee burn (on account of maximizing (4)) and hence cannot achieve joint utility V. Meanwhile, the all-zero bid vector $\mathbf{b}^* = \mathbf{0}$ leads to an arbitrary feasible set $T' \subseteq U$ of transactions, which is generally different than T. ■

Moreover, in the obvious OCA for the miner and users to employ in the proof of Proposition 8.4, the on-chain bids are zero and so there is no fee burn whatsoever!

> Burning the fees of a first-price auction moves all payments off-chain and leads to zero fee burning. In this sense, a non-trivial fee burn requires a base fee.

Remark 8.5 (Partial Fee-Burning) The same argument and conclusion apply more generally to a first-price auction in which any fixed positive fraction of the fees are burned.

8.3 Paying the Base Fee Forward

Section 8.1 shows that, for a block's base fee to be economically meaningful, revenues from it cannot be passed on to the miner of the block. Perhaps the simplest way to withhold this revenue, as in the current EIP-1559 spec [20], is to burn these revenues, effectively issuing a lump-sum refund to all ETH holders. An alternative solution, discussed explicitly in [17], is to transfer these revenues to one or more miners of *other* blocks.

8.3.1 The ℓ-Smoothed Mechanism

Concretely, consider the variant of the 1559 mechanism in which, for some window length ℓ (hard-coded into the protocol), the base fee revenues from a block are split equally among the miners of the next ℓ blocks. (The 1559 mechanism can be thought of as the special case in which $\ell = 0$.) Thus, a miner of a block receives a $1/\ell$ fraction of the sum of the base fee revenues from the previous ℓ blocks, along with all of the tips from the current block.

We can define the *ℓ-smoothed mechanism* as follows. Fix a blockchain history $B_1, B_2, \ldots, B_{k-1}$ with $k \geq \ell + 1$. Let $r_i = \alpha(B_1, B_2, \ldots, B_{i-1})$ denote the base fee of block B_i, where α is the iteration of the EIP-1559 update rule (2). Let $R_k = \beta(B_1, B_2, \ldots, B_{k-1})$ denote the paid-forward base fee revenues:

$$\beta(B_1, B_2, \ldots, B_{k-1}) := \frac{1}{\ell} \sum_{i=k-\ell}^{k-1} r_i \cdot G_i,$$

where $G_i = \sum_{t \in B_i} g_t$ denotes B_i's size in gas. The allocation, payment, and burning rules of the ℓ-smoothed mechanism are formally identical to those of the 1559 mechanism (Definition 6.1–6.3), with the understanding that the burning rule (a constant function always equal to r_k) now indicates a payment that is paid forward to future miners rather than burned. Technically, the paid-forward base fee revenues R_k should be added to a miner's utility function (Definition 5.13), but because R_k is independent of the miner's current actions, it has no effect on the optimal strategy of a myopic miner (or user). In effect, R_k serves as a fixed bonus added to the standard block reward.

8.3.2 Properties of the ℓ-Smoothed Mechanism

Because users are indifferent to how their payments are directed, and because a myopic miner cares only about its revenue from the current block, all of the game-theoretic guarantees for users and myopic miners satisfied by the 1559 mechanism (Theorem 6.4, Corollary 6.5, Theorem 6.8, and Theorem 6.12) carry over to the ℓ-smoothed mechanism (for any ℓ).

Theorem 8.6 (Guarantees for the ℓ-Smoothed Mechanism) *For every $\ell \geq 0$, the ℓ-smoothed mechanism is:*

(i) MMIC;

(ii) $(r + \mu)$-costly, where r is the current base fee and μ is the marginal cost of gas;

(iii) UIC, provided the current base fee is not excessively low for the current demand; and

(iv) OCA-proof.

Theorem 8.6 holds no matter how the base fee and pay-forward rewards are defined (i.e., for any functions α and β).

The discussion on sustained collusion by miners under EIP-1559 (Section 7) applies also to the ℓ-smoothed mechanism, with some small changes. First, for a demand curve with monopoly quantity more than 12.5M gas, a 100% miner will be better off in the ℓ-smoothed mechanism (with $\ell \geq 1$) because it will avoid the non-zero fee burn it would otherwise have paid (see Section 7.3). Second, the reasoning behind the game-theoretic fragility (Section 7.4.5) of a cartel of miners simulating a 100% miner strategy is more complicated. As before, lost tip revenue disincentives a cartel member from manipulating the base fee downward—the only manipulation of concern when base fee revenues are burned. With the base fee revenues returned to a 100% miner by the ℓ-smoothed mechanism, it's now also important that fake transactions are costly (Theorem 8.6(ii)) to disincentive manipulations of the base fee upward. Finally, with base fee revenues going to miners rather than ETH holders, the caveats in Section 7.6 become moot.

8.3.3 Pros and Cons of the ℓ-Smoothed Mechanism

A basic question, worthy of lengthy debate by the Ethereum community, is: Who should benefit from the user payments that are inevitably generated by a fully utilized blockchain? The fee burn in the 1559 mechanism explicitly favors ETH holders, while the ℓ-smoothed mechanism favors Ethereum miners. Different stakeholders in Ethereum will of course have their own reasons for preferring one over the other.

A second trade-off between the 1559 and ℓ-smoothed mechanisms concerns whether variability in demand (and hence fees) translates to variability in security or in the issuance of new currency. In the 1559 mechanism, every block changes the money supply in two ways: minting new coins for the block reward (currently 2 ETH), and burning the coins used to pay the base fee. Because the base fee rises and falls with demand, Ethereum's inflation rate would be variable and unpredictable. On the other hand, assuming negligible tips, every block confers roughly the same total reward to the miner (the block reward); the security of the Ethereum network scales with this total reward [10, 14] and should therefore also stay relatively constant (modulo fluctuations in the price of ETH). Meanwhile, in the ℓ-smoothed mechanism, inflation would be as predictable as it is under the status quo (currently around 4% annually). Instead, total miner reward would vary with the revenue generated by the base fee, leading to an unpredictable level of security (though never less than that with the 1559 mechanism).

Finally, because of its variable total reward, the ℓ-smoothed mechanism is vulnerable to certain attack vectors that would be fruitless under the 1559 mechanism, especially when ℓ is small. For example, imagine that $\ell = 1$ and a miner m_1 mines a block B_1 with an unusually large sum R of transaction fees. This windfall would be reaped by the miner m_2 of the next block B_2; suppose further that the sum of transaction fees in B_2 is much less than R. At this juncture, a miner m_3 might consider trying to extend B_1 with a block B_3 in order to orphan B_2; if other miners happen to extend B_3 rather than B_2, m_3 will effectively have stolen the reward of R from m_2.[50] Such examples suggest choosing a large value of ℓ (e.g., $\ell = 1000$) to guarantee that consecutive blocks will have nearly identical total rewards associated with them.

