DYNAMIC TRANSACTION FEE MECHANISM DESIGN

Mallesh Pai
Max Resnick

SBC 2024

SPECIAL MECHANISMS GROUP

USERS MAY BE WILLING TO WAIT

- Mainstream Transaction Fee Mechanism makes a strong assumption. . .

SPECIAL MECHANISMS GROUP

USERS MAY BE WILLING TO WAIT

- Mainstream Transaction Fee Mechanism makes a strong assumption. . .
- It assumes that users are not willing to delay inclusion in exchange for lower fees.

USERS MAY BE WILLING TO WAIT

- Mainstream Transaction Fee Mechanism makes a strong assumption. . .
- It assumes that users are not willing to delay inclusion in exchange for lower fees.
- Users in the classical models only care about inclusion in the next block (see e.g. Roughgarden (2021), Chung and Shi (2023) . . .)

SOME TRANSACTIONS ARE TIME SENSITIVE

- Dex trades

SOME TRANSACTIONS ARE TIME SENSITIVE

- Dex trades
- Payments that involve the exchange of a physical good (coffee shop)

SOME TRANSACTIONS ARE TIME SENSITIVE

- Dex trades
- Payments that involve the exchange of a physical good (coffee shop)
- Actions in real time games

SOME ARE NOT

- Simple transfers

SPECIAL MECHANISMS GROUP

SOME ARE NOT

- Simple transfers
- L2 posting

SPECIAL MECHANISMS GROUP

SOME ARE NOT

- Simple transfers
- L2 posting
- Contract deployment

SPECIAL MECHANISMS GROUP

THIS PAPER

Analyze transaction fee mechanism design in a dynamic context where users face an explicit tradeoff between faster inclusion and lower fees.

Formally, we adapt the model and queueing techniques of Huberman, Leshno, and Moallemi (2021) to consider different block arrival processes (PoW vs PoS) and fee mechanisms (e.g. EIP 1559).

MAIN RESULTS

- Proof of stake networks have lower congestion and fees than proof of work networks with the same demand, throughput, and expected blocktime.
- EIP 1559 price does not converge to the first price fee market price on average. Time sensitive users pay more, time insensitive users are delayed longer.

SPECIAL MECHANISMS GROUP

Transactions arrive at random. The number of new arrivals r in any given interval of time v is distributed according to a Poisson distribution with parameter λ:

$$A(r; v) = \exp^{-\lambda v} \frac{(\lambda v)^r}{r!}.$$

Each transaction is associated with a unique user with type $\theta = (\theta_v, \theta_c)$—here θ_v is their willingness to pay for a transaction, whereas their delay cost per unit time is θ_c. The net payoff of a user of type θ whose transaction is included after delay W for a price (bid) of b is:

$$U(W, b, \theta) = \theta_v - \theta_c W - b.$$

We assume that $\theta_c \sim F[0, \bar{c}]$.

Transactions are served at intervals in batches of at most K transactions (blocks).

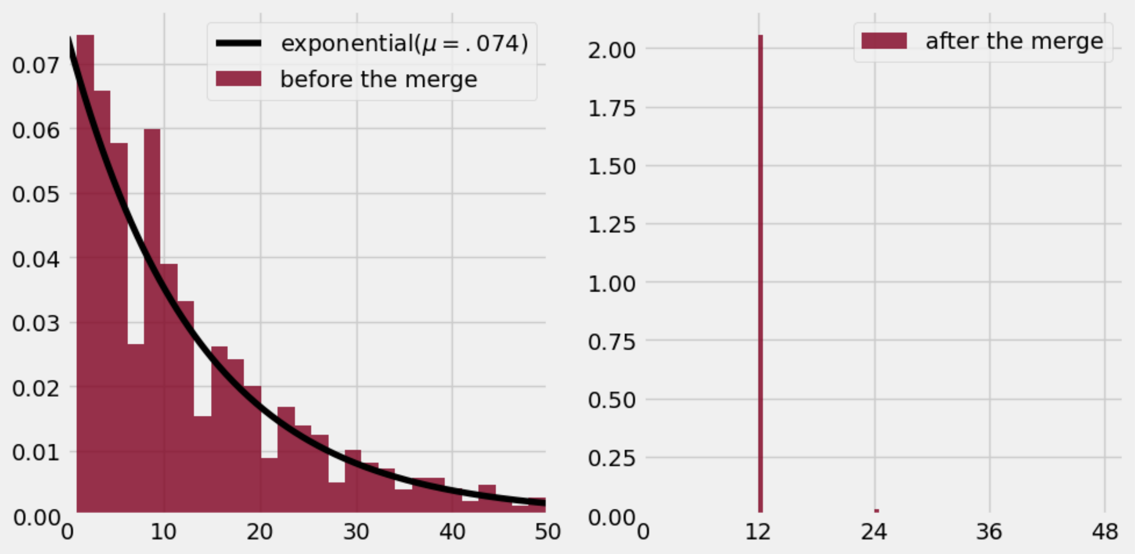
We assume that the intervals of time v between blocks are independent of each other and have the same probability distribution B, which has density $dB(v)$ equal to a χ^2-distribution with an even number $2p$ degrees of freedom, i.e.,

