EIP-1559 – analysis of the research and discussions

Tomasz Stanczak, Nethermind

I provide a summary of Nethermind's current state of understanding of EIP-1559 and planned participation in the EIP-1559 research and implementation.

The initial pages are followed by the latest copies of various referenced links / materials (so most of this document is just a copy/snapshot of existing document).

Based on the goals, solutions, concerns that follow we suggest the current actions:

- Prepare models that separately treat escalator and EIP-1559 first and only then potentially merge them and analyse combined impact.
- For Ropsten – only fork when spec is finalized, and implementations are ready, and tests are written. Ropsten should remain at a 50%-50% split between the old and new model until mainnet is switched.
- Prepare an analysis for the PoA networks like Goerli and Rinkeby (to confirm whether they stay on the old transaction model or they fully transition to the new model).
- Address the mostly repeated concern of miners potentially bringing BASEFEE down through detailed simulations and analysis.
- Add an analysis of the impact of 20 million gas blocks as an attack vector.
- For network simulation – prefer simulation tools like the one provided by RIG team (Jupyter notebook) and agent models with easy to play with parameters and potentially a game for the community to successfully execute an attack on the network within the simulation model.
- Most likely do not rely on historical mempool and transaction analysis as these transactions were created in a very different model and will not be relevant enough for the new model.

Agent system / simulation game:

- The game would have agents with different objectives
    - Urgent tx push
    - Bring the network down
    - Pay low fees
    - Reliable tx push
- The objectives can be further generalized into a simple single objective of gathering most resource by all means possible
- Genetical algorithms and/or machine learning should be used to create most efficient agents under given assumptions
- On the outside the simulation game should be simple and user friendly so it provides great visualization for EIP-1559 and invites people to play with parameters
- Some incentivization can be offered to community if they provide an agent strategy that can bring the network down
- For each simulation we will analyse

- liveness of the chain (whether new blocks are produced, and transactions are taken out of the mempool)
- average gas price paid
- burn ratio
- reliability of tx broadcasting
- maximum increase in the size of the stat
- maximum increase in the size of block bodies

Within Nethermind team we express willingness to cover some of the research and to work on simulations. I believe that RIG solution is so good and so similar to what our approach would be that the most reasonable way for us would be to cooperate with RIG team / build on top of their existing solution. We can also add some integrations and insights for RIG simulations based on our experience with the nodes / networks.

We base our understanding of EIP-1559 on the following documents:

Date (DD/MM/YYYY)	**Document / discussion / call**	**Who**
09/05/2020+	Discord Channel	Micah, Barnabe, Tim, Sam, Abdel, Vitalik, Rick, James, Dan, aliatiia, Ivan, meridian, Gichiba, Rick
16/05/2020	RIG team analysis	Barnabe
14/05/2020	Measuring Eth1 network metrics	Tas, Horacio, Sandra, Tim, Abdel, Piper, Tomasz
30/04/2020	EIP-1559 Implementers Call #1	Barnabe, Ian, Tim, Vitalik, Dan, Abdel
03/02/2020	Moloch DAO statement of work - economics	Tanner Hoban, Thomas Borgers
November 2019	EthResearch Escalator and EIP-1559 discussion	Dankrad, Patrick, danfinlay, Vitalik, Barnabe, Micah

September 2019 – January 2020	Vulcanize Implementation in Geth Geth PR	Ian, Rick
13/04/2019	EIP-1559 Draft EIP-1559 Update EIP-1559 PR	Vitalik, Eric, Matt, Ian, Rick
01/03/2019	Ethereum Magicians discussion	Eric + discussion
August 2018	Resource Pricing	Vitalik + discussion
July 2018	First and Second Price Auctions	Vitalik + discussion

SUMMARY of EIP-1559 discussions:

GOALS:

Goal	**Stated by whom and where**	**Stated when**
Our goal is to discourage the development of complex miner strategies and complex transaction sender strategies in general	Vitalik in First and Second Price Auctions	July 2018
Remove the block congestion	Vitalik in discussion to First and Second Price Auctions	July 2018

CONCERNS:

Concern	**Stated by whom and where**	**Stated when**
As a user, I can already do this by targeting a price that would have gotten into the last n blocks plus some	Micah Zoltu in discussion to First and Second Price Auctions	July 2018
The gas prices are also volatile because a lot of miners pay their pool participants in the own blocks they mine. The average gas price is around 60 GWEI right now, but a lot of blocks have 1 GWEI lowest transactions in them. These transactions are actually miners paying their pool participants and pay 1 GWEI to themselves for this (from the account which is mining the block). This hence gives a wrong image about the actual gas price of a block.	MoonMissionControl in discussion to First and Second Price Auctions	July 2018
From a miner's perspective fee is payment for an increased uncle risk, so decreasing the total received fee must lead to reduced capacity, assuming rational miners. For this reason, the minimum fee system would increase volatility of fees and reduce throughput, as fees would have to be the same as they are currently *on top* off minimum fee. It is a dynamic tax creating a deadweight loss.	nootropicat in discussion to First and Second Price Auctions	July 2018
Something to be concerned about is that miners will have an incentive to keep the price minimum artificially low, because	MPR in discussion to First and Second Price Auctions	July 2018

they will only get paid the difference between the real price and the minimum price. If more than 50% of mining pools are capable of colluding with eachother then they will be capable of lowering the price minimum to near zero by only accepting a tiny fraction of transactions.		
Back to what I started with: I fail to see how lowering user-born fees (and by necessity lowering F, since F is the tips part in basefees+tips, and ostensibly basefees+tips < fees…) will result in anything but the miners becoming reluctant to include transactions.	Veox in Ethereum Magicians discussion	Mar 2019
Why did 1559 make the change where the tx bid over the base_fee is now ignored, instead of pocketed by the miner like in this draft *(July 2018 draft)*? I'm not saying the prior was better, but it's a change from this foundation that I didn't see described in 1559.	Dan Finlay in discussion to First and Second Price Auctions	May 2020
Why is the multiplier downwards same as upwards? (1 * 0.8 * 1.2 = 0.96 problem)	Nethermind, here	
One thought: if the "base fee" portion of the txn fee paid is burned, and the only portion that a miner/validator earns would be the "tip" above and beyond what is paid for the base fee, this means that we are net reducing the economic rewards that miner/validators earn (this should be obvious).	fubuloubu in Ethereum Magicians discussion	Mar 2019

If we reduce the issuance reward portion of what miner/validators earn (as is planned for future block reward reductions in PoW and the rewards system in PoS), this reduces the incentive for the miner/validators to actually include transactions in a block (since they are getting paid regardless via inflation), meaning the total congestion goes down until txns with “tips” are included.		
I have tried to run a simple model to see how MIN_FEE would look like for the existing history of blocks and transactions. Assuming that TARGET_GAS_USED is equal to the half of the block’s gas limit. The first thing I noticed is that formula for MIN_FEE gets stuck at 0 if it ever gets there. Any suggestions?	Alexey Akhunov in Ethereum Magicians discussion	April 2019
What is to stop miners constantly reducing MINFEE to 0, at which point we’re back to a pure auction-based system? The transaction pool cannot be considered a valid measure of network capacity (given that it’s variable across nodes) so it would have to come down to block capacity. Why wouldn’t miners mine empty blocks until MINFEE is 0, then only fill blocks half-full to keep it there?	mcdee in Ethereum Magicians discussion	April 2019
https://medium.com/@hongji/an-economic-analysis-on-eip-1559-fe72cd332d4f	editor-Ajian in Ethereum Magicians discussion	May 2019

A complication might arise around transaction propagation. Currently transactions have a total order, so full nodes can pick the best N transactions to keep and discard anything afterwards as SPAM). The base fee model could make things messy here, because the total order can change across blocks. We need to ensure that it's not possible to create transaction propagation scenarios where data rotates in the network indefinitely.	Karalabe in Ethereum Magicians discussion	May 2019
Furthermore (because the BASEFEE is equivalent to taxation) there will be lower welfare. A BASEFEE is also highly *unfair* since rich and poor individual pay the same fee. Imagine a scenario where there is a period of high network usage and therefore a high BASEFEE and tip. People with high willigness to pay will get their transactions included(probably rich people). Now imagine a period of low network usage where there would be low tips and a person with low willigness to pay could get his transcation included. Problem is that he inherits a high BASEFEE from the previous period. It`s like you're driving at night to avoid traffic and then the police stops by to say: "Sorry there have been a lot of traffic today please drive slowly" and you think ** because the street is fully empty.	STAGHA in Ethereum Magicians discussion	May 2019

SUGGESTED CHANGES / EXTENSIONS:

Change / extension	**Stated by whom and where**	**Stated when**
Fee as a function of time and/or block height is a very good idea, although hypothetically it could incentivize intentional orphaning. The difference is that the current way of replacing transactions hides future pricing information.	nootropicat in discussion to First and Second Price Auctions	July 2018
There's no dos risk as long as nodes verify blocks in parallel (even on single core), I think parity already does it. Ie. when two blocks at the same height appear, the one that wins (locally) is the one that gets verified faster, verification of the other one stops. So even a 1B gas block would do nothing bad to the network.	nootropicat in discussion to First and Second Price Auctions	July 2018 *Our comment: we consider this comment very dangerous if ever accepted by the community as valid.*
Minimum fee introduces potential incentive problems here, as blocks that increase the minimum fee make child blocks less valuable. If major miners/pools can keep the 8M limit, they can enforce ~0 minimum fee by orphaning blocks that go above 50%, possibly with some stealth like 'orphan only if younger than x seconds'.	nootropicat in discussion to First and Second Price Auctions	*Our comment: this is both very interesting and hard to simulate*
Rather than adjusting the MINFEE by 1/8 in each block, is there a reason we can't adjust it once a day to remove fluctuation even more? I would like to see a target of 50 Billion gas *per day* rather than 8 million per block, and triple the max size of each block to 24 million gas (uncles allowing). That way if a big ICO or surge of spam hits, we could see 100 Billion in gas for 24 hours, and then MINFEE increases once every 6200 blocks (like Bitcoin-style	BFire in Ethereum Magicians discussion	April 2019

difficulty adjustments) until the daily gas usage falls below the target…at which point it slowly starts coming back down. This would have the nice effect of doubling current transaction capacity during busy periods while keeping the state size growth where it is now.		
With gas burning and EIP1559 we could define a rule where any unused gas below half of the block gas limit would be taken of the miner's block reward at the gas price equal to the base fee. Incentivizing transaction inclusion and making a statement that empty blocks are less important for the network.	Nethermind, here	
Excalator https://agoric.com/papers/incentive-engineering-for-computational-resource-management/full-text/	Danfinlay in Ethereum Magicians discussion	November 2019

SUGGESTED SOLUTIONS:

Solution	**Stated by whom and where**	**Stated when**
Minimum block number	Vitalik in First and Second Price Auctions	July 2018
The mechanism maintains a minimum fee F. Every transaction specifies a fee. For a transaction to be included in a block, the transaction must pay at least F. The fee is adjusted every block by the following formula,	Vitalik in First and Second	July 2018

where prevBlockGas/prevBlockMaxGas is the portion of the previous block that was full, and k is a constant ($0 < k < 2$): $curBlockFee=prevBlockFee*(1+k*(prevBlockGas/prevBlockMaxGas-1/2))$	Price Auctions	
If you can capture and burn a percentage of transaction fees, then you can burn them them proportionally to the number of non-participating validators, thus providing an in-protocol incentive for validators to support the inclusion of other validators.	Dan Robinson in discussion to First and Second Price Auctions	July 2018
Fee to be calculated via EVM bytecode in the transaction	Dan Robinson in discussion to First and Second Price Auctions	July 2018

NETHERMIND IDEAS

Idea	**Critique**	**More comments**
Address Ranges (you can be assigned to a group of addresses with whom you compete for tx inclusion)	Seems useless	
Limitation on the transactions sent from an address	Sender can create multiple accounts, bureaucratic / regulatory	
Limitations on the transactions sent to a given contract within a given block range	Bureaucratic / regulatory	

Binding difficulty of the next block with the size of the previous block	Complex to test	Bigger blocks would result in higher next block difficulty, thus compensating longer processing times.
Miner could select higher difficulty in order to increase the block size.	Potentially a dominant miner could attack the network by raising difficulty significantly and raising block sizes at the same time.	