Remark 8.7 (A Blended Mechanism) The 1559 and ℓ-smoothed mechanisms can be easily blended to balance the competing concerns of miners and ETH holders and the variability in issuance and security. For example, for a parameter $\lambda \in [0, 1]$, a mechanism could burn a λ fraction of the base fee revenues and pay forward the remaining $1 - \lambda$ fraction. Theorem 8.6 and the subsequent discussion on miner collusion remain valid for such blended mechanisms.

8.4 The BEOS Mechanism

A variant of the "pay-it-forward" design philosophy in Section 8.3 was proposed also by Basu et al. [11] for a transaction fee mechanism that is not directly related to the 1559 mechanism. We next

[50] This is similar to the undercutting attack of [21] for a regime in which transaction fees dominate block rewards; see Section 9.1 for further discussion.

explain a slightly simplified version of their proposal, which we call the *BEOS mechanism* (after its proposers).

There is a fixed block size, say 12.5M gas, and no base fee. The first key idea is to charge all transactions included in a block a common price (per unit of gas), namely the lowest bid of an included transaction. Miner revenue is then the block size (in gas) times the lowest bid of an included transaction, and so a revenue-maximizing miner may exclude transactions in order to boost the lowest included bid.[51] For example, for a block with room for three transactions and a mempool containing three transactions with bids 10, 8, and 3, a revenue-maximizing miner would include the first two transactions while excluding the third (to earn revenue $2 \times 8 = 16$). (Cf., Example 5.16.)

The second key idea is to automatically charge only a minimum transaction fee—for example, just enough to cover the marginal cost μ of gas—to all transactions in any block that is not (almost) full. This rule by itself is toothless and leads to an equivalent mechanism, as a miner can costlessly extend its favorite underfull block with minimum bid b to a full block with minimum bid b using fake transactions (all with bid b).

The final key idea in the BEOS mechanism is to pay transaction fees forward, with the transaction fee revenue from a block B split evenly between B's miner and the miners of the $\ell - 1$ subsequent blocks. Thus, the miner of a block gets a $1/\ell$ fraction of the transaction fee revenue in that block, along with a $1/\ell$ fraction of the combined revenue of the preceding $\ell - 1$ blocks. As a result, for $\ell \geq 2$, fake transactions now carry a cost: the miner pays their full transaction fees but recoups only a $1/\ell$ fraction of them as revenue.

The BEOS mechanism is arguably simpler than that proposed in EIP-1559, as there is no base fee to keep track of. Its game-theoretic guarantees are considerably weaker, however. While the "pay it forward" idea helps discourage fake transactions, the BEOS mechanism is not in general MMIC.[52] It is "approximately UIC" as the number of users grows large, in the sense that no bidding strategy generates significantly more utility than truthful bidding. It is not OCA-proof (for $\ell \geq 2$), for the same reasons that a first-price auction with fee burning is not OCA-proof (Proposition 8.4). Thus, from a game-theoretic perspective, the BEOS mechanism does not appear competitive with the 1559 mechanism.

8.5 The Tipless Mechanism: Trading Off UIC and OCA-Proofness

The 1559 mechanism uses tips to achieve OCA-proofness in all blocks (Theorem 6.12), at the expense of losing the UIC condition in blocks with excessively low base fees (see Section 6.3.2). This section presents an alternative design with the opposite trade-off—one that is always UIC, and OCA-proof except in blocks with an excessively low base fee.

[51] This is exactly the "monopolistic price" mechanism proposed by Lavi et al. [37]; they were motivated by the problem of maximizing the security provided by transaction fees (at the expense of economic efficiency) in a future in which Bitcoin's block rewards are negligible. This mechanism is MMIC; is "approximately UIC," in the sense that truthful bidding is an approximately dominant strategy for users as the number of users grows large [37, 58]; and is not OCA-proof (on account of failing to maximize the joint utility of the miner and users).

[52] Basu et al. [11] prove that the mechanism becomes "approximately MMIC" in the case of a very large number of transactions with i.i.d. valuations drawn from a distribution with bounded support.

8.5.1 The Tipless Mechanism

We next define the *tipless mechanism*, so-called because it is essentially the 1559 mechanism with constant and hard-coded tips rather than variable and user-specified tips. As with the 1559 mechanism, each block has a base fee $r = \alpha(B_1, B_2, \ldots, B_{k-1})$ that depends on past blocks and is burned (or alternatively, paid forward as in Section 8.3). The creator of a transaction t specifies a fee cap c_t but no tip. This parameter induces a bid b_t for the transaction with respect to any given base fee r, namely

$$b_t = \min\{r + \delta, c_t\}. \tag{20}$$

Here δ is a hard-coded tip to incentivize miners to include transactions—for example, equal to (or perhaps slightly higher than) the marginal cost μ of gas to miners.[53] The only difference between the tipless mechanism and the 1559 mechanism is the number of user-specified parameters and their interpretation as bids relative to the current base fee—that is, the types of bidding strategies available to users. The allocation, payment, and burning rules of the tipless mechanism are formally identical to those of the 1559 mechanism (Definitions 6.1–6.3). Given that all the tips are the same and cover a miner's marginal cost of gas, the allocation rule (12) boils down to packing a block as full as possible with transactions t with a bid $b_t \geq r + \delta$. The creator of an included transaction pays $r + \delta$ gwei per unit of gas, of which r is burned and δ is transferred to the miner.[54]

8.5.2 Properties of the Tipless Mechanism

The proof that the 1559 mechanism is incentive compatible for myopic miners (MMIC) depends only on the form of the allocation, payment, and burning rules of the mechanism; it is agnostic to the process by which transactions' bids are set (see Theorem 6.4). Thus, the same proof applies equally well to the tipless mechanism.

Theorem 8.8 (The Tipless Mechanism is MMIC) *The tipless mechanism is MMIC.*

Now consider a block in which the base fee r is excessively low (Definition 6.7), meaning that the demand for gas at price $r + \delta$ is more than the maximum block size G. In the 1559 mechanism, the creators of transactions willing to pay at least $r + \delta$ must then compete for inclusion through their tips. As a result, analogous to a first-price auction (Example 5.22), in this case the mechanism is not incentive compatible for users (UIC)—there are no "obvious optimal parameters" to associate with a transaction.

In the tipless mechanism, such transaction creators do not have the vocabulary to differentiate themselves by offering to pay extra. As a result, the mechanism remains UIC even in blocks with an excessively low base fee.

Theorem 8.9 (The Tipless Mechanism is UIC) *The tipless mechanism is UIC.*

Proof: Fix an on-chain history $B_1, B_2, \ldots, B_{k-1}$ and corresponding base fee $r = \alpha(B_1, B_2, \ldots, B_{k-1})$, and a set T of transactions. Suppose the creator of a transaction $t \in T$ sets its fee cap equal to its

[53] More generally, the hard-coded tip δ could be adjusted over time in the same way as the block reward, through social consensus and hard forks.