$$dB(v) = \frac{\mu^p}{\Gamma(p)} v^{p-1} e^{-\mu v} dv.$$

$$dB(v) = \frac{\mu^p}{\Gamma(p)} v^{p-1} e^{-\mu v} dv.$$

- exponential (PoW): $p = 1$
- deterministic (PoS): $\lim_{p,u \to \infty}$ keeping the ratio of p and u constant.

BLOCK ARRIVALS



Pai, Resnick (2024): Dynamic Transaction Fee Mechanism Design

PRIORITY MECHANISM

Theorem

Fix any pay-your-bid transaction fee mechanism such that there exists a steady-state bidding equilibrium $b^\star$ and corresponding delay as a function of bids $W^\star$. Then the following must be true:

1. $W^\star(b^\star(\theta_c))$ *is non-increasing in* θ_c.
2. *The equilibrium bids and delay function must jointly satisfy:*

$$b^\star(\theta_c) = -\theta_c W^\star(b^\star(\theta_c)) + \int_0^{\theta_c} W^\star(b^\star(c))dc + b^\star(0). \tag{1}$$

SPECIAL MECHANISMS GROUP

PRIORITY MECHANISM CONT.

Corollary

Suppose the transaction fee mechanism is a pay-your-bid mechanism. Suppose further in a priority queue, the wait time for a transaction measured in blocks is $W_K(\hat{\rho})$ given effective load $\hat{\rho}$ of higher priority transactions. Assuming all users participate (i.e. do not take the outside option, equivalently that θ_v is high enough), we have that:

$$b^\star(\theta_c) = -\theta_c W_K\left(\frac{\lambda(1-F(\theta_c))}{K\mu}\right) + \int_0^{\theta_c} W_K\left(\frac{\lambda(1-F(c))}{K\mu}\right) dc$$

EXPONENTIAL

Theorem (Huberman, Leshno, Moallemi)

Exponential equilibrium wait times and bids are given by:

$$W_K^E(\hat{\rho}) = \frac{1}{\mu} \frac{1}{(1-z_0)(1+K\hat{\rho}-(K+1)z_0^K)} \tag{2}$$

$$b^\star(\theta_c) = -\theta_c W_2^E\left(\frac{\lambda(1-F(\theta_c))}{K\mu}\right) + \int_0^{\theta_c} W_2^E\left(\frac{\lambda(1-F(c))}{K\mu}\right) dc \tag{3}$$

SPECIAL MECHANISMS GROUP

DETERMINISTIC

Theorem

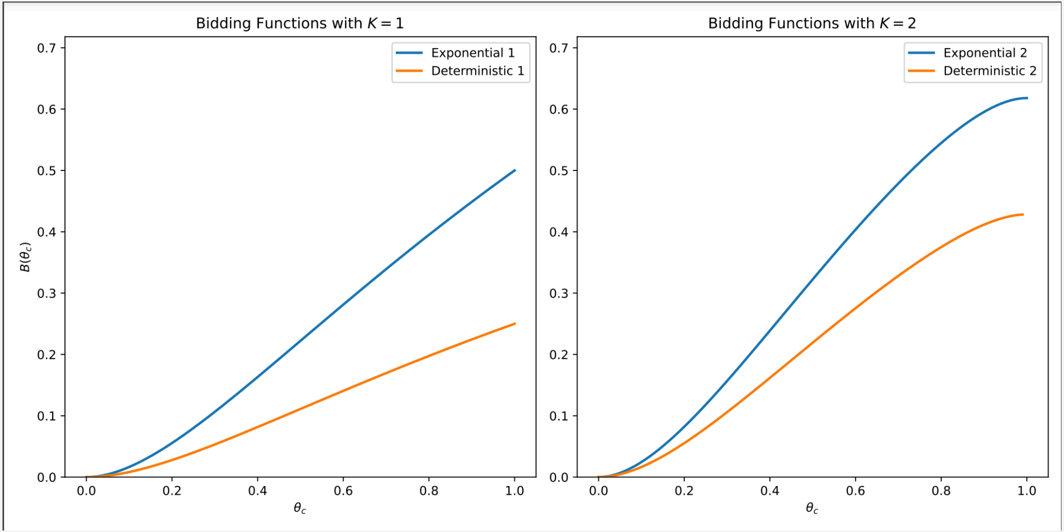
Deterministic equilibrium wait times and equilibrium bids are given by:

$$\mu W_2^D(\hat{\rho}) = \frac{1}{2(1-\rho)^2} + \frac{1}{(1-z_1)^2}\frac{dz_1}{d\hat{\rho}},$$

$$b^\star(\theta_c) = -\theta_c W_2^D\left(\frac{\lambda(1-F(\theta_c))}{K\mu}\right) + \int_0^{\theta_c} W_2^D\left(\frac{\lambda(1-F(c))}{K\mu}\right) dc \qquad (4)$$

SPECIAL MECHANISMS GROUP

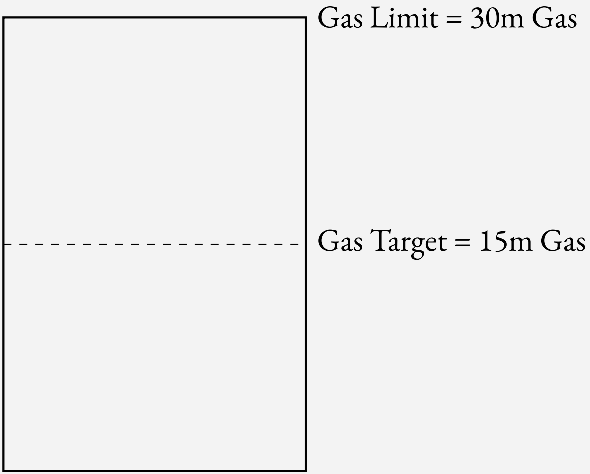
POS > POW

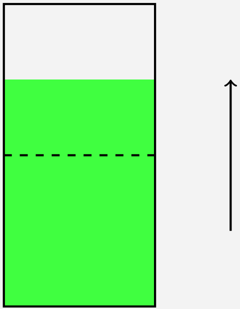


SPECIAL MECHANISMS GROUP

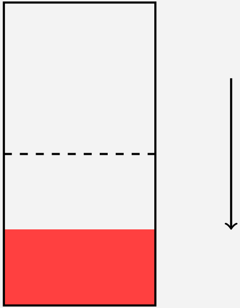
Pai, Resnick (2024): Dynamic Transaction Fee Mechanism Design

1559 EXPLANATION



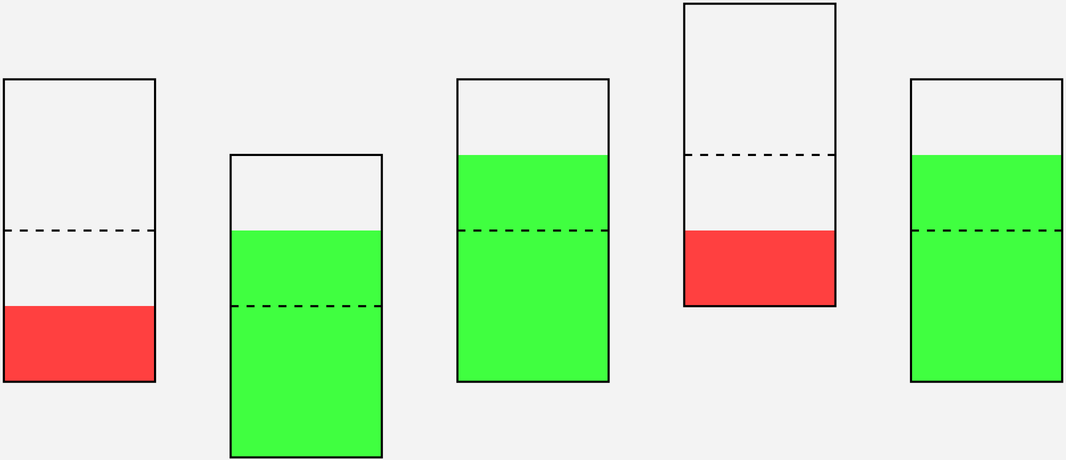


$$r_{t+1} = r_t \times \exp\left\{\left(\frac{k_t}{K/2} - 1\right)\ln(1+\delta)\right\}$$



$$r_{t+1} = r_t \times \exp\left\{\left(\frac{k_t}{K/2} - 1\right)\ln(1+\delta)\right\}$$

Pai, Resnick (2024): Dynamic Transaction Fee Mechanism Design



SPECIAL MECHANISMS GROUP

EVENTUALLY THIS CONVERGES TO THE PRIORITY MECHANISM RIGHT?