ATTACK STRATEGIES:

Optimize For	**Stated by whom and where**	**Stated when**
Try to fill the blocks		
Try to raise the gas price		
Try to make the block sizes too low		
Try to make the block sizes too high		
Try to raise the base fee		
Making the network unusable requires no mining hardware, only money. Assuming 8M gas limit, spamming enough transactions to keep the gas price at 200 gwei costs a maximum of $4.3M/day, in reality less due to transactions made by genuine users. What would happen with adoption after a 10 day attack? Even 2000 gwei is feasible.	nootropicat in discussion to First and Second Price Auctions	

Exciting research overall and the big picture is crystal clear, but from what I understand the choice of the actual update rule for dynamic pricing is not justified very well. For this specific proposed formula, I have an attack in which a small number of users with only 10% of transaction volume can halve the price in less than 1000 blocks only by a particular timing for broadcasting their transactions.	Mtefagh in Ethereum Magicians discussion	April 2019

AGENT OPTIMIZATION:

Optimize for	Who and why?	
Low gas fees	Users will seek lowest fees to execute transactions that are useful to them	
Highest gas fees + block rewards	Miners targeting max potential block revenue	
Network failure via: • Too big blocks ○ Slower propagation ○ Faster state growths • Network stuck • Fees to high • Network only accepting useless transactions • Multiple network forks	Competitors benefitting from network failure	
Fast execution	Arbitrageurs / traders planning to execute as fast as possible	

STAKEHOLDERS:

Who?	Why?	
ETH Investors	They believe that burning fees would lead to a higher ETH price. They are worried about potential failure / instability.	
Miners	They are worries about fees lost (fees burnt).	
Wallet Developers	They are worried about required code changes. They hope for easier gas price calculations.	
Users	They hope for less network congestion. They hope for more predictable fees. They are worried about higher total average fees.	

Simple Summary

The current "first price auction" fee model in Ethereum is inefficient and needlessly costly to users. This EIP proposes a way to replace this with a mechanism that adjusts a base network fee based on network demand, creating better fee price efficiency and reducing the complexity of client software needed to avoid paying unnecessarily high fees.

Abstract

There is a base fee value in protocol, which can move up or down by a maximum of 1/8 in each block. The base fee value is adjusted by the protocol to target an average gas usage of 10 million, increasing base fee if usage is higher and decreasing it if usage is lower. Transaction senders specify their fees by providing two values:

- A gas premium which gets added onto the base fee to calculate the gas price. The gas premium can either be set to a fairly low value (eg. 1 gwei) to compensate miners for uncle rate risk or to a high value to compete during sudden bursts of activity. The base fee gets burned, the gas premium is given to the miner.

- A fee cap which represents the maximum total (base fee + gas premium) that the transaction sender would be willing to pay to get their transaction included.

The current miner-voting based gas limit is changed to a hard-coded gas limit of 16 million. Instead of miners directly adjusting the gas limit in response to changes in network demand, the protocol adjusts the base fee to apply economic pressure towards a target gas usage of 10 million.

Motivation

Ethereum currently prices transaction fees using a simple auction mechanism, where users send transactions with bids ("gasprices") and miners choose transactions with the highest bids, and transactions that get included pay the bid that they specify. This leads to several large sources of inefficiency:

- **Mismatch between volatility of transaction fee levels and social cost of transactions**: transaction fees on mature public blockchains, that have enough usage so that blocks are full, tend to be extremely volatile. On Ethereum, minimum fees are typically around 2 gwei (10^9 gwei = 1 ETH), but sometimes go up to 20-50 gwei and have even on one occasion gone up to over 200 gwei: https://etherscan.io/chart/gasprice. This clearly creates many inefficiencies, because it's absurd to suggest that the cost incurred by the network from accepting one more transaction into a block actually is 100x more when gas prices are 200 gwei than when they are 2 gwei; in both cases, it's a difference between 8 million gas and 8.02 million gas.
- **Needless delays for users**: because of the hard per-block gas limit coupled with natural volatility in transaction volume, transactions often wait for several blocks before getting included, but this is socially unproductive; no one significantly gains from the fact that there is no "slack" mechanism that allows one block to be bigger and the next block to be smaller to meet block-by-block differences in demand.

- **Inefficiencies of first price auctions**: see https://ethresear.ch/t/first-and-second-price-auctions-and-improved-transaction-fee-markets/2410 for a detailed writeup. In short, the current approach, where transaction senders publish a transaction with a fee, miners choose the highest-paying transactions, and everyone pays what they bid, is well-known in mechanism design literature to be highly inefficient, and so complex fee estimation algorithms are required, and even these algorithms often end up not working very well, leading to frequent fee overpayment. See also https://blog.bitgo.com/the-challenges-of-bitcoin-transaction-fee-estimation-e47a64a61c72 for a Bitcoin core developer's description of the challenges involved in fee estimation in the status quo.
- **Instability of blockchains with no block reward**: in the long run, blockchains where there is no issuance (including Bitcoin and Zcash) at present intend to switch to rewarding miners entirely through transaction fees. However, there are known results showing that this likely leads to a lot of instability, incentivizing mining "sister blocks" that steal transaction fees, opening up much stronger selfish mining attack vectors, and more. There is at present no good mitigation for this.

The proposal in this EIP is to start with a base fee amount which is adjusted up and down by the protocol based on how congested the network is. To accommodate this system, the total network capacity would be increased to 16 million gas. When the network exceeds the target 10 million gas usage, the base fee increments up slightly and when capacity is below the target, it decrements down slightly. Because these increments are constrained, the maximum difference in base fee from block to block is predictable. This then allows wallets to auto-set the gas fees for users in a highly reliable fashion. It is expected that most users will not have to manually adjust gas fees, even in periods of high network activity. For most users post 1559 implementation the base fee will be estimated by their wallet and a small gas premium- which acts as a 'tip' to compensate miners (e.g. 0.5 gwei)- will be automatically set. Users can also manually set the transaction fee cap to bound their total costs.

An important aspect of this upgraded fee system is that miners only get to keep the tips. The base fee is always burned (i.e. it is destroyed by the protocol). Burning this is important because it prevents miners from manipulating the fee in order to extract more fees from users. It also ensures that only ETH can ever be used to pay for transactions on Ethereum, cementing the economic value of ETH within the Ethereum platform. Additionally, this burn counterbalances Ethereum inflation without greatly diminishing miner rewards.

The transition to this gas price system will occur in two phases, in the first phase both legacy and EIP1559 transactions will be accepted by the protocol. Over the course of this first phase the amount of gas available for processing legacy transactions will decrease while the amount of gas available for processing EIP1559 transactions will increase, moving gas from the legacy pool into the EIP1559 pool until the legacy pool is depleted and the EIP1559 pool contains the entire gas maximum. After all of the gas has transitioned to the EIP1559 pool, the second- finalized- phase is entered and legacy transactions will no longer be accepted on the network.

Specification

Parameters

- `INITIAL_FORK_BLKNUM`: TBD
- `BASEFEE_MAX_CHANGE_DENOMINATOR`: 8
- `TARGET_GAS_USED`: 10,000,000
- `MAX_GAS_EIP1559`: 16,000,000
- `EIP1559_DECAY_RANGE`: `MAX_GAS_EIP1559 / 20 == 800,000`
- `FINAL_FORK_BLKNUM`: `INITIAL_FORK_BLKNUM + EIP1559_DECAY_RANGE`
- `EIP1559_GAS_INCREMENT_AMOUNT`: `(MAX_GAS_EIP1559 / 2) / EIP1559_DECAY_RANGE == 10`
- `INITIAL_BASEFEE` : 1,000,000,000 wei (1 gwei)
- `PER_TX_GASLIMIT`: 8,000,000

Proposal For all blocks where `block.number >= INITIAL_FORK_BLKNUM`:
For the gas limit:

- `MAX_GAS_EIP1559` acts as the hard in-protocol gas limit, instead of the gas limit calculated using the previously existing formulas
- The `GASLIMIT` field in the block header is the gas limit for the EIP1559 gas pool, and over the transition period this value increases until it reaches `MAX_GAS_EIP1559` at `FINAL_FORK_BLKNUM`
- The gas limit for the legacy gas pool is `MAX_GAS_EIP1559 - GASLIMIT`, as `GASLIMIT` increases towards `MAX_GAS_EIP1559` gas is moved from the legacy pool into the EIP1559 pool until all of the gas is in the EIP1559 pool
- At `block.number == INITIAL_FORK_BLKNUM`, let `GASLIMIT = (MAX_GAS_EIP1559 / 2)` so that the gas maximum is split evenly between the legacy and EIP1559 gas pools
- As `block.number` increases towards `FINAL_FORK_BLKNUM`, at every block we shift `EIP1559_GAS_INCREMENT_AMOUNT` from the legacy pool into the EIP1559 gas pool
- At `block.number >= FINAL_FORK_BLKNUM` the entire `MAX_GAS_EIP1559` is assigned to the EIP1559 gas pool and the legacy pool is empty
- We enforce a maximum gas usage on individual transactions: `PER_TX_GASLIMIT`

For the gas price:

- We add a new field to the block header, `BASEFEE`
    - `BASEFEE` is maintained under consensus by the ethash engine
- At `block.number == INITIAL_FORK_BLKNUM` we set `BASEFEE = INITIAL_BASEFEE`
- `BASEFEE` is set as follows
    - Let `delta = block.gas_used - TARGET_GASUSED` (possibly negative).
    - Set `BASEFEE = PARENT_BASEFEE + PARENT_BASEFEE * delta // TARGET_GASUSED // BASEFEE_MAX_CHANGE_DENOMINATOR`
    - Clamp the resulting `BASEFEE` inside of the allowable bounds if needed, where a valid `BASEFEE` is one such that `abs(BASEFEE - PARENT_BASEFEE) <= max(1, PARENT_BASEFEE // BASEFEE_MAX_CHANGE_DENOMINATOR)`
- We add two new fields to transactions: `GAS_PREMIUM` and `FEECAP`
- During the transition phase, these fields can be left `nil` and a `tx.gas_price` can be set as usual to generate a backwards compatible legacy transaction
- To produce an EIP1559 transactions, `tx.gas_price` is set to `nil` while the new `GAS_PREMIUM` and `FEECAP` fields are set whereby:
    - `GAS_PREMIUM` serves as a "tip" to the miner
    - `FEECAP` serves as the absolute maximum that the transaction sender is willing to pay
- During transaction execution, for EIP1559 transactions we calculate the cost to the `tx.origin` and the gain to the `block.coinbase` as follows:
    - Set `GASPRICE = min(BASEFEE + tx.GasPremium, tx.fee_cap)`
    - Let `GASUSED` be the gas used during the transaction execution/state transition
    - The `tx.origin` initially pays `GASPRICE * tx.gas`, and gets refunded `GASPRICE * (tx.gas - GASUSED)`
    - The `block.coinbase` gains `(GASPRICE - BASEFEE) * GASUSED`.
        - If `GASPRICE < BASEFEE` (due to the `FEECAP`), this means that the `block.coinbase` *loses* funds from this operation; in this case, we check that the post-balance is non-negative and throw an exception if it is negative.

Backwards Compatibility

We split the EIP1559 upgrade into two phases with a transition period during which both legacy and EIP1559 transaction can be accepted so that compatibility with wallets and other ETH-adjacent software is maintained while their maintainers have time to upgrade to using the new transaction type. During this transition period legacy transactions are accepted and processed identically to the current implementation, with the only difference being that the amount of gas (gas limit) dedicated to processing legacy transactions is calculated as above and incrementally decreases over this period.

