EIP 1559 and Fee Structure

Why a fee market

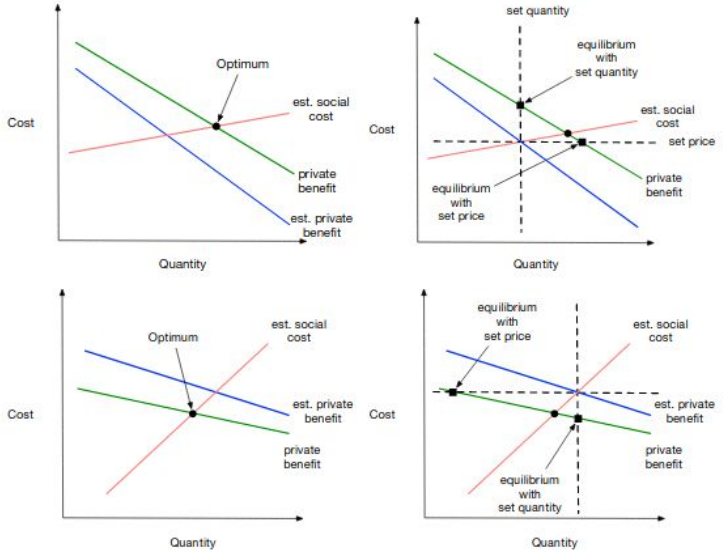
- Transactions have some private benefit to users
- Transactions have some social cost to the network
  - Computing/bandwidth costs
  - Increase to hardware requirements
  - Increase to centralization risks
- Goal: accept transactions only if benefit > cost

Why a fee market

- Standard Pigovian solution: charge social cost to the user, user will only send if their private benefit exceeds this cost
- Problem: social cost is hard to measure, and we only have very vague intuitions
- Social cost even changes over time!
    - eg. if there's 10x more users, social cost of each tx goes up 10x
- Zero is not an acceptable approximation; there's near-infinite very-low-value uses of the blockchain (eg. backing up your cat videos)

Prices vs quantity limits

- Under perfect information, prices and quantity caps are equivalent regulatory tools
- Under imperfect information…
  - If per-item social costs are relatively fixed, setting a price is better
  - If per-item social costs are very nonlinear (increasing), setting a quantity cap is better

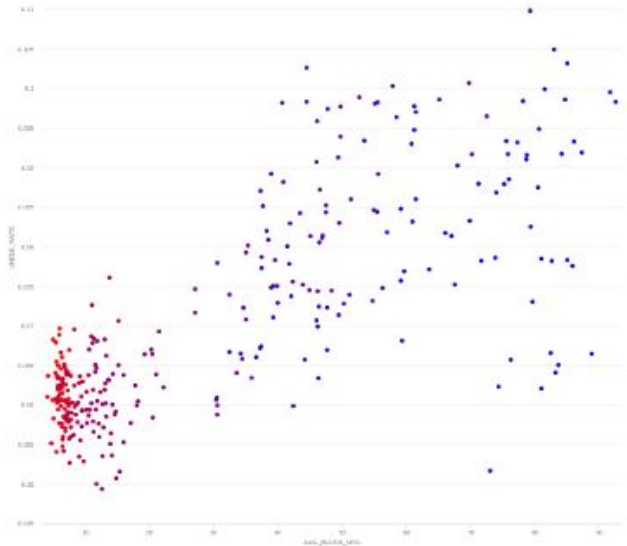


Figures (a)–(b) show the first scenario in which it's better to set a price. Figures (c)–(d) show the second scenario where it's better to set a quantity.

In the blockchain case.....

- There's a common intuition that
  - Costs depend on maximum load, not average load
  - Costs are low but quickly increase approaching a maximum
- But IMO neither claim is true!

Uncle rate as a function of block size is linear



Relationship between block gas limit and uncle rate, Dec 2016 to Sep 2017

Block size is bounded by things other than physical capability of nodes handling the load

- Safety factor against DoS attacks
- Lower centralization risks if uncle rate is lower
- Long-run storage size growth
- Ease of syncing recent history
- Ease of running a node (% CPU)

Claim: acceptable short-run limit is higher than acceptable long-run limit

- In Ethereum, 12.5m gas blocks take ~200 ms to process on average
- Would users *really* be harmed if blocks would instead take sometimes ~50 ms and sometimes ~350 ms? Not really
- Even applies to DoS attacks: there is no sharp cutoff of “19 seconds ok, 20 seconds bad”; each additional second adds somewhat to the cost of an attack

Proposal: compromise between fixed-fee and fixed-limit approach with hybrid strategy

- Have a fixed limit
- Have a fixed target at half the limit
- Allow usage to float around the target, have a “base fee” that adjusts so that usage in the long run approximates target*

```
new_basefee = old_basefee * (1 + (usage - target) / k)
```