Empirically, Ethereum does use 15mn gas/block on average over long windows (e.g., a few hours and longer)

Does that mean EIP-1559 as currently instantiated finds the “right price”?

Current folk intuition: since we are using the desired amount of gas, we are pricing appropriately

A PROBLEM WITH THIS INTUITION

Consider a simplified model where exactly 15mn gas arrives every period:

- Exactly half, i.e., 7.5mn of this demand is patient ($\theta_t = 0$)
- The remaining 7.5mn gas is impatient and suffers a disutility of 1$ per block they wait.

In this model, the following 2 pricing rules both use 15mn gas on average per period:

1. Price of inclusion is 0: All 15mn gas transactions included immediately for free.
2. Price of inclusion is 0.99$ for immediate inclusion upon arrival, 0$ for inclusion with a 1 block delay: Impatient transactions pay 0.99 and get included immediately, every patient transaction gets included in the next block.

SPECIAL MECHANISMS GROUP

A PROBLEM (CONTINUED)

Both of these pricing rules use 15mn gas on average

However, the revenues from these pricing rules are different, and so is the average user welfare/ delay

A similar problem applies to EIP-1559 with strategic users:

- The price charged can go up due to randomness rather than change in the underlying demand rate
- Time sensitive users will still pay this higher price to get immediate inclusion/ avoid delay
- Patient users will get delayed

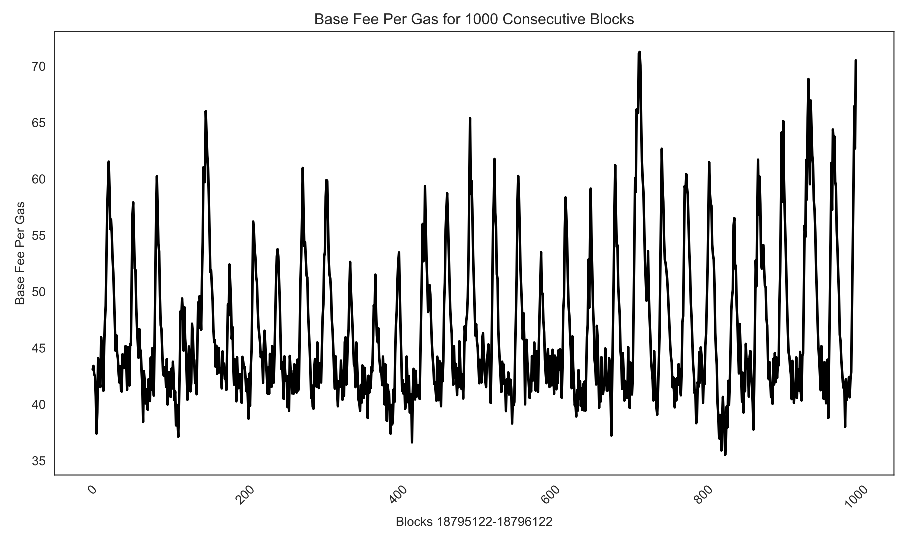
DYNAMIC MONOPOLY PRICING

Put differently, a monopoly seller can extract more from users by charging a high price and excluding low WTP buyers

In a dynamic setting, delay is also a form of exclusion

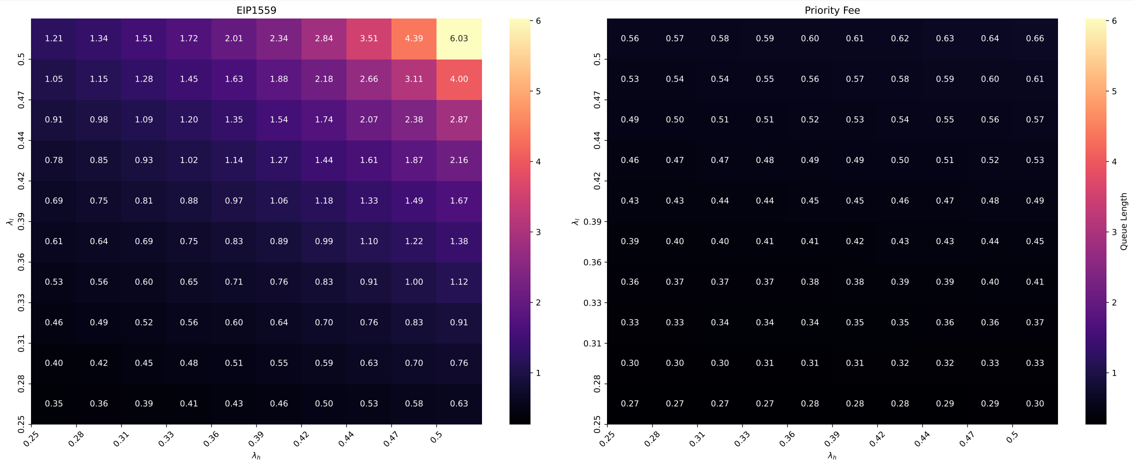
For efficient pricing, the base fee needs to not just use 15mn gas on average, but also do it in the lowest variance way possible

A PROBLEMATIC EXAMPLE



We consider a simplified version of 1559 where there are two prices, high and low and transactions are either high or low willingness to pay.