Test Cases

Implementation

Go-ethereum implementation by Vulcanize Inc: https://github.com/vulcanize/go-ethereum-EIP1559

Security Considerations

The security considerations for this EIP are:

1. The consequences of raising the gas limit
    - This concern was brought up here
    - This EIP currently proposes to raise the total gas limit from 8,000,000 to 16,000,000
2. The consequences of the new gas pricing on total transaction order
    - This concern was brought up here
    - This issue is avoided by maintaining a single total ordering of transactions by price and nonce, where the derived EIP1559 gas price is used like the legacy gas price
3. The effects on miner incentives
    - This concern was brought up here
    - This concern is addressed here
4. Concerns of BASEFEE manipulation
    - This concern was brought up here
    - To avoid this, the BASEFEE is included as part of the header structure and is maintained under consensus by the ethash engine
5. Implications of BASEFEE burning on proposals to cap the total ether supply (e.g. EIP-960).
    - This concern was brought up here

Resources

- Call notes
- Original Magicians thread
- Ethresear.ch Post w/ Vitalik's Paper
- Go-ethereum implementation
- Implementation-specific Magicians thread

Copyright

EIP-1559

Martin: That is Fee market change. I wonder, if there is anyone here who is it sufficiently knowledge about it?

Rick: Hi, it's this is Rick and Ian, the dev who wrote the patches is here as well.

Hudson: Awesome, go ahead and speak on it if you'd like.

Rick: Yeah, just a friendly reminder of what the purpose of it. The high-level points

- it's an EIP by Vitalik.
- to **add stability to Gas Ether price**.
- it has some other useful side effects, in terms of **removing zero fee transaction**. And the way it does this is basically we add a base fee, we bring the gas pricing under consensus and then each transaction instead of having a single component in terms of the fee that goes to the miners.
- you now have **two components** - some of the gas is burned and some of the gas goes to the miners, that's basically it.
- Vitalik wrote skinny 1559 which is not what we implemented,
- we implemented what was in the implementation study which is two phases.
- another major benefit is it really simplifies. you **no longer need Ether gas station**. it really simplifies the user experience, that's much easier to figure out what your gas fee is going to be in advance. because it changes how transactions work, you know you can't just flip a switch and expect all of the downstream tooling to have changed overnight. To me, that was a much greater risk and so we've **made it two phases** so that there's a period of time where both transaction types are valid which adds some complexity but I think it's necessary to actually get adoption without breaking everything. Any questions?

Martin: Yes, actually two questions. So first of all it sounded that you're talking about EIP as one thing and implementation as different thing. I'm kind of wondering what to discuss today? The EIP or the alternate version of the EIP?

Rick: Yeah, so it's the implementation. It's a pretty big change and the EIP was a little, in my opinion, a little under specified. we had hopes to have a EIP that referenced our implementation. well that's inaccurate. what I had hoped to have happened was that we'd actually do some modeling and simulations to sort of proves that this isn't going to blow everything up. but we couldn't get funding for that. so what we got funding for was the implementation. so we wrote an implementation. And in that process, there was changes need to be made and you know the EIP is forthcoming. but those are separate tracks, those are two different people working on those things and so the implementation was the priority and it also coincidentally, frankly, happened to be done first.

Martin: Okay, so it sounds to me that the this point is real **a bit too early to make some kind of call about the inclusion**. I mean, you have an implementation but not really an EIP to discuss. But I do have some more general question on 1559. So, it has this premium base fee and the premium. I don't really understand **what prevents all the miners from basically colluding and setting the base fee to 0 and still only accepting transactions which have highest premium and failing back, basically to the same situation we're out today?**

Rick: Before I answer that question, I completely understand what you're saying about the EIP and in the review. In my discussions with people, to be frank, it seem as though it's the shortest path to having a discussion of frankly. Since no one responded to any of my comments or anything else that wasn't. And I mean to say no one responded, I mean very few people responded. I was having a difficult time.

Martin: I think, it's fine. Probably, we won't be able to make a decision today.

Rick: No, I don't expect. perfect to have a discussion about. There is an averaging of the base fee over a large number of blocks and so the idea that the miners would need to kind of have an overwhelming amount of the transaction volume for a long period of time to adjust the price. They basically need to be the majority of demand, it's kind of a weird.

Martin: Does the base fee follow the premium ? what does the bass fee follow?

Rick: The base fee target the half-full block. We set up an initial value that in the original EIP, took a snapshot of time but the gas price at the time, that is deployed and so basically the idea is that initial price is set and then it can only vary so much per block and the target price is determined by taking some average number of blocks. These are the sort of questions where I thought they were very difficult to answer and this attack that you pointed out we've sort of this sketch solution but I felt like giving the importance of the change we needed a lot more engagements actually answer a question.

Peter: There's another question. If I get it correctly, the idea would be the gas tries to keep blocks half-full. If blocks are getting fuller than the gas prices go up. The question is **how does this relate to the dynamic block sizes?** On Ethereum mainnet we kind of have it fixed at a million currently, but in theory, it should have been Dynamic. so if we add this, how will it do values in the place? because as the blocks are getting fuller the miners in theory will push the block size up which would make transactions cheaper and your proposal is doing the exact opposite. If we were to remove the limit on mainnet, this artificial 10 mil gas cap, then what would happen?

Rich: I'm thinking about that. How do we decide what the criteria is for changing it?

Peter: changing what ?

Rich: the 10 million gas cap. what in your suggestion, how is that changed?

Peter: The 10 mil gas cap limit currently, is an arbitrary limit set by miners. But based on the Ethereum protocol, it should be pushed upward if blocks are full.

Rich: I am sorry. That's under consensus.So, I think it would stay effectively, we don't change that, we just change the price.

Peter: yeah, so essentially the problem is that in theory what these Ethereum protocol specs is that if blocks are getting full, the original spec was that the gas limit should be raised. Now, you're saying that the price should be raised. But I think **it should be important to touch on what happens on a network where we don't have this limit**. eg Rikeby. Currently, we configured if the blocks are 10 mil in size but they are allowed to go up until 15 mil if there is high network traffic. Now in this case,the trigger for pushing the block limit up would be that the blocks are full but at the same time in your EIP, this would also trigger transactions to be so expensive that the blocks won't be pushed up.I just want to make sure that we're not accidentally murdering an existing mechanism with this one.

Rick: Well, I appreciate that comment. I think that it would be intentional. I didn't realize that Rinkeby had that dynamic pricing or the dynamic sizing.

Peter: If you're creating an EIP that deliberately murders it, that's fine from my perspective. I am completely fine with proposing an EIP that clearly states that this will be murdered.

Rick: Thanks for the feedback. That is exactly the feedback that I was having a difficult time receiving it.

Martin: I have another question, the right now there is a cap. we know that even when blocks are full, we won't go over but, here in this proposal, it looks like, we'll target 8-10 million but actually the hard cap is at three times that amount. So it might be suddenly 24 million gas block would be valid, am I reading it right ?

Rick: yeah it targets much lower than. right exactly so during normal it has room for congestion.

Martin: It feels almost reckless, but I mean there are security implication about having the roof three times higher than where we want to be. maybe want to be a bit more conservative actually.

Peter: It would be really denial of service. If somebody figures out the way to attack Ethereum all of a sudden if you have three times as much leeway.

Rick: I completely agree with that and I think that if someone were able to sustain that. As far as I'm concerned you know, I sort of volunteered to shepherd the EIP through. Of course these stuff, you guys know this stuff better than anyone else. I think these are really great questions and these are exactly the types of questions that I was trying to surface prior to writing any code. but this seems to be what most of the people who were giving me feedback in the community wanted, they wanted to see the implementation before we answer these sorts of questions.

Hudson: It sounds like the right way to go. I think it's good that you did that because then people can look at the code and then dissect it a little further than just a lot of hypothetical, I guess people would say.

Rick: It's not how I worked in my other professional capacities. I appreciate the feedback and will definitely keep that in mind.

Hudson: yeah taking this to like The Ethereum Magicians thread is going to be very helpful, I think to Rick and the rest of his team. so if anyone here has further stuff after looking deeper into the implementation, I think that would be important. and then even more important than that in my opinion would be an update to the EIP itself. even if it's not pushed through the EIP process having a PR that has the changes Rick that you and your team have implemented that might be different than the PR, which I think was last updated in April, that would be pretty important, so that people can comment on the latest one and not have to refer to a previous specification that's not updated.

Rick: yeah we'll take care of that hopefully this week.

Hudson: It's holiday, so it's not like huge rush or anything.

Peter: Before we kind of deflecting to different topic, I want to emphasize on this because I've a feeling that it's not taken as seriously as Martin intended. Currently 10 mil gas limit that Ethereum Network is running on. The Reason why. it was capped at 8 million because that was considered the only sane limit, so that this guy doesn't murdered the network. Yes, we did some optimization and now people pushed up the gas limit to 10 mil. But, we really don't want to get into position that all of a sudden (random number) 15 million things starting to get screwy. Now if you all of a sudden allow people to expand 24 mil then it's going to be really bad. that's why I'm saying that if we kind of currently feel that Ethereum network capacity is at 8 or 10 million, we should really have some very very hard caps in place so that you cannot really over blow the resource usage. So honestly instead of a 3x multiplier, maybe a 1.5 would be a lot more saner starting point.

Rick: Frankly, we went with the parameters that Vitalik gave us, where he gave us then, so I don't know why he picked such a large value and I'll definitely keep that in mind.

Peter: The 3 X is not a horribly bad idea if you would look at the average Network usage. so currently gas can process blocks in them maybe around 150 milliseconds. So, if you act 3x to that, that would mean may be 0.5 sec, so that's not that bad. But the thing is the whatever people throw at it when they are using Ethereum, it's not the worst place possible effects scenario. and you need to keep the limit in control for that scenario.

Rick: yes so so that when he did look at the distribution, I do remember from the EIP that it was based on an assumption of block distribution that I thought didn't really fit reality, and I think that you're touching on that point from a different perspective. yeah I agree . we're here having this discussion because I don't know how to demonstrate or simulate or make any sort of formal assertion about what that value should be? I'm definitely open to suggestion my intuition is that you would have to run a fairly robust simulation to answer that question.

Hudson: Was there anyone else with comments? Anything to add at the end of Rick ? It's best to reach out to you on Ethereum magicians I'm guessing, was there any other outlets that you wanted to bring up as far as how to address this or contribute?

Rick: we can keep these conversation at Eth Magician, that would be great. I don't know what the convention is around PRs. I think the code size is relatively small obviously the impact is very large. when I say PRs, they mean I don't know if they want to help, people want to interact with, if they do just ping me in the gitter. link is already provided. but If people have a hard time finding it or whatever we can sort of engage in the GitHub and EthMagicians.

Hudson: Awesome and just to extend my support on this, you can reach out on telegram, if you do have any questions about the EIP process or the process of getting this through for more potentially rapid discussion, I'm happy to talk to you about that.

Rick: Great thank you !

EIP 1559 Implementer’s call transcript

April 30

_

I have like a pretty I guess lightweight agenda for the call because this is the first first time I had a goal is mostly to get a feel for where everybody’s at and then what the next steps would be that move something like 1559 or Dan’s proposal forward.

06:03
It is obviously a pretty like sweeping change so we want to make sure that we roll it out with a lot of community engagement. Um, I’m not sure if I guess what I had in mind is just first, you know getting like a status update for the different teams which I think is just Besu and Geth right now working on this. and then

maybe it would make sense for you Dan could then give an overview of your your EIP and then we can probably get into the nitty gritty of like the various issues with either of the themes and and and how we make sure that we actually test those that make sense for everyone.

06:43
Yeah, no objections are just wrong around and yeah, maybe it makes sense for you Ian to start I know your team has been working on this for for the longest so that maybe just gives like a bit of context for people on the call like kind of the whole process you went through on a high level and and where the implementation is is up right now.