[54] This variant of the 1559 mechanism has been implemented in the NEAR protocol [5]; see Hasu [29] for further discussion.

maximum willingness to pay, corresponding to the bid

$$b^*(v_t) = \min\{r + \delta, v_t\}. \tag{21}$$

Could some other bid be better? For a low-value transaction (with $v_t < r + \delta$), every alternative bid $\hat{b}_t$ either has no effect on t's utility or leads to t's inclusion in the block; the latter only occurs when $\hat{b}_t \geq r + \delta$, in which case the creator's utility drops from 0 to $(v_t - \hat{b}_t) \cdot g_t < 0$. For a high-value transaction (with $v_t \geq r + \delta$), every alternative bid $\hat{b}_t$ either has no effect on the creator's utility or, if the alternative bid triggers t's exclusion, drops its utility from a nonnegative number $(v_t - r - \delta) \cdot g_t \geq 0$ to 0.[55] We conclude that the bid in (21) is always utility-maximizing for t's creator.[56] ■

Further, the tipless mechanism is OCA-proof except during periods of rapidly increasing demand.

Theorem 8.10 (The Tipless Mechanism Is Typically OCA-Proof) *Fix an on-chain history $B_1, B_2, \ldots, B_{k-1}$ and corresponding base fee $r = \alpha(B_1, B_2, \ldots, B_{k-1})$, and a set T of transactions for which r is not excessively low. With $\delta = \mu$, the tipless mechanism is OCA-proof.*

Proof: The joint utility (9) of the miner and users for the current block B_k is

$$\sum_{t \in B_k} (v_t - r - \mu) \cdot g_t. \tag{22}$$

Because r is not excessively low for T, the total gas consumed by transactions t with $v_t \geq r + \mu$ is at most the maximum block size G. The joint utility (22) is therefore maximized by including precisely these transactions. This outcome can be achieved on-chain (with $b_t = \min\{r + \mu, v_t\}$ for each $t \in T$), and thus cannot be improved upon by an OCA. ■

Remark 8.11 (The Tipless Mechanism Is Not Always OCA-Proof) The tipless mechanism is not generally OCA-proof when the base fee r is excessively low (even with $\delta = \mu$). In this case, a miner is instructed by the allocation rule to pack its block as full as possible using transactions with bid at least $r + \mu$. With an excessively low base fee, the feasible subset of such transactions that maximizes the block size $\sum_{t \in T} g_t$ is generally different from the feasible subset that maximizes the joint utility $\sum_{t \in T}(v_t - r - \mu) \cdot g_t$.[57] The miner and users can then strictly increase their joint utility with an OCA that instead includes the latter subset of transactions (for example, with transfers arranged to share the increase in joint utility equally among the miner and users).

[55] If r is an excessively low base fee and the demand at price $r + \delta$ is more than the maximum block size, the miner maximizes its revenue by packing its block as full as possible (as the tip-per-unit-gas δ is the same for every transaction). That is, the miner includes the feasible set of transactions that maximizes the total gas used. We assume that, if there is a tie between two or more such feasible sets, the miner breaks the tie in a consistent way, independent of transactions' fee caps.

[56] Moreover, the bidding strategy $b^*(\cdot)$ is a symmetric dominant-strategy equilibrium in the sense of footnote 29. That is, the suggested bid $b^*(v_t)$ is utility-maximizing for t's creator no matter what the other bids are (i.e., even if the other bids differ from those suggested by the strategy $b^*(\cdot)$).

[57] For example, with $G = 2$ and $r + \mu = 1$, consider one eligible transaction with $v_t = 2$ and $g_t = 1$ and another with $v_t = 1$ and $g_t = 2$.

8.5.3 Pros and Cons of the Tipless Mechanism

Perhaps the strongest argument in favor of the tipless mechanism over the 1559 mechanism is its simplicity. On the user side, there are several simplifications. The creator of a transaction t only has to specify one parameter (a fee cap c_t) rather than two (a fee cap c_t and a tip δ_t). The "obvious optimal bid" in the tipless mechanism (setting $c_t = v_t$) is optimal for every block and no matter what the bids of the competing transactions. The "obvious optimal bid" in the 1559 mechanism (setting $c_t = v_t$ and $\delta_t = \mu$) is optimal only in blocks without an excessively low base fee, and only after assuming that other transactions' bids were set in the same way. On the miner side, the revenue-maximizing strategy simplifies to maximizing the block size while using only transactions with a bid that is at least $r + \mu$ (where r denotes the current base fee). Relatedly, miners have no levers by which to pressure users to increase their tips (cf., footnote 38).

What about the mechanism's drawbacks? First, the hard-coded tip δ is yet another somewhat arbitrary parameter than may need to be adjusted over time through network upgrades.[58] Second, when there are blocks with excessively low base fees (due to rapidly increasing demand), OCA-proofness breaks down. At such times, one might expect miners and users to simulate the on-chain tips of the 1559 mechanism with an off-chain agreement. Even with a base fee that is not excessively low, such agreements might be used to accommodate transaction creators angling for a specific block position (as opposed to mere inclusion).[59]

8.6 Alternative Base Fee Update Rules

The game-theoretic guarantees for the 1559 mechanism (Sections 6–7) and inseparability of easy fee estimation and fee withholding (Section 8.1–8.3) argue strongly for a history-dependent base fee, the revenues from which are burned or otherwise withheld from a block's miner. Accordingly, in this section we consider only designs with such a base fee.

But how, exactly, should the base fee be computed from the blockchain's history? The MMIC (Theorem 6.4), typically-UIC (Theorem 6.8), and OCA-proof (Theorem 6.12) guarantees from Section 6 hold no matter how the base fee is set. The impediments to miner collusion identified in Section 7 likewise give little guidance as to how the base fee should evolve over time. The goal of this section is to clarify the assumptions baked into the update rule in the current EIP-1559 spec (2) and identify a few axes along which to experiment.

8.6.1 Assessing Update Rules

The ideal base fee for a block is the market-clearing price for the current mempool and block size (see Section 3.1). An ideal base fee update rule would magically guess this price, immediately adjusting to sudden changes in demand. A good base fee update rule should reasonably approximate this magical one, without introducing any undue incentives for base fee manipulation by miners and users or vulnerabilities to outside attacks.

Desiderata for a Base Fee Update Rule

1. Adjusts upward reasonably quickly after a sudden spike in demand.

[58] Possible counterargument: with so many such parameters already (e.g., opcode gas costs [56]), what's one more?

[59] Depending on how miners choose to break ties among eligible transactions for inclusion in such a block, on-chain shenanigans may also be possible (e.g. [38]).

2. Adjusts downward reasonably quickly after a sudden drop in demand.
3. Adjusts slowly enough to avoid overreacting to small or very short-lived changes in demand.
4. Cannot be manipulated by a cartel of users and/or miners in a game-theoretically robust way (cf., Section 7.4.5).
5. Expensive for an attacker to exploit.