*Technically you want `new_basefee = old_basefee * e^(usage - target)/k`, but the first-order approximation is good enough in practice

Proposal: compromise between fixed-fee and fixed-limit approach with hybrid strategy

- Transactions specify two parameters
  - `MINER_TIP`: extra payment to a miner to compensate for tx processing expenses and marginal uncle risk from including the transaction
  - `FEE_CAP`: maximum total fee (including base fee and miner tip) that they are willing to pay
- Backwards compatibility strategy: for old-style transactions, set `MINER_TIP = FEE_CAP = old_fee`

Immediate benefit: quicker inclusion

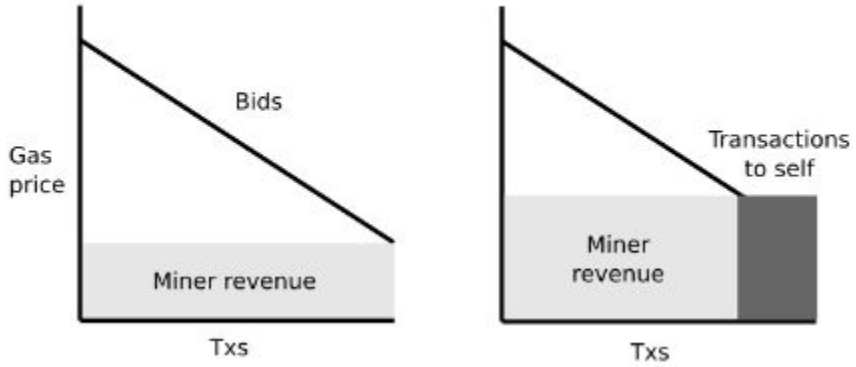
- In any mature blockchain, blocks are full, and usage is volatile
- If demand during some block is overfull, those transactions normally have to wait
- This waiting is terrible for UX, and is socially unproductive (the world gains nothing from the fact that those txs were included 1 min later instead of 1 min earlier)
- The hybrid strategy removes the waiting except during exceptionally serious usage spikes

First-price auction inefficiency

- Current tx inclusion is a *first-price auction*: if you win, you pay what you bid
- Game theorists/auction theorists dislike first-price auctions, because they have no clean optimal strategy
    - Imagine you are willing to pay up to $1 for an item. How much do you bid? Well, you have to guess what everyone else is bidding: if everyone else is only bidding $0.05, you can safely bid $0.06, but what if there's uncertainty?

First-price auction inefficiency

- A common alternative is *second-price auctions*, but those are vulnerable to manipulation and collusion of various kinds.



The adaptive fee approach

- In the normal case (demand < limit), everyone just pays the basefee plus a small tip to compensate miners for computing expense and marginal uncle risk
- In the limiting case (sudden usage spikes), it degrades to a first-price auction as before

What to do with the basefee?

- Can’t give it to the miner - otherwise that’s economically equivalent to the BASEFEE being zero (miners could even give a portion back to the tx sender)
- Option 1: put it into a long-term miner pool
  - Eg. each block withdraw 1/10000 of the pool and give it to the current miner
  - Target stable issuance
- Option 2: burn it
  - Goal: target fixed security level, variable issuance
  - In “windfall” situations (eg. Ethereum’s recent fee spike), the network as a whole benefits instead of simply leading to an unneeded temporary spike in the security level
- Either approach reduces instability inherent in fee-dominant blockchains

Other benefits

- Simplicity of wallet development (no/less need for complex fee estimation strategies)
- Privacy (less need to choose fees means less opportunity to reveal bits of info about yourself)

Is it exploitable?

- Could miners somehow exploit the system to reduce the basefee so they get more tips? Let's see...
- If a single miner (or a <50%* collusion) does not include a transaction, the next miners will, so the effect on the long-run base fee is zero
- They *could* not include transactions with a tip below some value, to push senders to increase their tip to get included faster. But this strategy has low effectiveness: participants suffer the full cost of not including transactions but only get part of the benefit of miner tips going up
- If a >50% collusion constantly keeps blocks on average slightly less than half full, they can push the base fee down to zero. But…..

*Technically 46.87%; use the exponential formula mentioned previously to get an exact 50%

Instability of attack via pools

- Compare selfish mining
- Selfish mining is good for participants in the attacking pool, bad for participants in other pools
- Hence, a selfish-mining pool could potentially quickly grow to 51%+, at which point it can censor at will for even more profit
- Here, an attack is good for participants in the attacking pool, but *even better* for participants in other pools
- Hence, a fee market attacking pool would quickly *lose* members (!!)
- See also: https://medium.com/@MicahZoltu/eip-1559-51-attacks-should-you-live-in-fear-d817be3759dc

Where are we at?

- Ethereum: development in progress, have a testnet
- Filecoin: actively running on their testnet