Ian:
Yeah, certainly so. I’ve been working on this for number of months now it started out with a higher level overview that was done by MS slipper. Matthew Slipper and I’m not sure everyone here has access to that document that might be helpful for directing implementations elsewhere and so that’s the higher level overview that we use to direct these implementation in go-ethereum.

https://docs.google.com/document/d/1yqvvfrQ_He0fN1SsUcvZNBdyhv__d8-1QPyteCbNT6Q/edit

07:33
So that PR was opened. I think it’s been a couple months now since that was opened and so to be honest, I haven’t really been thinking about it all that much and the time since. But the status of that go with that PR is that everything is ostensibly in place and ready to go for further testing before we go ahead and merge everything

The few things that I am certain need more work are to ensure that the light chain and the LES packages are covered by the changes and to also test the gas oracle for these new pricing is working properly and then also to ensure that the other consensus engines are hovered as well the current implementation focuses on the fast engine, of course, so the the main. The main thing that needs to be done as I see it at this point is larger scale system testing. there is unit testing included in that pull request but they're to my knowledge there has not been the larger scale system testing that needs to be done to prove that this model is safe and secure and ready to be merged over to mainnet.

08:55
And just today I went back over the PR and replied to some common comments on there for a few days and made some changes and specifically made the change to enforce that the gas price is greater than or equal to the base fee and also updated the max gas in EIP 1559 to 20 million instead of 16 million.

Abdel:
I can give the update for Besu. So this one implementation is almost aligned with the geth implementation. I was able to run a small test nets with some geth notes and some besu nodes. We are almost aligned except we don't have the mining for example, but I started the test net with some geth nodes and besu nodes and I was able to correctly synchronize blocks, etc.

09:57
So all consensu rules are okay. So especially the changes about new transactions structure and the block header structure. So everything works fine. To make the test with real transaction. I also submitted the PR in web 3G and the PR is merged. So in the next release of web 3G, it will be possible to kind of play with the new EIP1559 transactions.

10:27
So this is how I tested these small test net locally. And yeah, we need to work on the mining. And we did also the changes to enforce the gas price to be higher than the base fee, and change about the hard block unit to 20 million.

Tim:
Thanks guys. Yeah. I think Dan it probably makes sense for you now to give a high level overview of what your proposal is and and I think that the EIP did a good job like contrasting with with the 1559 but yeah. I think it would be valuable with everybody's here kind of have that context.

Dan:
Yeah should I the screen share or just describe it a high level. --I go ahead and screenshot if you want yeah sure.-- Uh, yeah, I'm just gonna show the the um, yeah, so yeah sorry for sharing this kind of so late not being more engaged in in

the kind of debate process, but um what I wanted to highlight is just a kind of alternative fee structure.

11:37
It was written in 1988 by Mark Miller and Eric Drexler. Mark Miller's cited in the ethereum yellow paper for pioneering the concept of smart contracts. And this algorithm was designed for allocation of CPPU over a bunch of bitters. In the escalator algorithm and this EIP structures it for the purposes of Ethereum.

11:58
Each bit has a starting price and ending price in its duration that its price its bid price would escalate during which and so on each block each miner would look at their mining capacity and of the current price of all available transactions, they would pick the highest ones they could fit into the block and include those.

12:21
Um I think that so there's a lot of different conditions – so I just wanted to highlight because I've seen like a lot of analysis of 1559 and you know, a lot of people are just focusing on the burning of fees and and I think that it's very important that we focus on user experience also.

There's also been a lot of advocacy under the case of normal conditions like high probability of next block inclusion and then I've also seen people suggest that it 1559 performs better under highly volatile conditions like under the recent transaction glut with the DAI kind of crash.

12:56
On but I think that if we compare each of these against specifically you so if we consider those various types of conditions with users that have different preferences both from price and for urgency. I think urgency is kind of a consideration parameter that's not getting considered very much in these analyses the simulation it was posted on ethresearch didn't consider it, but it actually is very much I think of parameter that's important, um, for example.

13:25
I think it was during the status ICO there was it was also right after. The ENS had launched and there were a lot of people whose ENS bids were closing. and so they had like a very very specific kind of end block and so they maybe didn't have like a very high cost preference in the short term, but they definitely had a time at which they would want to pay their highest price and escalator algorithms is very friendly to to that kind of preference.

There's been a little bit of a so I tried to do like a situation comparison so the blocks are regularly full and the user wants urgent inclusion so like this is I think the normal or wait a Conclusion okay so blocks are I I want blocks maybe under

full maybe half full user wants urgent conclusion so this is kind of the normal condition that I think 1559 is often hailed as is like really kind of ideal for because if you bid over the current price tier, you're likely to get included in the next block.
I think under current conditions, you know, it's not hard to over pay a little bit but that's the core criticism of the current single price auction.

14:36
I think the escalator algorithm also performs well here because the wallet can just start the Bid within or near the current accepted price range so I think it actually has a very high probability of getting the same next block inclusion as as you get out of 1559 except that unlike 1559 in situations where the blocks do fill up suddenly, the escalator algorithm prioritizes the person who's bidding the highest and I think that this is a kind of key a differentiator right now the the TIP parameter becomes the tiebreaker in situations where a block is full and and I know a lot of a lot of critics of the escalator algorithm.

15:16
I Highlight well a 1559 makes sure that blocks usually aren't full, you know, the price goes up but there's still a pool of transactions under that price and and so once the the price falls down to that the TIP comes the tie breaker and so so it seems to me like it's kind of recreating the single price auction and creates a kind of rare situation where sometimes people with urgent transactions can be basically in the same price pool as a bunch of people who were just waiting for the price to come down to that tier.

15:49
So I think that's kind of it in a nutshell. There's a lot of conditions to consider but I think that I think that escalator performs similarly under normal conditions and under extreme conditions. I think it performs better. And that's probably a very very high level summary of why I somewhat prefer it.

16:10
I don't I don't think it's like a deal-breaker. I don't think it's destroys protocol. I think it's got some slightly beneficial properties specifically from my experience dealing with users who are freaking out during high urgency scenarios, which which are moments when users care a lot about proper transaction processing.

16:41
Yeah, any questions?

Ian:
So this is probably a pretty naive question but just immediately one of the things that comes to mind is what wise that the price is increasing over time instead of decaying it seems like a user should you know would be willing to pay more if

the transaction got accepted right away and if they have to wait longer than they should pay less.

Dan:
I think it I think that by increasing you have like better low price discovery. If you were decaying then you'd have a similar problem to today where somebody who wants an next block inclusion has to dramatically overpay and and now it's going to decay down and so you have a lot of people just overpaying, you know, 10x or 100x over the current price. By gradually increasing you're going to find the the lowest price that you would pay with similar to 1559, right?

Like the price is going to go up until you know, it reaches where you would get processed. Except under this condition, you kind of have a way of spiking through a little faster even if there's A lot of transactions at the current price tier.

Ian:
Kind of related does this add a new gameability to the system that miners as a whole are somewhat incentivized to wait until the final block where they'll be paid the most.

Dan:
Yeah you know I think there's a lot of games like that I was trying to think of a good word for this is like the opposite of chicken where like a bunch of people are waiting because like incentives getting bigger over time. But because there are other miners watching you don't want to be the you don't be the slow one.

18:15
Right? Yeah, I was I was thinking about this because I berries ripping and I I'm playing this game with my birds, you know the birds and my yard. Yeah, I think that this has an equilibrium where where you know, maybe you want to wait a little bit but you're not gonna wait it forever because you know, there are other people racing for it.

Tim:
I guess users and wallets can also adapt to that right like say if because I assume from the user perspective you might want to abstract say, you know what's like the median price you're willing to pay and whatnot and then the wallet can kind of, you know, submit the minimum price that's like under that and in a higher price it's over that and and and over time will have data on like, you know at what that's what percentage of that of that curve do transactions typically get a included.

19:06
Dan:
Yeah yeah and and as a wallet like you can also start a little below the current

median and and that can help drive overall market prices down. while the developers today don't have a great way to to drive the prices down like users want, you know near next block inclusion and the only way to do that is to bid over median, which means that prices tend to upwards over time.

19:37
Ian:
So this is um, not really related to the last question but um, one of the things that I think some members in the community are excited about for 1559 was the burning of the base fee and the you know, the apparent deflationary aspect of that so just wondering if you could talk to that and whether or not, 1559 could use that to kind of garner attention and support and if there's some way that we could translate that to to this EIP as well.

20:11
Dan:
Yeah I mean my impression is that the burning was introduced to solve the problem where the second price action doesn't fit under this kind of game. but I think that you could introduce burning on any transaction like burn a percentage of every transaction for example and you could slap it on to any any bidding mechanism.

Vitalik:
So the reason why this gets subtle is because like there are a lot of the kind of more naive proposals that set people sometimes make the reform fee markets, like I think 1559 and your escalator algorithm are both immune. But like some, One of the academic ones I saw that a couple years ago is not. which is collusion between miners and and the transaction senders by making side payments.

so basically if you have a mechanism where like even just taking as an example of partial fee burning, so we burn 50% of the fee, then you like if you're a major at transaction sender you have an incentive to go make a private deal with say either mine and you go say like hi ether mine, how about all send some I'll send my transactions with a gas price of two wei and At the same time kind of off on the side we'll do some economic how much money I owe you and I'll pay you the the real fee using some other mechanism. and so like that can't get taxed right and the reason why 1559 is immune to this the becausethe the fee isn't the amount that gets burned isn't dependent on any parameter that's set the transaction. The fee that gets burned that's only dependent on the amount of gas consumed, so like no matter how you do it there's going to be gas the times base fee that gets burned and there's not any way to get to get around it.

And in the escalator case, that's basically just. The existing fee market with a with a modified bidding structure, so it just inherits the existing fee marketss immunity that sort of thing

but like the reason why a lot of the alternative proposals do end up breaking down is basically because of that right because there isn't this kind of one-to-one correspondence between what you pay and how you influence the miners income and whenever those two things are out of sync there's some opportunity to kind of get around this kind of payment.

Tim:
I feel like this is probably something you've thought about Dan and curious see what you think, but could you solve I guess both problems so the problem of like fee burning seems to be kind of an important thing for part of the community and also the other problem you mentioned before of like miners waiting to include transactions

Could you have something where like the percentage of the fee burn kind of goes up similarly like I guess quicker than your bid or something? where you know, if miners included when it's close to lowest bid almost none of the fees burn and if they wait all the way to the end.

I don't know maybe there's 50% fee burn. is that an idea you've kind of thought about?

Vitalik:
I think the problem with that is that would kind of counteract the the whole reason why the burn motivates minors or sorry why increasing fee would motivate the miners to include the transaction, which is that miners are more motivated than who transactions at the later step in the escalator because those transactions actually do pay more to the miner.

23:53
Like I think the design space of burning is that like actually is a pretty constricted design space. The design space basically is you have to set some base fee and that base fee gets burned.

It's strange but they're literally are no alternative to that... and that base fee could be adjusted if you want 1559 style, it could just be set to a constant one gigawei, it could be some non-linear function or what whatever but I think I burning in amount that's only dependent on the amount of gas consumes in a in a block and possibly in previous blocks is like the only way to burn.

Ian:
Seems like if you burned a fraction of the transaction you'd preserve both the

incentive to to process the high speed and and the burning I can't see it well definitely

Vitalik:
the problem is getting around the side with side payments right

so if say I am would pay 10 get 10 gwei and you would get five gwei or it's it's a more marginal thing… Or this time you get five, and then we make a side payment or whatever.

Dan:
But the minor would have to be allowed to process something that had a lower bid than I guess the top well yeah. I guess they always have the ability to pick what they want yeah yeah, so that does give away to bypass the rent yeah.

25:25
Tim:
I think another like really important a bit around 1559 is is the fact that like it introduces two transaction pipes for a while and that basically breaks everything… you know, like and and I'd be curious to Dan get kind of your thoughts on that because you're the only wallet developer in the room, you know, if we went ahead with with 1559, what do you think the impact would be on metamask and and other things you think we can do to mitigate it

26:00
Dan:
Oh yeah part of my motivation for being here is partly that like oh well we're gonna have to implement something in response to this so so better get involved, you know, we haven't done a design exercise on this yet and I think that would be really valuable for any of these proposals while there's two I guess that's just an intensive for us to move over to the new input type as quickly as possible.