How quickly is "reasonably quickly"? How expensive is "expensive"? Such questions are outside the scope of this report and best answered through experimentation and community discussion. The rest of this section assesses the update rule in the EIP-1559 spec according to these criteria and suggests some alternatives to explore.

8.6.2 Decomposable Update Rules

In principle, the base fee r for a block B_k can be an arbitrary function α of the blockchain history $B_1, B_2, \ldots, B_{k-1}$:

$$r = \alpha(B_1, B_2, \ldots, B_{k-1}).$$

In practice, however, the base fee should not be overly burdensome to compute. This point motivates the next definition.

Definition 8.12 (Decomposable Update Rule) An update rule α is *decomposable* if it can be written

$$\alpha(B_1, B_2, \ldots, B_{k-1}) = \zeta(B_{k-1}) \cdot \alpha(B_1, B_2, \ldots, B_{k-2}),$$

where ζ is the *adjustment function.*

Definition 8.12 encodes two different restrictions. First, the base fee of a block should depend on only the base fee and the contents of the most recent block. Second, the adjustment function ζ depends only on the contents of the most recent block B_{k-1} and not on its base fee.

Example 8.13 (The EIP-1559 Update Rule Is Decomposable) The update rule (2) in the EIP-1559 spec is decomposable with

$$\zeta(B_{k-1}) = 1 + \frac{1}{8} \cdot \left(\frac{g(B_{k-1}) - G_{target}}{G_{target}} \right), \tag{23}$$

where $g(B) = \sum_{t \in B} g_t$ denotes the size (in gas) of block B and G_{target} a target block size (e.g., 12.5M gas).

A base fee computed by a decomposable update rule can be expressed in a compact product form.

Proposition 8.14 (Product Form for Decomposable Update Rules) *If α is a decomposable update rule with adjustment function ζ and r_0 is the base fee of the genesis block B_1, then for every blockchain history $B_1, B_2, \ldots, B_{k-1}$,*

$$\alpha(B_1, B_2, \ldots, B_{k-1}) = r_0 \cdot \prod_{i=1}^{k-1} \zeta(B_i). \tag{24}$$

Non-decomposable update rules are more complex but could potentially be useful. For a reasonably natural example, suppose we wanted to limit the lifetime over which any given block affects the base fee:

Example 8.15 (Sliding Windows Are Not Decomposable) Consider a base fee update rule that depends on only the most recent ℓ blocks, for some parameter ℓ (e.g., 100 or 1000):

$$\alpha(B_1, B_2, \ldots, B_{k-1}) = r_0 \cdot \prod_{i=k-\ell}^{k-1} \zeta(B_i),$$

where k is assumed to be at least $\ell + 1$, and r_0 and ζ denote the initial base fee and adjustment function, respectively. Because the change in base fee depends on both the block entering the sliding window (the most recent one) and the block exiting this window (from $\ell + 1$ blocks back), this update rule is not decomposable.

Remark 8.16 (Oscillatory Behavior of Decomposable Update Rules) M. Ferreira, D. Moroz, and M. Stern (personal communication, October 2020) point out that, in certain pathological scenarios, decomposable update rules can oscillate between two base fees rather than converge to a market-clearing base fee, even during a period of stable demand. For example, consider such a rule with an adjustment function ζ satisfying $\zeta(B) = \frac{3}{2}$ for maximum-size blocks B and $\zeta(B) = \frac{2}{3}$ for empty blocks B. Suppose the current base fee is r and there is a huge mempool of transactions, the fee caps of which all happen to land in the interval $[1.1r, 1.4r]$. (Assume that all tips are negligible.) What happens next?

Because all transactions are willing to pay the current base fee of r, r is an excessively low base fee and the next miner will produce a maximum-size block. As a result, the base fee will jump from r to $\frac{3}{2}r$, at which point no transactions are willing to pay the base fee! The next miner has no choice but to produce an empty block, and the base fee will return to r. This oscillation between the base fees r and $\frac{3}{2}r$, and between maximum-size and empty blocks, could in principle continue forever.[60]

Such oscillatory behavior may be unlikely in a real deployment, given the variety of tools and considerations likely to be used when specifying the bidding parameters for a transaction. If necessary, Ethereum clients could explicitly inject randomness into these parameters to avoid such pathological outcomes.

Having noted that non-decomposable update rules may be worth experimenting with, we now narrow our focus to decomposable rules.

8.6.3 What Should the Adjustment Function Depend On?

Designing a decomposable update rule boils down to designing its adjustment function ζ. By assumption, this function depends only on the contents of the most recent block B_{k-1}. In principle, the function $\zeta(B)$ could depend on B's contents in arbitrarily complex ways. In the EIP-1559 adjustment function (23), $\zeta(B)$ depends only on the total gas $g(B) = \sum_{t \in B} g_t$ used in B, and not on any finer-grained information about its transactions.

[60] With the adjustment function in the current EIP-1559 spec (23), such an oscillation will stop eventually, although possibly only after a large number of blocks (depending on how tightly concentrated transactions' bids are).

While it's easy to imagine alternative adjustment functions, care must be taken with the incentives. As a cautionary tale, consider an adjustment function ζ that tries to do away with variable-size blocks through its dependence on the bids attached to the transactions in a block B.

Example 8.17 (Incorporating Bids into the Adjustment Function) In this design, the target block size and the maximum block size are the same (e.g., 12.5M gas). If a block B has size less than the maximum, then the adjustment function satisfies $\zeta(B) < 1$ and the base fee decreases for the next block (as in EIP-1559). For a full block B, the adjustment function considers the minimum (or average, or median, or. . .) tip of a transaction in the block. If this statistic is close to 0, the base fee remains unchanged ($\zeta(B) = 1$); if it's significantly larger than 0, the base fee increases ($\zeta(B) > 1$).

The problem? When the current base fee is excessively low, there is no disincentive to the users or miners from colluding to keep it low. For suppose miners and users moved the tip market off-chain, similar to the proof of Proposition 8.4. Users and miners are indifferent to whether payments are on- or off-chain, as the fee burn and gas costs are the same either way. But now the on-chain tips are all 0 and the base fee will not increase.

Remark 8.18 (OCA-Proofness vs. Miner Collusion) The off-chain agreement in Example 8.17 does not violate OCA-proofness (users and miners are equally well off with the OCA, but not strictly better off) and hence does not contradict the aforementioned fact that the 1559 mechanism remains OCA-proof no matter how its base fee is computed. However, the OCA in Example 8.17 does show that this variant of the 1559 mechanism encourages the most concerning type of coordinated miner strategy (Section 7.4.6)—one that favors the miners at the expense of the network, is potentially undetectable, and is game-theoretically robust.