1559 SIMULATION RESULTS



SPECIAL MECHANISMS GROUP

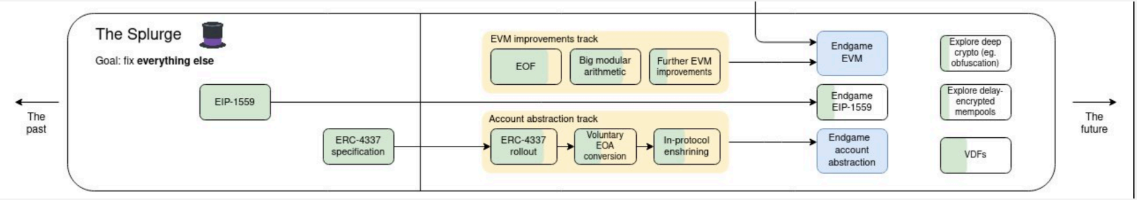
Potuz
@potuz_eth

Ethereum should not price resources in order to accrue value. Ethereum prices resources in order to not be Dosed. We will always have the minimum price that nodes can handle on the target hardware

11:44 AM · Aug 8, 2024 · **808** Views

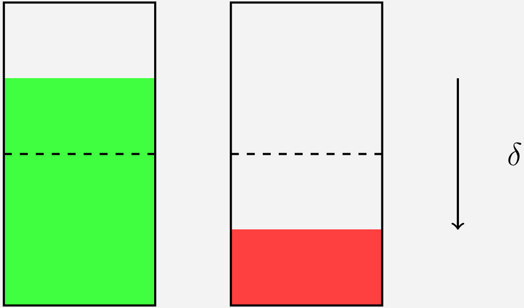
1 2 13

EIP 1559 ENDGAME

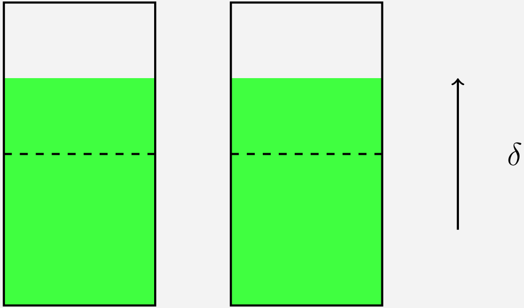


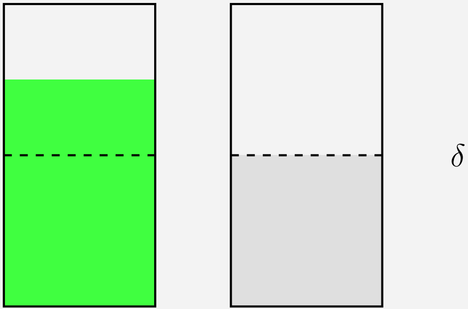
SPECIAL MECHANISMS GROUP

Pai, Resnick (2024): Dynamic Transaction Fee Mechanism Design



SPECIAL MECHANISMS GROUP


δ


δ

PROPOSED RULE

python Copy

```python
def update_delta(delta, last_update, gas_used, gas_target, delta_prime):
    utilization_ratio = gas_used / gas_target
    adjustment = delta_prime * (utilization_ratio - 0.5)

    if last_update == "increase":
        if utilization_ratio < 0.5:
            delta -= adjustment
        else:
            delta += adjustment
    elif last_update == "decrease":
        if utilization_ratio < 0.5:
            delta += adjustment
        else:
            delta -= adjustment

    # Ensure delta stays within reasonable bounds
    delta = max(0.01, min(0.25, delta))  # These bounds can be adjusted

    return delta

def update_base_fee(old_base_fee, gas_used, gas_target, delta):
    utilization_ratio = gas_used / gas_target
    fee_change = delta * (utilization_ratio - 1)
    new_base_fee = old_base_fee * (1 + fee_change)
    return new_base_fee
```

WHERE TO FIND US

SPECIAL MECHANISMS GROUP