26:22
I think that the escalator has a kind of graceful migration path because the current fee structure is just equivalent to a flat escalating bid, so I think that it's kind of a subset of that mechanism but yeah, I you know, either way there's gonna be a new input designed for while it's developers and I think there's an open question of how friendly we can make either one.

26:49
Ansgar:
Could you combine like an 1559 style base fee with the escalator bidding structure?

Dan:
I think tha's what Vitalik was just suggesting… as long as it was a flat fee it probably wouldn't be gameable, so I guess the answer is probably?

Tim:
The TIP becomes the escalator

V: right right yeah, you can definitely make the tip be the escalator. or be an escalator rather hmm.

Tim:
I'd be curious to hear like client developers thoughts on that like, how would like would that change to complexity of it that much and are there any things you can imagine that would break.

Ian:
I don't think it would make it much more complex than it is right now you would need to hold in context the block number when you're calculating the gas price but you already have to hold in context the base fee currently, which also comes from the header so….

I don't think you would add much more complexity than we've already added with 1559.

28:05
Tim:
And this might be like a really bad idea but because 1559 is already like a very large change would it maybe make sense to do it in two phases right?

like 1559 version one you just ship the the tip as is and 1559 version two, you ship the escalator is that simpler or is that actually worse because then it's like breaks metamask and everything twice?

28:32
Dan:
I'm not I'm not totally sure. I I'm partly trying to think through how the the combination would work like whether we could abstract it to the user basically has one escalating TIP you know, one escalating bid where you know, it's actually just prone to the 1559 style base fee or if they would actually have to set two prices that sounds like a lot of needless user complexity, but but there's probably a way of combining them into a single user for a you know, kind of expression.

29:08
And and yeah, yeah, I would be obviously to things that implement but hey, I I care more about keeping users happy.

29:19
Ian:
It’s a good perspective to have.

29:27
Tim:
There’s a lot of potential breaking points with you know either of these proposals, like obviously the UX is a big one. I think there was a lot of concerns with like the block size increase or variability for 1559 and and how that would affect the network

So one thing I kind of wanted to get out of this call from all the people here is

what are the things we kind of need to validate if we want to move this proposal forward?

30:01
I feel like Dan you hinted at one already would just like the wallet designs and maybe getting a feel for what it looks like and getting feedback from the community, but on the technical side of like the implementations of the EIPs. Ian you had kind of started just mentioning like more larger scale testing what do you think that would look like?

30:25
Ian:
Well I think for the most part we need to test a number of the parameters that right now are just kind of hard-coded for example, the per transaction limit, the max gas limit, the the range that we have the transition occurring over, and as an effect of that also the amount of gas that is transition from the legacy pool to the EIP 1559 pool per block.

so all these parameters that are kind of hard coded at this time and, They were somewhat arbitrarily selected or not arbitrarily selected but they’re I don’t think there’s been the proper modeling and testing done to support settling on those parameters.

31:07
Abdel:
Yeah, I agree especially for the hard block limits so Vitalik you mentioned that we could go up we could go higher than the factor of two so we will have to determine the optimum factor for that so I was wondering basically would it be better to express the hard block limit as a factor of the target gas usage to kind of mitigate … to Change this limit basically.

Yeah that definitely seems reasonable and I think even the original EIP 1559 I set a scaling factor instead of setting … yeah so setting a scaling factor is definitely a good idea.

31:58
I mean, I think. In the short term like there's enough kind of I think clients developer kind of uneasiness about making block sizes much bigger that I don't expect a scaling factor of higher than two to be in a political viable, but if if two works for some time and we have a lot of data showing that three or four or five are safe then having an easier path to increasing seems like very smart.

32:25
Abdel:
Okay, so maybe at this testing phase it will be great to have it configurable to avoid the rebuilding clients every time we want to change it.

32:43
Vitalik:
Seems reasonable.

32:52
Ian:
And another question that came up in the magicians thread was the transaction ordering. And whether or not we should place that under consensus and maybe this isn't the time to bring that up, but I'm go ahead and mention them.

Vitalik:
Whether or not we should put what about transaction ordering under consensus?

Ian: the ordering itself?

V: Meaning like enforcing a particular order or?

I: Yeah, exactly.

V: What would the benefits of enforcing an order be?

I: I can't really speak to those myself. I think the supposed benefits would be predictability for contract developers. Just being able to make certain assumptions about the ordering of transactions based on the perimeters.

V: Like what would be an example of an of in ordering a guarantee that would make things nice to developers?

I: I'll have to look at the thread sorry.

V: Right and I think I'm the challenge is um, like the current default behavior is just increasing is just including them in decreasing order of gas price, right? I guess my concern is that if we Start mandating kind of inclusion based on some order based on weird parameters that and instead of things like a front-running being just between a moderately transparent gas auction, it becomes some even weirder thing that's more centralized and opaque and harder to participate in.

34:38
Ian:
That makes sense to me. I was actually opposed to it myself. It was just one of the ideas that was brought up in the magicians for it.

V: That's fair.

34:54
Tim:
So it seems like so far there's like three kinds of big buckets that we need to look at. The first is just like trying to mock this in the UI and see and see you know, what it looks like from a user perspective. The second is trying to understand the escalator versus the base fee and and what would you know, what what are the trade-offs maybe a bit better?

35:16
And the third is modeling I guess all of these parameters. For the third bit around the modeling as you have Barnabay, I hope I pronounced it right on the call. I think you had worked on the the model for 15 the original like Python notebook for 1559.

35:36
I'm curious if you have any thoughts on my modeling these kind of constants and in the EIP and what that would require?

Barnabe:
Hey, thanks. Yeah. I do want to look a bit more into modeling and actually reply to Dan in the ethresearch thread.

So it's more about like figuring out how do we write down like being incentives of the miners or the users? How do we write on like the parameters? On the transaction level but also let's go on to the protocol level like the block size. These kinds of things.

I'm less concerned about actually parameters like the gas limit. I think it it's more down to like technical things than it is down to incentives.

36:21
I may be wrong on this but I think this is the case. So yeah, my work only is more like to try get like a more game theoretical model and make sure that there's no let's say bad incentive, but some people have been speaking of some kind of let's say Blackmailing where miners could be saying "well, we don't include any transaction that has less than five gwei in the tip. So just want to make sure that this is not possible.

36:53
Tim:
Got it. So in that case for the more I guess at technical or like, you know in protocol parameters like the per transaction and the the range of the the transition the amount of gas per block... would th best way be to set up like a small test network between say geth and basu and basically try, you know, various ranges and and see what breaks.

37:18
Abdel do you have thoughts on that?

37:25
Abdel:
Mm-hmm. You mean a real test net?

Tim I think at first maybe just like a slide I don't know what you can't qualify by for real right but like not a fork of Ropsten basically no I think I would yeah so for context I think at some point they would make senseto fork a real testnet that has states that has contracts and what not and and get that up and running that maybe an empty test net yeah be better at first it just test these things quickly but I am not sure yeah.

Abdel:
I agree it would be better to have an idea of the optimal parameters when we go through the. Kind of real testnet, yeah.

38:23
Tim:
So. After is there any other kind of big like I'm not sure what to call them but like areas of this change that people want to discuss like so far we have kind of the UX, the tip versus escalator, both the theoretical models and maybe like the setting up a test net.

38:47
Yeah, is there anything else that people feel is like kind of the potential deal breaker on this EIP?

38:58
Vitalik:
Do we want to talk about this question of like basically our we concerned about kind of all miners colluding to try to push the base beat down to get more revenue?

Abdel:
Oh yeah, by the way, I wanted to ask how did you choose the value of the fee smoothing constant?

39:21
Oh this is the um, in my EIP1559 that I linked to... Um I am included a proposal which is basically that like so I have some analysis in there for why I think it's not that miner collusion to push fees down is not that big of a problem you're not that likely to be stable, but if people are still concerned there is a modification that kind of reduces that risk even further which is basically that you would burn you would burn part of the base fee but the other base fee you would kind of redirect to a much larger group of them of miners like the miners of the last that was like 8,000 blocks.

40:16
And I think the calculation there is basically that so first of all like it has to be a larger than something like one or one or two probably even larger than larger than four basically just that because if it's really small then it would still give a substantial amount of revenue to adjacent miners and you could take advantage of that to collude somehow.

And then whatever it should be a larger or much larger like the difference is basically acts as a kind of stabilizer on minor revenue as so do you want to kind of minor revenue to be more volatile or do it or do you want to be more stable? And if you set it to long then I guess just because you have miner churn like miners are not really going to care about it and so it existing miners might still end up colluding to the detriment of future miners.

41:15
So, you know, you don't want it to be one because the that just makes it a whole thing meaningless you don't want it to be a million because then current miners and future miners become misaligned. And so I think realistically like 800 would be okay 1,000 would be okay 8,000 is okay, 65,000 would also be okay. I don't really get matters too much.

Abdel: That makes sense.

Barnabe:
You’re talking about the size of the pool right not the multiplicative constant of the basically yeah?

Vitalik:
I’m talking about so in my proposal or amendment proposal where basically I put half the EIP 1559 burn into a pool and then every block you take one over n of that pool and you give that one over n to the miner. Like basically house like how large should that potion then be?

42:10
Right Thank you.

42:24
Tim:
Oh are there any other big topics people want to bring up?

Abdel: Jjust a last comment about that. I was wondering on the other hand wouldn’t have miners incentive to make the base fee high if it is a fixed ratio?,

Vitalik:
So it does so if the portion of the base fee that was given to miners what higher than one then yes, they would be able to collude to abuse it.

42:57
But like remember that the amount like the base fee like in the status quo that amount like if we’re talking about analyzing the case for miners colludes, so we just treat them as a collective right

So in the status quo kind of 100% of the “base fee” goes to the miners, in the original 1559 its zero, and and in this amendment like it would be 50. Yeah, yeah. So it’s something between anyway, yeah, I see.

43:39
Tim:
Any other big topics people we should bring up?

43:46
Ansgar:
just very briefly I wanted to clarify for my own understanding and because one of the three topics we mentioned earlier… Escalator just to like see if I basically understood that correctly summary it seems like basically 1559 and escalator are fairly orthogonal where you could like basically do both both or either or combine them as as you would want.

44:13
So you can bet you the basically like, Split the discussion there or is there anyone who disagree with that characterization?

44:26
Dan:
This is the first call where I'd heard the consideration of combining them and I definitely personally need to think more about how that could combine. I heard the one suggestion of it being added as the tip parameter that could work it's also possible that escalator could just work with the moving base fee.

44:42
I think both of those are to me totally new ideas from this call and I'd have to think through.

Abdel:
I am wondering if we can buy them we lose one advantage of the escalator proposal. because it will become a broken change as well and you also kind of promoted that it will be backward compatible right so…

Ian:
Yeah, I'll second that certainly. I think the fact that it's doesn't break and it's completely backwards compatible is one of the more promising aspects of that yeah yeah it actually what's what's the whatever wins out wins out, you know, you leave the choice up to the users.

Tim:
Questions or clarifications?

45:49
So if not yeah I looking at my desk here just like four main things I think we need to figure out:

so UX mocks with this looks like

how tip and escalator and base fee kind of all, you know could interact together and and maybe like more clearly thinking true that the tradeoff is getting some community feedback on that

looking at the in protocol parameters and and probably doing that by setting up an empty for a new testament between geth and besu who and then looking at the extra protocol incentives with Barnaby you were talking about.

46:26
First of all, this is that seem about right the people?

And the second of all who wants to do what? I guess. Yeah. I think yeah, it would be really valuable to make progress on on all these fronts. I understand, you know, everyone here has a day job and and most of those day jobs are not working on the EIP 1559, but yeah if people want to volunteer for different efforts, I think it could make sense to try and move those forward and then maybe have a call like in a month or something that the share progress is that some reasonable.

47:05
Abdel:
Yeah, I can interrupt the test net report it will be great if we could sync together, Ian, to be there yeah yeah to see how we can first we had to make the parameters configurable and then because currently it's not part of any release on geth, so it's fine because I just built from the source code of the PR but just to think when things are already or not to test the,

Ian: Yeah, absolutely yeah that's that sounds good to me.