More generally, Example 8.17 illustrates how OCAs can be used to manipulate any attempt to incorporate the bids attached to a block's transactions into the adjustment function. Given these dangers, it is unsurprising that the adjustment function in EIP-1559 depends only on the gas consumed by the included transactions, and it is unclear if any additional information could be safely used.

8.6.4 The Functional Form of the Adjustment Function

Even after committing to an adjustment function that is a function solely of the block size there remains flexibility in the function's form and parameters. The choices of these in the EIP-1559 adjustment function (23) appear fairly arbitrary and are prime candidates for experimentation; we next offer some possible alternatives.

By assumption, we are now considering update rules of the form $\zeta(B) = f(g(B))$, where f is a univariate real-valued function and $g(B) = \sum_{t \in T} g_t$ denotes the size of block B. Only nondecreasing functions are sensible choices for f—big blocks suggest excess demand and that the base fee should be increased, small blocks that it should be decreased. Any continuous such function f effectively has a "target block size," meaning a gas threshold G_{target} such that $f(G_{target}) = 1$.

What functional form should f have? The current EIP-1559 spec uses an adjustment function (23) with the form

$$f(x) = 1 + h(x), \tag{25}$$

where h is an increasing linear function with $h(G_{target}) = 0$. Linear functions are attractive for their simplicity, but a nonlinear function h might well strike a better balance between the competing goals listed in Section 8.6.1.

V. Buterin (personal communication, October 2020) suggests an alternative functional form, motivated by the fact that the function $1 + x$ is well approximated by e^x when x is small (where $e = 2.718\ldots$ is Euler's number):

$$f(x) = e^{h(x)}, \tag{26}$$

where h is an increasing function equal to 0 at G_{target}. Decomposable update rules with an adjustment function of this form are especially aesthetically appealing when written in product form (24):

$$\alpha(B_1, B_2, \ldots, B_{k-1}) = r_0 \cdot \prod_{i=1}^{k-1} \zeta(B_i) = r_0 \cdot \prod_{i=1}^{k-1} e^{h(g(B_i))} = r_0 \cdot \exp\left\{\sum_{i=1}^{k-1} h(g(B_i))\right\}.$$

For example, plugging in the function $h(x) = (x - G_{target})/8G_{target}$ used in the current EIP-1559 spec:

$$\begin{aligned}\alpha(B_1, B_2, \ldots, B_{k-1}) &= r_0 \cdot \exp\left\{\frac{1}{8}\sum_{i=1}^{k-1}\left(\frac{g(B_i) - G_{target}}{G_{target}}\right)\right\} \\ &= r_0 \cdot \exp\left\{\frac{1}{8}\left(\frac{\sum_{t \in B_1 \cup \cdots \cup B_{k-1}} g_t}{G_{target}} - (k-1)\right)\right\}.\end{aligned}$$

The final expression makes clear that this base fee depends only on the amount of gas consumed to date (along with the block height k and initial base fee r_0), and not on how this gas was distributed across the past blocks. The adjustment function (23) proposed in EIP-1559 does not have this property; for example, two blocks with size equal to the target leave the base fee unchanged, while an empty block followed by a block with size double the target (or vice versa) have the cumulative effect of multiplying the base fee by $\frac{63}{64}$.

Remark 8.19 (Compromising with Taylor Approximations) Exponential functions are less convenient numerically than polynomials. The adjustment function in (25) can be viewed as a degree-1 polynomial approximation of the exponential adjustment function (26). A natural compromise is to instead use the degree-2 polynomial approximation suggested by the exponential function's Taylor series:

$$f(x) = 1 + h(x) + \frac{h(x)^2}{2}.$$

For example, plugging in the function $h(x) = (x - G_{target})/8G_{target}$ gives a novel adjustment function:

$$\zeta(B) = 1 + \frac{1}{8} \cdot \frac{g(B) - G_{target}}{G_{target}} + \frac{1}{128}\frac{(g(B) - G_{target})^2}{G_{target}^2}.$$

8.6.5 Choosing the Rate of Change

One "magic number" that jumps out from the adjustment function (23) proposed in EIP-1559 is the factor of $\frac{1}{8}$, which controls how rapidly the base fee can change from one block to the next. More generally, an important design question is the minimum and maximum values that an adjustment function $\zeta(B)$ can take on (in (23), $\frac{7}{8}$ and $\frac{9}{8}$, respectively). The goal should be to strike a balance between the desiderata listed in Section 8.6.1.

The factor of $\frac{1}{8}$ in (23) means that a sequence of maximum-size blocks (with double the target size) would double the base fee in under 1.5 minutes (assuming one new block on average every 13–15 seconds [2]) and increase it by an order of magnitude in under 5 minutes. A sequence of empty blocks would decrease the base fee at a similar, slightly faster, rate. Thus, for demand shocks that persist for tens of minutes or more, the base fee should have sufficient time to adjust. The base fee would not respond much to short-lived demand shocks, although sudden demand increases would be mitigated by the additional throughput offered by variable-size blocks (cf., Example 3.3). Overall, for balancing the first three desiderata in Section 8.6.1, the initial choices of the factors $\frac{7}{8}$ and $\frac{9}{8}$ for the minimum and maximum change in base fee seem as good as any. However, this design choice should clearly be revisited after there is more data from experiments with and deployments of the 1559 mechanism.[61]

A different principled way to derive a maximum rate of change for the base fee is to consider an attacking cartel of miners that strives to overwhelm the network with a sequence of maximum-size blocks (cf., the fifth goal in Section 8.6.1). For example, consider the adjustment function in (23) and suppose that the minimum base fee is 1 gwei and the maximum block size is 25M gas. Five minutes of such a "double-full block attack," starting from the minimum-possible base fee and assuming that all blocks during this period are mined by the cartel, would typically cost at least

$$\underbrace{25000000}_{\text{max block size (gas)}} \times \sum_{i=1}^{20} \underbrace{\left(\frac{9}{8}\right)^{i-1}}_{\substack{\text{base fee of} \\ i\text{th block (gwei)}}} \approx 1.9 \text{ ETH};$$

thirty minutes would cost roughly

$$25000000 \times \sum_{i=1}^{120} \left(\frac{9}{8}\right)^{i-1} \approx 275000 \text{ ETH},$$

or roughly 165 million USD at an exchange rate of 600 USD/ETH; and so on. Similar calculations can be used to reverse engineer an appropriate maximum rate of base fee change from a target cost for a double-full block attack of a given duration.

Remark 8.20 (Variable Block Sizes vs. Variable Rate of Block Creation) Short (e.g., five-minute) double-full block attacks appear unlikely to significantly harm the Ethereum network, provided the existing vulnerabilities to adversarially constructed blocks [49] are addressed. A sequence of n double-full blocks in a given time period imposes roughly the same load on the network as $2n$ target-size blocks during the same period. Because blocks are effectively created by a Poisson process rather than deterministically, the Ethereum network must already accommodate short periods during which the gas consumption is double its expectation.[62]

[61] For example, the factor of $\frac{1}{8}$ could be added to the list of hard-coded parameters whose values are revisited with every network upgrade, joining the block reward, opcode gas costs, and so on.

[62] And with the proof-of-stake design in ETH 2.0, the rate of block creation will be roughly deterministic; there, the new variability in block sizes under EIP-1559 will effectively be canceled out by the variability eliminated from the rate of block creation.

8.6.6 Choosing the Block Elasticity

A second "magic number" in EIP-1559 is the ratio of 2 between the maximum and target block sizes. Holding the target block size fixed, why not a larger maximum block size? Or a smaller one?

For flexibility and to absorb short and sudden demand spikes (cf., Example 3.3), a bigger maximum block size is better. The problem with a big maximum block size is the computation and bandwidth required by full nodes to process blocks, and the consequent risks of greater centralization. A ratio of 2 is one simple compromise between these two competing forces.

A ratio of 2 between the maximum and target block size is also convenient because only a 51% cartel of miners could significantly manipulate the base fee or the long-run average block size. (For example, with a 49% cartel, the non-colluding miners can negate maximum-size blocks with empty blocks and vice versa.) With a ratio of only $\frac{3}{2}$, say, one of two compromises must be struck: (i) leave the base fee adjustment function as in (23), in which case a 34% cartel could manipulate the base fee downward (as it would now take two maximum-size blocks to negate an empty block produced by the cartel); or (ii) make the adjustment function in (23) asymmetric so that empty and maximum-size blocks continue to negate each other, in which case a 34% cartel could reduce the long-run average block size to less than the target block size, thereby reducing throughput (again by producing empty blocks). Ratios bigger than 2 seem less problematic, as a 34% cartel of miners would presumably not want to manipulate a burned base fee upward.

Overall, these points suggest taking the ratio between the maximum and target block size as large as possible, subject to the network having the computational resources to process a short burst of maximum-size blocks. The "best" choice of this parameter may evolve over time, and could be added to the list of parameter choices that are revisited with each network upgrade.

9 Additional Remarks

9.1 Side Benefits of EIP-1559

This report assesses the transaction fee mechanism proposed in EIP-1559 from the perspective of easy fee estimation for Ethereum users (formalized by the "typically-UIC" guarantee of Theorem 6.8). Several byproducts of the design are of value in their own right.

First, as we observed in Section 3.2, easy fee estimation and the introduction of variable block sizes should decrease the variance in transaction fees during periods of changing demand.

Second, EIP-1559 introduces fee burning through its burned base fee. Fee burning (or otherwise withholding base fee revenues from a block's miner) is necessary for the base fee to be economically meaningful (see Section 8.1), but arguably is a "necessary good" rather than a "necessary evil." Ethereum's current rate of inflation—due to block, uncle, and nephew rewards—is roughly 4%. If transaction fees continue to be high, and a significant portion of them are burned, the inflation rate will decrease and could even turn negative.[63,64] In any case, because burned fees are effectively a lump sum refund to ETH holders, the value of ETH would be tied directly to the intensity of network usage. Additionally, burned fees must be paid on-chain and in ETH, thereby imbuing ETH with unique functionality.[65]

[63] For example, in September 2020, Ethereum miners made more money from transaction fees than from block rewards [36].

[64] The inflation rate will become less predictable, however.

[65] In contrast, mere transfers between users and miners can be moved off-chain and paid using a different asset

Third, EIP-1559's base fee can serve as a difficult-to-manipulate proxy for the current market-clearing gas price, which can in turn enable a variety of new smart contracts (e.g., gas futures markets).

Finally, there are well-documented incentive issues when transaction fees dominate block rewards, for example the incentive for a miner to launch an undercutting attack that forks a block with an unusually large amount of transaction fees [21]. By directing transaction fees away from miners and to the network, EIP-1559 decreases the importance of transaction fees to miners and makes such attacks less attractive.[66]

9.2 The Escalator: EIP-2593

EIP-2593 (a.k.a. the "escalator") is another proposal, orthogonal to EIP-1559, that strives to improve the user experience through more convenient fee estimation [27].[67] Its goal is not to change Ethereum's transaction fee mechanism (which would remain a first-price auction), but rather to make bidding easier for Ethereum users through a richer menu of bidding options. Specifically, rather than a single gas price, an Ethereum transaction would now come equipped with four bidding-related parameters:

(i) the smallest block height at which the transaction is valid, and a bid for that block;

(ii) the largest block height at which the transaction is valid, and a bid for that block.

Bids for all intermediate blocks are then determined automatically via linear interpolation. For example, a bid of 100 for block 10 and 150 for block 20 induces the bids 105, 110, . . . , 145 for blocks 11–19. One would expect an impatient user to specify a relatively short interval of blocks and a relatively high bid for the first block. A patient user, who favors a cheap price over immediate inclusion, would presumably opt for a long interval and a low initial bid.

An Ethereum user could simulate the functionality of EIP-2593 by rebroadcasting a transaction with successively higher gas prices. The goal of EIP-2593 is to automate this process in-protocol, eliminating the added computational burden of resubmitted transactions.

EIP-2593 increases the number of bidding parameters relative to the status quo—in effect, adding a rate of increase parameter to the existing gas price parameter. More parameters means more in-protocol bidding options for users, but they also potentially complicate the task of choosing a bidding strategy.[68] EIP-2593 also locks users into a single type of bidding strategy (with a linear bid increase), even though a user might be better served by a different type of strategy (e.g., a more general concave or convex function).

Remark 9.1 (Combining EIP-1559 and EIP-2593) EIP-2593 was initially proposed in part as an alternative to EIP-1559, as a way to make fee estimation easier for users without introducing any major changes to the Ethereum protocol. The two proposals are easily combined, however, by plugging in EIP-2593's linear bidding strategies to set transaction tips in EIP-1559; the base fee would evolve independently, according to the usual update rule in (2). Because EIP-1559's

(e.g., USDT).

[66] Though if the bulk of transaction fees come from a small number of transactions willing to pay much more than the base fee (e.g., submitted by front-running bots), such attacks will remain an issue.

[67] The idea behind this proposal was inspired by Miller and Drexler [42].

[68] Hasu and Konstantopoulos [31] point out that another possible drawback of supporting richer bidding strategies is a decrease in privacy, with more clues about a user's preferences publicly available on-chain.

tips are necessary primarily in blocks with an excessively low base fee (see Definition 6.7 and Theorem 6.8), EIP-2593's additional functionality may be relevant only in the occasional period of rapidly increasing demand.[69]

The scope of EIP-2593 is narrower than that of EIP-1559 and it makes much less radical changes to the status quo. The good news is that the former proposal accordingly carries less risk than the latter; the bad news is that it offers none of the side benefits listed in Section 9.1. For the specific objective of easier fee estimation, the arguments currently justifying EIP-1559 appear stronger and more rigorous than those for EIP-2593. In particular, because EIP-2593 retains the first-price transaction fee mechanism, there is no hope for "obvious optimal bids" in the sense of the "typically-UIC" guarantee for the 1559 mechanism (Theorem 6.8).