47:40
I can certainly commit to keeping up with any changes that need to be done that PR.

47:49
Okay.

47:57
Abdel:
By the way, one of the issues I had with the PR I don't know if it is fixed or not but there is no support for eth 65 protocol so I don't know if you merged the master branch on that yet,

48:30
Ian: yeah I I need to rebase that branch...

Abdel: good luck!

[haha]

Dan:
I had talked with some of our designers about doing some wire frames for these different algorithms. I think we're willing to give a limited amount of effort to that but maybe it would be even more beneficial to like, I don't know if it's crazy to do like a design contest and we can throw it out to the ecosystem and just see what the wider community comes up with.

49:11
If we could pick a few judges from this call or something that might be an interesting. Thing to explore.

Abdel:
Do you think it would be possible to have a kind of abstraction to make a single design for both proposals?

49:30
I mean that could be a nice stretch goal… I think they have different enough parameters that yeah, they probably yeah, yeah,

Tim:
I think I I mean if it's like meta masks can contribute even if they're super rough wire frames, you know and kind of the next month. I think that would be extremely valuable because it gets people, you know at first view of what this implies.

We had Ken from the grants team who left but I think maybe we can we can also reach out to parallel to ken and see if like the EF grants team would be willing to do like a fairly small bounty to get some more proposals on the design side.

50:09
Um, yeah, I'm happy to follow up on the bounty, but yeah Dan if you if you could get some like really like simple mocks that that would be super valuable. Yeah, yeah. I will bring this back to the team. I have already the idea.

50:30
Tim:
I guess that leaves us with the whole tip versus escalator at a more technical level. I don't know if anybody kind of wants it to look at that and then the other bit was around the the modeling you have like the extra protocol incentives. Barnabe you said you were kind of already thinking about working on that sorry to put you on the spot there…

Barnabe:
yeah yeah kind of like I'm focused let's say not extra protocol but the like the more economic incentive rather than protocol parameters like the block size or yeah yeah, so yeah, but I do want to make like a comparison between the escalator and EIP 1559, so I'll probably like connect with them as well.

51:13
To us like more questions on this.

Tim:
Yeah, if you were able to have like a comparison of the two I think that would be super valuable as well to help kind of make it a decision here.

51:33
Barnabe
So the comparison would also like someone pointed out I think on the github discussion of this call trying like to get past transaction data and analyze it to kind of understand like what the patterns are. I think any comparison will be easier to make if we also kind of are able to predict how often we see spikes how often like do we see the kind of black swan events like what can we expect as the normal conditions so it's something I kind of left open and that we end of my notebook but I think it'd be something like useful to to get an idea like, A like you can't really just like compare them maybe in the abstract like you would make more sense to use actual data.

52:20
Tim:
Yeah, it's basically I guess like a back test almost of how like each of them would have performed throughout like the various phases of the network, right?

Barnaby:
In any way it's a bit hard to do because I mean it was a different mechanism for all this time that you you can still understand like what were the time preferences of a users especially if you have access to things like when are they canceling transactions or when are they bumping the fees so just trying like to get a sense of what is going on on the network like how do people interact with it?

52:58
Tim:
Cool yeah. I think we're pretty much covered then I guess does anyone else on the call have any like initiative or thing they'd like to move forward and the next couple weeks to the months?

53:15
Ian:
That makes me think is there like a general timeline that we have with this?.

Tim:
So my optimistic timeline is I would really love to see something like this on main net by kind of the end of 2020 and the reason for this is there's a lot of work going on on stateless ethereum and that better probably gonna bring a lot of breaking changes and it's my fear is if both these proposals are kind of in R&D at the same time then they end up having to like constantly iterate off each other and and you never get to a final spec for either of them, so if we could kind of,

you know, wrap this up in 2020 and then you’re going to start to see a lot of like I think less stuff go live on the network. That would be like the cleanest separation of concerns. And you know Pegasus we’ve kind of made it a priority to help push that but it’s obviously kind of a broader community effort.

54:11
Well, yeah, that’s kind of the timeline understand of like, you know things change and often take longer than you expect them to especially on Ethereum… That’s kind of what I had in mind.

Thank you.

And I guess you know breaking this down a bit more It means, if you have like a fork going on main nets at the end of the year, you probably need to test nets a month before you need the fork be finalized around Devcon more or less.

54:39
And because this is a big change you probably need a couple months of like actual data on some sort of a forked ropsten or something like that. So I can see it being like the summer being mostly prototyping and getting like some initial concepts out, you know, part of the fall getting some more thorough kind of testing in the wild data to refine some edge cases and then towards the later half of the fallopia goes into the the standard EIP process phases. But that’s my view as kind of a product manager on this so Ian or Abdel might completely disagree that this is possible.

55:22
A&I: It sounds visible. Yeah. I think it’s possible. I’m.
Tim:
Cool and doesn’t make sense to schedule another one of these calls like roughly a month out from now. And and kind of report back then. Yeah does that work for it for people? It’s good to meet. Yes.

55:48
And one last small thing a couple people on Twitter asked if this was livestream obviously it’s not this time around but if we had another call would anybody opposed to it being livestream like the Corps of Dev Calls because I think there is a lot of community enthusiasm about this and it’s it’s good to get to get more eyes on it.

56:08
Good to me. No opposition here.

56:15
Cool. Well, yeah then I guess we can wrap up. Thanks love everyone. We

appreciate really appreciate your taking the time. Thank you. Thanks nice to meet you, all right? Yeah. Thanks.

Micah09/05/2020
I'll wait another day and if I don't hear any arguments for keeping it I'll submit a PR to the EIP to rip out that section.

Micah09/05/2020
Has anyone considered block size as a function of time since previous block?

[11:44]

This potentially could result in incentive for miners to set `block.timestamp` to be closer to reality (right now a significant portion of them set timestamp to be pretty old to minimize the chance of getting rejected due to time synchronization issues). It also has potential to help alleviate the problem of gas price volatility due to long blocks (stuff starts to backup anytime there is a long block, and it clears out on short blocks).

11 May 2020

meridian (Sam Bacha)11/05/2020
I dont think that block.timestamp has a a direction correlation to gas price volatility , and I think block.time should be its own unit of measurement for "time" in the sense that its dependent on previous blocks and not on UTC

timbeiko11/05/2020
Fun progress update: @abdelhamid has gotten a local test network to mine and process blocks

with 1559-style transactions across Geth and Besu
https://twitter.com/PegaSysEng/status/1259919084445204482

PegaSys (@PegaSysEng)

Our team has been hard at work on EIP-1559 Here's a short demo showing #HyperledgerBesu and geth mining and processing 1559 blocks and transaction on a small local

network Expect more from us on this in the coming months
https://t.co/XvAJNwQrn3 https://t.co/ywZY9XtSDG

Twitter

5

4

14 May 2020

timbeiko14/05/2020
Should we have another implementers' call in the next 2-3 weeks? Do people have a preference between the last week of May and the first week of June?

danfinlay14/05/2020
I've created a Kialo board for graphing out the pros & cons for different transaction processing strategies: https://www.kialo.com/what-is-the-best-transaction-fee-model-for-ethereum-to-have-37162?path=37162.0

Kialo

What is the best transaction fee model for Ethereum to have?

Ethereum 1.0 uses the same basic single-price auction type of transaction ordering as Bitcoin, but with the added notion of gas and gas-price per op-code to…

1

15 May 2020

Micah15/05/2020
@danfinlay I think the two being debated solve separate problems and are compatible with each other.

[02:41]

So it feels weird to put them against each other on a Kialo board.

[02:42]

I would be fine with both getting included in the next hard fork, or either, or neither (though, I would prefer both).

danfinlay15/05/2020

@danfinlay I think the two being debated solve separate problems and are compatible with each other.

@Micah I haven't heard anyone explain how they would work in tandem, although one person asked if that might be possible. If you have an idea of how, you should write it down for consideration.

[03:45]

I could imagine "escalator for the tip"? That doesn't seem like a graceful synthesis of the mechanisms to me, as the tip doesn't matter for ordering except within a block, and the bid would escalate *between* blocks.

Micah15/05/2020
@danfinlay Yes, it would be escalator tip.

[03:47]

The idea being that escalator tip would give high guarantee of inclusion within n blocks, even during times of transient congestion.(edited)

danfinlay15/05/2020
The mechanisms in combination have way too much interaction to consider merging as an option without weighing the option of merging independently of merging either option on its own. So yeah, I think it deserves its own Kialo thesis.

Micah15/05/2020
They solve different problems though.

[03:47]

Like a Kialo of whether global warming is better than apples.

danfinlay15/05/2020
By all means, explain what problems you see them as solving.

[03:48]

I see the problem at hand as "the efficient ordering of transactions".

Micah15/05/2020
1559 addresses transient short term variability in gas prices for inclusion, largely caused by variability in block times due to poisson distribution, and partly due to randomness in transaction arrival times.

[03:49]

Oh. I don't think 1559 has anything to do with ordering.

danfinlay15/05/2020
But transaction inclusion is always an act of ordering.

Micah15/05/2020
It more has to do with how you can end up with a 5 second block that is mostly empty followed by a 100 second block that has high gas price due to being overfull is just a matter of luck, yet gas prices will vary wildly between the two.

[03:52]

(at a computer, can be a bit more verbose for a bit before I have to step away again)

[03:54]

If you look at time between blocks historically, while they average around 15 seconds there is huge variability. You'll sometimes end up with 3 blocks in a row that are under 5 seconds followed by a few blocks in a row that are well over 30 seconds. Transaction arrival times are also well distributed, but random (human input), so you can end up with random short term surges and lulls in transaction arrival.

danfinlay15/05/2020
Yes, so you find that when there is even mild congestion, the prices can surge dramatically.

Micah15/05/2020
These two things result in high variability in gas costs to include in "next block". For example, you could have 3 long blocks in a row that coincide with a random surge in transaction arrival that results in those long blocks all being over-full, and then following that you have a lull in transaction arrivals and a few short blocks that completely clears out the pending queue.

danfinlay15/05/2020
I'd call this a sign of very inefficient price discovery when evaluating transaction order. No one knows how long the spike will last, so they have to bid like it might last forever, and overshoot.

Micah15/05/2020
1559 tries to address this by making it so individual blocks are ellastic, so when you have a transient surge the block can expand to fill it, without users having to wait for a lull/short blocks to clear the queue.

[03:56]

With 1559, *transient* surges (less than 2x, which is incredibly common) would just result in block size elasticity being exercised and *transient* lulls would result in the block size ellasticity "regenerating".

[03:57]

This does *nothing* to address medium/long term congestion, which is what the escalator algorithm addresses.

[03:57]

And the escalator algorithm does almost nothing to address short-term (e.g., 1 block) surges.(edited)

[03:59]

1559 is saying, "we want *average* block size to be X, but it is OK if some are bigger and some are smaller to help smooth out surges and lulls caused by randomness in the process". Escalator is saying, "we want to automate the currently highly inefficient and manual process of gas repricing to give regular users a high guarantee of inclusion within n time while minimizing the amount they pay for that guarantee as much as possible".

danfinlay15/05/2020
So would you say you expect the BASEFEE to tend towards zero?

Micah15/05/2020
I *do* think both of these are good things, and I would advocate for Escalator Algorithm if proposed as an EIP.

[04:00]

No, and in fact it is game theoretically impossible for it to go to 0 I believe.

danfinlay15/05/2020
Of course to go to, but I mean to tend.

Micah15/05/2020
I expect the BASEFEE to reach an equillibrium at "the demand for block space is equal to supply of block space over some period of time".(edited)

danfinlay15/05/2020
What's weird to me is the fee not dropping fast when demand disappears.

[04:01]

I mean, that's one weird thing.

[04:01]

I mean, it's the other side of "you can't bid higher to get included faster during congestion"

Micah15/05/2020
You mean if there is a medium term demand surge that clears up? Like when those crazy airdrop contracts flood the network with transactions for 8 hours?

danfinlay15/05/2020
Right, or these ponzis nowadays.

1

[04:01]

It's just kinda constant.

[04:02]

I mean maybe I have to be clear, I have a base assumption of continuous demand of low-value transactions, I expect congestion to be the norm.

Micah15/05/2020
What is the current decay rate?

danfinlay15/05/2020
1/8 is the proposed in EIP

Micah15/05/2020
Congestion cannot be the norm under 1559.

[04:03]

1/8 drop per block?

danfinlay15/05/2020
I mean part of the whole point is that it can't move very fast. If it could move too fast, then suggesting a MAX_FEE to the user gets scary, you have to over-estimate a bunch.

[04:03]

1/8 drop/gain if under/over full.

Micah15/05/2020
That seems incredibly fast, like if the ponzi suddenly stops hammering the network it'll take ~8 blocks to half?

danfinlay15/05/2020
It's still weird to charge people for demand that isn't present.

[04:04]

But I do see a bit of what you mean.

Micah15/05/2020
I think this is similar to how WalMart has fixed prices, despite demand being irregular.

[04:05]

People *like* predictability, even it means *sometimes* paying more.

[04:06]

Walmart *could* vary their prices for produce (for example) based on transient demand surges (e.g., lots of people want tomatos today, so they are more expensive than tomorrow when a new batch comes in and randomly coincides with a reduced demand for tomatos).

[04:06]

Instead, they just factor in the cost of throwing out rotten tomatos and people *prefer* that.

danfinlay15/05/2020
But also it could lead to *sometimes* having high unpredictability: If the price surges abruptly outside your MAXFEE, there's not really any way to tell the user when it might go through without resubmission.

Micah15/05/2020
So everyone pays slightly more for tomatos in exchange for the more predictable shopping experience.

[04:06]

But also it could lead to sometimes having high unpredictability: If the price surges abruptly outside your MAXFEE, there's not really any way to tell the user when it might go through without resubmission.

This is exactly why I support elevator algorithm as well.

[04:07]

1559 does *not* deal with sudden large and medium/long term surges in demand well at all, and in fact the situation is no better than right now.

[04:07]

Elevator algorithm IMO *does* deal with that situation quite well.

[04:08]

Elevator algorithm also has a nice little bonus of incentivizing miners to more accurately set block timestamps.

[04:08]

Right now, miners routinely set block timestamps as much as 30 seconds behind reality (for selfish reasons).

danfinlay15/05/2020
Hmm... I guess if the tip starts very small, inconsequential in the next-block-or-two case (1559 ideal), then it can escalate to approach the person's MAX_BID, which gives miners incentive to prioritize those transactions *during* a spike...

2

[04:09]

You might have a point there. I need to eat, but I'll think about that more.

1

1

2

Micah15/05/2020
I expect tip will be around 1 nanoeth to 2 nanoeth in "normal" times.

[04:10]

Since that is the rough opportunity cost of transaction inclusion to miners due to increased uncle risks.

[04:11]

That goes down as block reward goes down, or up as block reward goes up. It also goes down as network connectivity/bandwith goes up and block processing goes down (lower chance of uncling).

danfinlay15/05/2020
That's a very empirical prediction, I'll try to give you credit if that all happens and you end up being right.

Micah15/05/2020
During congestion though, you are totally right, tip once again becomes "preference" for inclusion during times of congestion (where 1559 can't handle the increase).

[04:13]

@vbuterin wrote some blog posts a while back that tell miners how to calculate the optimal min gas price as a function of a bunch of things.

[04:13]

Many of the parameters are "fuzzy" (like network connectivity, block processing speed of *other* miners, etc.)

[04:13]

"A while back" being like 2015 or something.

danfinlay15/05/2020
Sounds very plausible, I may have read it back then.

Micah15/05/2020
But the function is well founded IMO (even if the inputs are fuzzy).

danfinlay15/05/2020
Ok: Dinner! Thanks for the chat.

1

1

barnabe15/05/2020
On merging escalator and EIP1559 the simplest implementation is likely for users to pick either of the two, with the condition that if their starting bid in the escalator + the current tip is lower than the basefee they can't get in. Whether this is a good idea or not I am not sure, as it seems to me this violates the intuition that your bid "rises". It's like being on an escalator in a flood and the water level is rising faster than you escalate.

[15:36]

So another implementation is escalating the tip of 1559. The metaphor is the escalator is on an inflatable raft, and your tip rises following the escalator, while the water level/basefee rises and decreases the whole escalator. This seems to have very unpredictible dynamics.(edited)

barnabe15/05/2020
If the price surges abruptly outside your MAXFEE, there's not really any way to tell the user when it might go through without resubmission.

I don't see how this is not a problem with the escalator either, once your maxbid is reached and you haven't been included, if you believe you went too low the first time around then you'll resubmit with a higher bid. If you really cared about next block inclusion though likely your maxbid should be close to your willingness to pay, otherwise you really have no guarantee to get in at a price that is feasible to you. If you didn't care about next block inclusion but lowballed your maxbid, then you should have the first time around set a higher maxbid and a longer escalator duration to soften the slope(edited)

danfinlay15/05/2020
Yeah, I think one of the user-experience challenges for either is simply asking for users to declare the most they'd pay. EIP 1559 seems to suggest most of the parameters being auto-set, so MAXBID being maybe `BASEFEE *2` (maybe I'm misinterpreting), so part of my issue with 1559 may actually just be the notion of presenting it as a "no thinking needed" user experience. Actually asking "the most you'd pay" is a kindof wrenching question to press on someone, although it is a very simple mere number in terms of inputs.

[17:19]

So another implementation is escalating the tip of 1559. The metaphor is the escalator is on an inflatable raft, and your tip rises following the escalator, while the water level/basefee rises and decreases the whole escalator. This seems to have very unpredictible dynamics.

I was thinking about this more, and the nice thing is that even if the raft floats up, you'd still be limited by your MAX_FEE, so while you may escalate faster than a linear slope, that will only happen to reflect fees spiking abruptly. I think that could be an acceptable behavior, it's not a very far stretch of the expectations already being set.

James Hancock15/05/2020
The protocal setting the BaseFee, and the user deciding the Max miner tip?

[17:23]

And the tip being an escalator so the user gets the best rate that a miner will give them?

barnabe15/05/2020
my initial impression is that 1559 is "no thinking needed" in most cases, i.e., not much congestion, while I find it harder to reason about the escalator parameter without thinking back to the first price auction case, where you are left to your own devices or wallet defaults to find a "baseline" from which your bids can start. I imagine the 1559 UX in high congestion cases could be "hey, the basefee is currently much higher than in the last 500 blocks, are you willing to pay basefee + x to be probably included in the next y blocks?" there is also some degree of certainty inherent to how the basefee evolves, for instance, given basefee b at some block t, you are guaranteed to be still "in the race" (but maybe not included) y blocks later as long as you set maxfee >= b * (9/8)^y. The wallet could show you that curve before you validate the transaction

James Hancock15/05/2020
Some thoughts I have had since my last debate about 1559.

[18:26]

I think we have in our community tied the concept that high fees are connected with high congestion.

[18:27]

It is totally possible under 1559 to have High Fees and Low congestion.

[18:28]

As in the Fee is high but there is still plenty of space in the block. Perhaps because it sits around the target gas usage.

[18:29]

A user may be waiting for lower fees, but it should be pretty unusual to wait for congestion, i.e. wait for space.

[18:32]

Minutes of high congestion should happen, but as the Base_fee rises congestion will let up.

[18:33]

Currently if the blocks are full the gas prices go up. I feel this has been ingrained a bit in our subconscious as a community.

[18:34]

This doesn't have to be the case.

James Hancock15/05/2020
imo, People waiting for full blocks to subside is different then people waiting for fees to go down.

[18:48]

That is a core difference between the old Fee structure and 1559.

Ivan Martinez (Kiwi)15/05/2020
Lots of people argue the miner tip will create the same effect though

James Hancock15/05/2020
There is a difference between stopped at a stoplight and being stuck in traffic. Still they are both still stopping.(edited)

[20:14]

Create the same affect *for a short time I believe is a significant difference.(edited)

[20:18]

Congestion for two hours, versus congestion for two minutes.

2

17 May 2020

moody 🦄17/05/2020
I would like to bring up the idea that we implement escalator for gas prices before EIP 1559, regardless of the future of EIP 1559 or EIP 1559 + escalating tips, for the simple reason that we need transaction 'deadlines' and escalating gas prices *yesterday* to handle the UX problems of today. Today's UX consists of constantly resubmitting and cancelling transactions because you never know when your transaction will be dropped from the mempool or included, and it's completely non-deterministic which TX wins. The escalator gas price algorithm is very simple, easily understood, and I think universally agreed upon, and significantly improves the current wallet UX. I remember it today because I just got into a situation where I had several old TXs queued up in MetaMask because I had set a very low gas price for an old TX, and cancelling an old one doesn't do anything about the newer ones I sent after it, so I have to send additional

transactions just to cancel the old ones with higher nonce than the bad one. The UX for setting gas prices very urgently needs help. If my TX didn't get in 3 days ago I probably want to resubmit with a higher gas price or cancel it, but there's nothing the wallet can do about it without my approval.

18 May 2020

Micah18/05/2020
@moody 🦄 I would support adding escalator. I don't think 1559 should be delayed for it though. Whichever is ready first goes in first, whichever is ready second goes in second.(edited)

[01:28]

How is the escalator EIP coming along?(edited)

barnabe18/05/2020
@moody 🦄 I would support adding escalator. I don't think 1559 should be delayed for it though. Whichever is ready first goes in first, whichever is ready second goes in second.

@Micah
how does that work? they are not obviously commutative, nor even composable, as we discussed above

Micah18/05/2020
I thought the conclusion of our discussion was that they are?

[02:27]

I don't see any conflict between them, nor any dependency.(edited)

[02:28]

Side note: Would be good to get the escalator EIP merged with an EIP number so it shows up on EIPs website.

James Hancock18/05/2020

Base _fee is set by the protocol. Youd have an escalator with one step

Micah18/05/2020
1559 still lets the transaction specify fee. The escalator effects the *transaction*.

[02:30]

If 1559 goes in first then Escalator would need to be reworded a bit, but it is a minor change.

[02:35]

Right now a transaction has a `gasPrice`. With 1559 it would have `gasPremium, maxFee`. Escalator pre-1559 would make `gasPrice` be a function of time. Escalator post-1559 would make `gasPremium` and `maxFee` be a function of time.

moody 🦄 18/05/2020
My thought is that escalator is much smaller in scope, so if 1559 ends up ready before escalator (even if escalator work only started officially tomorrow), it means we didn't commit very much time to the escalator implementation. I think we *should* prioritize escalator since it seems uncontroversially better than current state and is likely to have an immediate impact towards improving the user experience. Unfortunately I can't commit to helping with the work so I'm only asking as an ethereum user

Micah18/05/2020

Step 1 is getting escalator EIP merged as draft. (edited)

[02:37]

Unfortunately I can't commit to helping with the work so I'm only asking as an ethereum user

Sadly, this is my current state with regards to several EIPs as well.

[02:38]

I have been thinking lately of quitting all other work and just going full-time on getting some EIPs pushed through... but the work sounds... painful.

barnabe18/05/2020
Right now a transaction has a `gasPrice`. With 1559 it would have `gasPremium, maxFee`. Escalator pre-1559 would make `gasPrice` be a function of time. Escalator post-1559 would make `gasPremium` and `maxFee` be a function of time.

@Micah why would maxfee be a function of time? this seems like the one parameter that should be fixed, to commit to a transaction that never costs more than some limit I set

Micah18/05/2020
The "function of time" can include a max.

barnabe18/05/2020

it's possible the combination works, but I am simply saying that it's not obvious, and several designs exist too

[02:41]

i

Micah18/05/2020
Example function: `x > now + 60 ? max_fee : base_fee + block.timestamp - base_time`

barnabe18/05/2020
i'd like to write these designs down just so that we have a starting point

moody 🦄18/05/2020
I left a couple of comments on the EIP, I don't know the steps to getting an EIP merged though

abdelhamid18/05/2020
@moody 🦄 I would support adding escalator. I don't think 1559 should be delayed for it though. Whichever is ready first goes in first, whichever is ready second goes in second.