10 Conclusions

Does EIP-1559 offer an improvement over Ethereum's current transaction fee mechanism? The biggest potential benefits of the proposed changes are as advertised: easy fee estimation, in the form of an "obvious optimal bid" outside of periods of rapidly increasing demand (Theorem 6.8); lower variance in transaction fees due to increased flexibility in block size (Section 3.2); game-theoretic robustness to protocol deviations and off-chain agreements, both at the scale of a single block (Theorems 6.4 and 6.12) and of multiple blocks (Section 7); and reduced inflation due to fee burning (Section 9.1).

Most of the major risks in implementing EIP-1559 are the same as those for any major change to the Ethereum protocol: implementation errors; a fork caused by some parties rejecting the changes; extra complexity at the consensus layer; additional parameters to be tweaked with every network upgrade; and the spectre of unforeseeable downstream consequences. Additional risks specific to EIP-1559 include the possibility of a hostile reception by miners (due to lost revenue from burned transaction fees) and a coordinated response (Sections 7.5–7.6); and a new (if expensive) attack vector enabled by variable-size blocks (Sections 8.6.5–8.6.6).

Reasonable people will disagree on whether the benefits of EIP-1559 justify the risks in adopting it. Those who subscribe to a "why fix what isn't (too badly) broken" philosophy may prefer to stick with the status quo. For those who believe that consensus-layer innovation should continue to be a central part of Ethereum's future, however, the arguments in favor of EIP-1559 are strong.

References

[1] EIP-1559: Fee market change for ETH 1.0 chain. URL: `https://ethereum-magicians.org/t/eip-1559-fee-market-change-for-eth-1-0-chain/2783`, March 2019.

[2] Ethereum average block time chart. URL: `https://etherscan.io/chart/blocktime`, November 2020.

[3] Etherscan. URL: `https://etherscan.io/txs`, November 2020.

[4] Filecoin base fee variations. URL: `https://filfox.info/en/stats/gas`, November 2020.

[69] Monnot [44] explores this design through simulations, with inconclusive results.

[5] The NEAR white paper. URL: https://near.org/papers/the-official-near-white-paper/, 2020.

[6] PoolWatch.io. URL: https://www.poolwatch.io/coin/ethereum, November 2020.

[7] PoW 51% attack cost. URL: https://www.crypto51.app/, November 2020.

[8] Aijan. EIP-1599 is a critical mistake at all. URL: https://medium.com/@hongji/eip-1599-is-a-critical-mistake-at-all-433f4416f881, September 2020.

[9] M. Akbarpour and S. Li. Credible auctions: A trilemma. *Econometrica*, 88(2):425–467, 2020. URL: http://web.stanford.edu/~mohamwad/CredibleMechanisms.pdf.

[10] R. Auer. Beyond the doomsday economics of "proof-ofwork" in cryptocurrencies. BIS working paper #765. URL: https://www.bis.org/publ/work765.pdf, January 2019.

[11] S. Basu, D. Easley, M. O'Hara, and E. G. Sirer. Towards a functional fee market for cryptocurrencies. arXiv:1901.06830. URL: https://arxiv.org/pdf/1901.06830.pdf, January 2019.

[12] T. Beiko. EIP-1559 community outreach report. URL: https://medium.com/ethereum-cat-herders/eip-1559-community-outreach-report-aa18be0666b5, October 2020.

[13] J. Bonneau. Why buy when you can rent?: Bribery attacks on Bitcoin-style consensus. In *Proceedings of the Third Workshop on Bitcoin and Blockchain Research*, pages 19–26, Feburary 2016. URL: https://jbonneau.com/doc/B16a-BITCOIN-why_buy_when_you_can_rent.pdf.

[14] E. Budish. The economic limits of Bitcoin and the blockchain. NBER Working Paper 24717. URL: https://faculty.chicagobooth.edu/eric.budish/research/Economic-Limits-Bitcoin-Blockchain.pdf, 2018.

[15] V. Buterin. Ethereum: A next-generation smart contract and decentralized application platform. Unpublished white paper. URL: https://ethereum.org/en/whitepaper/, November 2013.

[16] V. Buterin. On transaction fees, and the fallacy of market-based solutions. URL: https://blog.ethereum.org/2014/02/01/on-transaction-fees-and-the-fallacy-of-market-based-solutions/, February 2014.

[17] V. Buterin. On inflation, transaction fees and cryptocurrency monetary policy. URL: https://blog.ethereum.org/2016/07/27/inflation-transaction-fees-cryptocurrency-monetary-policy/, July 2016.

[18] V. Buterin. Blockchain resource pricing. URL: https://ethresear.ch/uploads/default/original/2X/1/197884012ada193318b67c4b777441e4a1830f49.pdf, August 2018.

[19] V. Buterin. First and second-price auctions and improved transaction-fee markets. URL: https://ethresear.ch/t/first-and-second-price-auctions-and-improved-transaction-fee-markets/2410, July 2018.

[20] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norden, and A. Bakhta. EIP-1559 specification. URL: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1559.md.

[21] M. Carlsten, H. A. Kalodner, S. M. Weinberg, and A. Narayanan. On the instability of Bitcoin without the block reward. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 154–167, 2016. URL: `https://www.cs.princeton.edu/~arvindn/publications/mining_CCS.pdf`.

[22] V. M. Coppinger, V. L. Smith, and J. A. Titus. Incentives and behavior in English, Dutch, and sealed-bid auctions. *Economic Inquiry*, 18(1):1–22, 1980.

[23] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels. Flash Boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *Proceedings of the 41st IEEE Symposium on Security and Privacy*, pages 910–927, 2020. URL: `https://arxiv.org/pdf/1904.05234.pdf`.

[24] C. Decker and R. Wattenhofer. Information propagation in the Bitcoin network. In *13th IEEE International Conference on Peer-to-Peer Computing*, 2013. URL: `https://tik-db.ee.ethz.ch/file/49318d3f56c1d525aabf7fda78b23fc0/P2P2013_041.pdf`.

[25] Easley, M. O'Hara, and S. Basu. From mining to markets: The evolution of bitcoin transaction fees. *Journal of Financial Economics*, 134(1):91–109, October 2019. URL: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3055380`.

[26] M. V. X. Ferreira and S. M. Weinberg. Credible, truthful, and two-round (optimal) auctions via cryptographic commitments. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 683–712, 2020. URL: `https://arxiv.org/pdf/2004.01598.pdf`.

[27] D. Finlay. EIP-2593: Escalator fee market change for ETH 1.0 chain. URL: `https://eips.ethereum.org/EIPS/eip-2593`, March 2020.