@Micah EIP-1559 implementation is ready in Geth and Besu. All parameters have been made configurable so that we can try different configurations and find the optimal parameters. What are the next steps to setup a testnet ?

Micah18/05/2020

@abdelhamid From me *personally* the next steps for 1559 is to accept my PR to change it.

[10:33]

I'm *against* 1559 right now (but not for the reason most people are). I have an open PR to resolve the issue I have with it, but I cannot merge it without agreement from 1559 authors.

[10:34]

Once that is "dealt with" (either by silencing me, convincing me I'm wrong, or moving forward despite my disagreement) then I think getting it into OpenEthereum and Nethermind would be valuable, or if not creating a new Geth + Besu testnet.

[10:34]

Though, I think the "right" next step is probably to put it up for discussion during an All Core Devs call.

James Hancock18/05/2020
Where is your PR micah?

Micah18/05/2020
https://github.com/ethereum/EIPs/pull/2635 @James Hancock

GitHub

Removes block size and transaction size riders from EIP 1559. by Mi...

This EIP previously changed the mechanism for calculating block size from a miner-voted mechanism to a hard-coded mechanism. This change is not necessary for the implementation of EIP1559, so this...

James Hancock18/05/2020
Thanks

1

[13:53]

My concern with giving miners a lever to change the target gas size is miners are very motivated to manipulate the base_fee to as close to zero as possible.

[13:54]

I do agree with the rider concern. Making the decision explicit would also help avoid confusion.

Micah18/05/2020
@James Hancock Do you believe there is some profitable attack vector that can be exploited by miners to drive the base_fee down?(edited)

[14:03]

Prior to submitting that PR I asked around to see if anyone had a legitimate reason why block size needs to not be controlled by miners in order for 1559 to work and couldn't find an answer, but a solid attack vector would certainly be a valid reason for me to drop my PR and my concern about riders in 1559.(edited)

barnabe18/05/2020
some (very) preliminary notes here. not addressing any potential incentives issue nor whether any of these designs actually achieve what we like them to do, but these are already three different ways to implement a combination (there are certainly more)
https://github.com/ethereum/rig/blob/master/eip1559/combination.md

GitHub

ethereum/rig

Robust Incentives Group. Contribute to ethereum/rig development by creating an account on GitHub.

[14:11]

from my point of view the "floating escalator" seems the most reasonable (it can default to eip 1559-style txs, for one), though I worry that it might not be so easy to analyse given the interaction with basefee

Micah18/05/2020
Why does 1559 have a separate FEECAP and GAS_PREMIUM?

[14:18]

It feels entirely unnecessary.

[14:18]

The one exception is when you want `block.coinbase` to pay the fee, but I don't feel like solving for that is necessary or wort the added overhead of having separate variables.

[14:19]

It seems to me that users should specify `FEECAP` (which we can just call `GASPRICE`), and `BASEFEE` is burned and remainder goes to the miner.

barnabe18/05/2020
apart from the burning of basefee, that's not very different from a first price auction from the point of view of the user

[14:21]

the point of having two parameters is that your bid is indexed on basefee

[14:21]

and you have an upper bound in case basefee goes too high

Micah18/05/2020
Hmm, I see. The idea is that you can add padding in for base fee movements *without* having to pay more to miners than they normally ask.

[14:22]

"I'm willing to pay any BASE_FEE because I trust the algos to not try to screw me, but I'll only pay 1 neth per gas to miners, because I know that is the going rate and if I offer more they'll demand more."(edited)

barnabe18/05/2020
kinda!

[14:24]

i would say more "i am willing to pay basefee because that's the going rate, given the demand"

[14:24]

but yes the idea is that the premium can be relatively constant

[14:24]

that is, if we don't believe miners collude to push the basefee to very low levels

Micah18/05/2020
I didn't previously realize those two were separate. I now better understand why it is "not obvious" how escalator and 1559 interact.

[14:25]

(I still think it is very possible for them to interact).

[14:26]

I do not believe there is a profitable strategy to push the base fee down, even for miners colluding. The one exception *may* be if *all* miners collude.

[14:26]

But that seems unrealistic.

[14:28]

@barnabe What is the blue line in your charts?

[14:28]

And what is the purple line?

abdelhamid18/05/2020

@abdelhamid From me *personally* the next steps for 1559 is to accept my PR to change it.

@Micah Oh yeah i forgot those changes. If your PR is accepted we will have to update the Geth and Besu implementations accordingly.

[14:31]

Though, I think the "right" next step is probably to put it up for discussion during an All Core Devs call.

@Micah Yeah. Or the next EIP-1559 implementers call.

Micah18/05/2020
I would hope that it is an easy change.

[14:32]

Should just be a matter of rolling back the deletion of miner voting code more or less.

[14:32]

And some minor changes to the "slow rollout" code.

abdelhamid18/05/2020
Yeah indeed. Should not be a huge chunk of work

barnabe18/05/2020
@barnabe What is the blue line in your charts?

@Micah Sorry yeah i added the legend in a new commit purple is the gasprice, blue is the basefee

James Hancock18/05/2020
https://twitter.com/owocki/status/1262440325848502272?s=21

Ξwoki, GITer of Coins (@owocki)

Current Ethereum Network Gas Price (Y Axis) vs Confirmation Time (x axis) ⛽ In a sane world

this would be a diagonal line from top left to bottom right, but it's not. I'm pretty tired of fielding support requests about this. Long EIP 1559!

Twitter

19 May 2020

Matthew G19/05/2020
Hi everyone, Matthew from Unstoppable here. Read the proposal, looking forward to following along.

1

Micah19/05/2020
I have been thinking about escalator + 1559. It *feels* like 1559 already has a mechanism in place for "escalating" the base fee (in the form of the `feemax`). So I feel like the escalator only really needs to push the `gas_premium` up with time.

Micah19/05/2020
The `feecap` can be set to the max from the start since basefee is a "trustless" algorithm (economically) and not subject to miner exploitation like `gas_premium` is (miners don't directly control it).(edited)

barnabe19/05/2020
So I feel like the escalator only really needs to push the `gas_premium` up with time.

@Micah this is also why i find the "floating escalator" to be the most reasonable among the three hybrids in the document (https://github.com/ethereum/rig/blob/master/eip1559/combination.md)

[11:18]

until your premium is bounded by your maxfee, the premium is what sets you "ahead of the pack" compared to other users who have transactions in the pool

[11:19]

so escalating the premium with the floating escalator retains the intuition of "climbing above" the rest

[11:20]

but then again, this is just a design, I didn't really think through yet whether this introduces bad incentives

[11:20]

nor whether this is desirable/useful

[11:23]

it does introduce a bunch more parameters even though a few can be set to reasonable defaults, so even from a UX perspective it's not a clear win

Micah19/05/2020
Assuming the function is shared by everyone, then it should just be 2 additional inputs?

[12:28]

Hmm, actually maybe 3 inputs... (start, end, max_premium).

timbeiko19/05/2020
@Micah @barnabe @danfinlay @James Hancock @Ian Norden @moody 🦄 @Ivan Martinez (Kiwi) @vbuterin @meridian (Sam Bacha) @gichiba @afdudley @thenateway would 16:00 UTC this Thursday work for an implementers' call or is it too last minute? We could do next Thursday too, but that will be the day before AllCoreDevs, so I think there may be value to having it this week and maybe coming up with something to bring to ACD the week after to get more client feedback.

Micah19/05/2020
I can't make that, but I'm not much of a voice/video chat person anyway so I don't recommend trying to time it where I can make it.

[14:39]

My current feedback on 1559 is in the form of a PR that is waiting to be merged or argued

against.

timbeiko19/05/2020
Got it, I'll make sure to add your PR as a discussion point for the agenda to get everyone's thoughts.

1

Ivan Martinez (Kiwi)19/05/2020
Good by me!

[14:40]

@Micah mind linking your PR?

Micah19/05/2020
https://github.com/ethereum/EIPs/pull/2635

GitHub

Removes block size and transaction size riders from EIP 1559. by Mi...

This EIP previously changed the mechanism for calculating block size from a miner-voted mechanism to a hard-coded mechanism. This change is not necessary for the implementation of EIP1559, so this...

Ivan Martinez (Kiwi)19/05/2020

> thank you

1

Micah19/05/2020
I also have a concern that the during the migration period, large transactions will no longer be possible for a time.

[14:42]

For example, if a contract requires 8M gas to deploy, there will be a period of time during the migration where it is *impossible* to deploy that contract.

[14:43]

Probably something worth discussing, though I don't have an actual solution for that.

barnabe19/05/2020
@Micah @barnabe @danfinlay @James Hancock @Ian Norden @moody 🦄 @Ivan Martinez (Kiwi) @vbuterin @meridian (Sam Bacha) @gichiba @afdudley @thenateway would 16:00 UTC this Thursday work for an implementers' call or is it too last minute? We could do next Thursday too, but that will be the day before AllCoreDevs, so I think there may be value to having it this week and maybe coming up with something to bring to ACD the week after to get more client feedback.

@timbeiko sure!

[14:54]

Hmm, actually maybe 3 inputs... (start, end, max_premium).

@Micah when working it out I found at least 4 parameters were necessary to fully define the space. if you can't specify the starting premium then you also cannot reproduce 1559-style transactions, nor start from a little over the basefee. but then yes either the maxpremium or the maxfee or both makes sense

[14:56]

For example, if a contract requires 8M gas to deploy, there will be a period of time during the migration where it is *impossible* to deploy that contract.

@Micah is that because the gas pool in a block is split between 1559 txs and regular transactions?

moody 🦄19/05/2020
@timbeiko i can't make that but I've expressed my opinion a few times here already, perhaps @itamarl can join with another perspective

timbeiko19/05/2020
Thanks @moody 🦄 ! I asked in AllCoreDevs if anyone has a preference. Will wait till tomorrow to book the final time, but we can tentatively aim for Thursday this week.

Micah19/05/2020
is that because the gas pool in a block is split between 1559 txs and regular transactions?

Yeah.

[23:02]

@barnabe

20 May 2020

timbeiko20/05/2020
EIP 1559 implementers' call #2 is scheduled for next week Thursday May 28, 14:00 UTC
Agenda: https://github.com/ethereum/pm/issues/174

GitHub

EIP-1559 Implementers' Call 2 · Issue #174 · ethereum/pm

EIP-1559 Implementers&#39; Call 2 The purpose of this call is for various teams/individuals working on aspects of EIP-1559 to discuss the implementation plan and next steps to move the EIP forw...

6

23 May 2020

aliatiia23/05/2020
@Micah what do you mean by "riders" ?

Micah23/05/2020
It is a term used in governance that refers to something added to a piece of legislation that doesn't have anything to do with the legislation it is being added to.(edited)

[02:09]

It is very common in dysfunctional governance systems (like the US), where someone will propose a bill about say healthcare, and then someone else will agree to vote yes on the bill if it also includes a few paragraphs about how yacht taxes should be lower (or something equally unrelated).

1

[02:09]

@aliatiia

aliatiia23/05/2020
Ah I remember those. Agree they should be prevented generally (equivalently, modularity can be a requirement for EIPs generally). I don't think it was intentional in 1559 though.

Micah23/05/2020
I suspect not as well, but I do want to prevent it from becoming "accepted practice", so when I see it (accident or not) I try to address it.(edited)

1

26 May 2020

timbeikoYesterday at 16:44
EIP 1559 implementers' call #2 is scheduled in 2 days Thursday May 28, 14:00 UTC Agenda: https://github.com/ethereum/pm/issues/174

GitHub

EIP-1559 Implementers' Call 2 · Issue #174 · ethereum/pm

EIP-1559 Implementers&#39; Call 2 The purpose of this call is for various teams/individuals working on aspects of EIP-1559 to discuss the implementation plan and next steps to move the EIP forw...