[28] Y. Hashimoto and S. Noda. Pricing of mining ASIC and its implication to the high volatility of cryptocurrency prices. URL: `http://dx.doi.org/10.2139/ssrn.3368286`, May 2019.

[29] Hasu. Transaction fee economics in NEAR. *Deribit Insights*, October 2020. URL: `https://insights.deribit.com/market-research/transaction-fee-economics-in-near/`.

[30] Hasu and G. Konstantopoulos. Analysis of EIP-1559. *Deribit Insights*, June 2020. URL: `https://insights.deribit.com/market-research/analysis-of-eip-1559/`.

[31] Hasu and G. Konstantopoulos. Analysis of EIP-2593 (escalator). *Deribit Insights*, June 2020. URL: `https://insights.deribit.com/market-research/analysis-of-eip-2593-escalator/`.

[32] Hasu, J. Prestwich, and B. Curtis. A model for Bitcoin's security and the declining block subsidy. URL: `https://uncommoncore.co/wp-content/uploads/2019/10/A-model-for-Bitcoins-security-and-the-declining-block-subsidy-v1.02.pdf`, October 2019.

[33] M. Honkasalo. A case study in miner extractable value. *The Block*, October 2020. URL: `https://www.theblockcrypto.com/genesis/79937/a-case-study-in-miner-extractable-value`.

[34] N. Houy. The economics of Bitcoin transaction fees. Working paper #1407, Groupe d'Analyse et de Théorie Economique Lyon St-Étienne (GATE Lyon St-Étienne), Université de Lyon. URL: `ftp://ftp.gate.cnrs.fr/RePEc/2014/1407.pdf`, February 2014.

[35] G. Huberman, J. Leshno, and C. C. Moallemi. An economic analysis of the Bitcoin payment system. Columbia Business School Research Paper No. 17-92. URL: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3025604`, 2017.

[36] Y. Khatri. First time in Ethereum's history, miners made more from fees than from block rewards. URL: `https://www.theblockcrypto.com/linked/79452/ethereum-miner-revenue-september-gas-fees`, October 2020.

[37] R. Lavi, O. Sattath, and A. Zohar. Redesigning Bitcoin's fee market. In *Proceedings of the World Wide Web Conference (WWW)*, pages 2950–2956, 2019. URL: `https://arxiv.org/pdf/1709.08881.pdf`.

[38] livnev. Random ordering of equally-priced transactions incentivises competitive spam. URL: `https://github.com/ethereum/go-ethereum/issues/21350`, July 2020.

[39] J. Lopp. The challenges of Bitcoin transaction fee estimation. BitGo blog post. URL: `https://blog.bitgo.com/the-challenges-of-bitcoin-transaction-fee-estimation-e47a64a61c72`, May 2017.

[40] F. Memoria. Ethereum Classic suffers third 51% attack in a month. URL: `https://www.cryptoglobe.com/latest/2020/08/ethereum-classic-suffers-third-51-attack-in-a-month/`, August 2020.

[41] A. Miller. Feather-forks: enforcing a blacklist with sub-50% hash power. URL: `https://bitcointalk.org/index.php?topic=312668.0`, 2013.

[42] M. S. Miller and K. E. Drexler. Markets and computation: Agoric open systems. In B. A. Huberman, editor, *The Ecology of Computation*, pages 133–176. North-Holland, 1988. URL: `https://agoric.com/assets/pdf/papers/markets-and-computation-agoric-open-systems.pdf`.

[43] B. Monnot. EIP 1559: A transaction fee market proposal. URL: `https://github.com/ethereum/rig/blob/master/eip1559/eip1559.ipynb`, April 2020.

[44] B. Monnot. The floating escalator: Combining 1559 and the escalator. URL: `https://nbviewer.jupyter.org/github/barnabemonnot/abm1559/blob/master/notebooks/floatingEscalator.ipynb`, October 2020.

[45] B. Monnot. Strategic users in EIP 1559. URL: `https://nbviewer.jupyter.org/github/barnabemonnot/abm1559/blob/master/notebooks/strategicUser.ipynb`, September 2020.

[46] D. J. Moroz, D. J. Aronoff, N. Narula, and D. C. Parkes. Double-spend counterattacks: Threat of retaliation in proof-of-work systems. arXiv:2002.10736. URL: `https://arxiv.org/pdf/2002.10736.pdf`, Feburary 2020.

[47] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Unpublished white paper. URL: `https://bitcoin.org/bitcoin.pdf`, 2008.

[48] J. Newbery. An introduction to Bitcoin Core fee estimation. *Bitcoin Tech Talk*, September 2017. URL: `https://bitcointechtalk.com/an-introduction-to-bitcoin-core-fee-estimation-27920880ad0`.

[49] D. Pérez and B. Livshits. Broken Metre: Attacking resource metering in EVM. In *Proceedings of the 27th Annual Network and Distributed System Security Symposium (NDSS)*, 2020. URL: `https://arxiv.org/pdf/1909.07220.pdf`.

[50] Pintail. Ethereum fee market reform: EIP-1559 as a question of fairness. URL: `https://pintail.medium.com/ethereum-fee-market-reform-eip-1559-as-a-question-of-fairness-567c52dac017`, August 2020.

[51] H. Qureshi. Blockchain fees are broken. Here are 3 proposals to fix them. URL: `https://haseebq.com/blockchain-fees-are-broken/`, May 2019.

[52] P. R. Rizun. A transaction fee market exists without a block size limit. Block size limit debate working paper. URL: `https://www.bitcoinunlimited.info/resources/feemarket.pdf`, August 2015.

[53] D. Robinson and G. Konstantopoulos. Ethereum is a dark forest. URL: `https://medium.com/@danrobinson/ethereum-is-a-dark-forest-ecc5f0505dff`, August 2020.

[54] T. Roughgarden. *Twenty Lectures on Algorithmic Game Theory.* Cambridge University Press, 2016.

[55] T. Roughgarden. *Algorithms Illuminated, Part 3: Greedy Algorithms and Dynamic Programing.* Soundlikeyourself Publishing, 2019.

[56] G. Wood. Ethereum: A secure decentalized generalised transaction ledger (Petersburg version). URL: `https://ethereum.github.io/yellowpaper/paper.pdf`, September 2020.

[57] A. Yaish and A. Zohar. Correct cryptocurrency ASIC pricing: Are miners overpaying? arXiv:2002.11064. URL: `https://arxiv.org/pdf/2002.11064.pdf`, February 2020.

[58] A. C.-C. Yao. An incentive analysis of some Bitcoin fee designs. arXiv:1811.02351. URL: `https://arxiv.org/pdf/1811.02351.pdf`, November 2018.

[59] M. Zoltu. EIP-1559 51% attacks: Should you live in fear? URL: `https://medium.com/@MicahZoltu/eip-1559-51-attacks-should-you-live-in-fear-d817be3759dc`, August 2020.