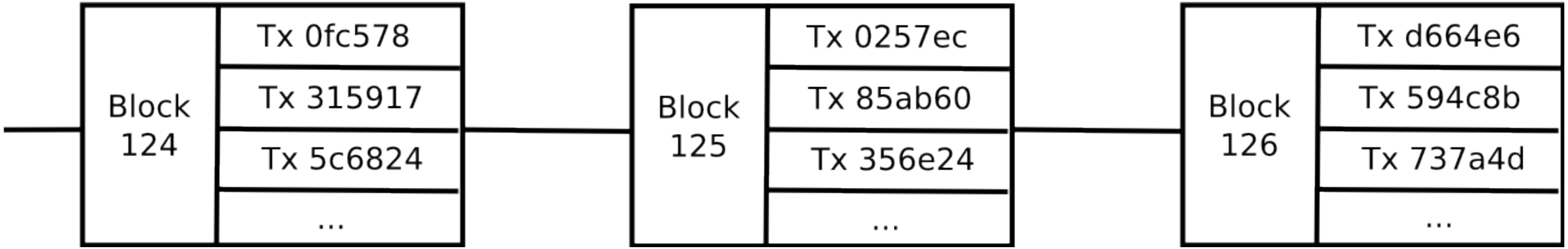
Transaction fee economics

Or, why is the fee/gasprice/rent so darn high?

Fundamentals



- Users send transactions, these transactions get included into blocks
- Each transaction creates:
  - A private benefit to its sender
  - A private cost to the proposer that includes the transaction in a block
  - A social cost to other nodes in the network, including:
    - Proposers/validators, and other users
    - Present users and future users
- The social cost is an externality that is unaccounted for if proposers have full free choice of what to include

Pricing of externalities

- Pigouvian taxes (eg. carbon taxes)
- Cap and trade

Prices vs Quantities (Weitzman 1974)

Prices *vs.* Quantities[1,2]

MARTIN L. WEITZMAN
Massachusetts Institute of Technology

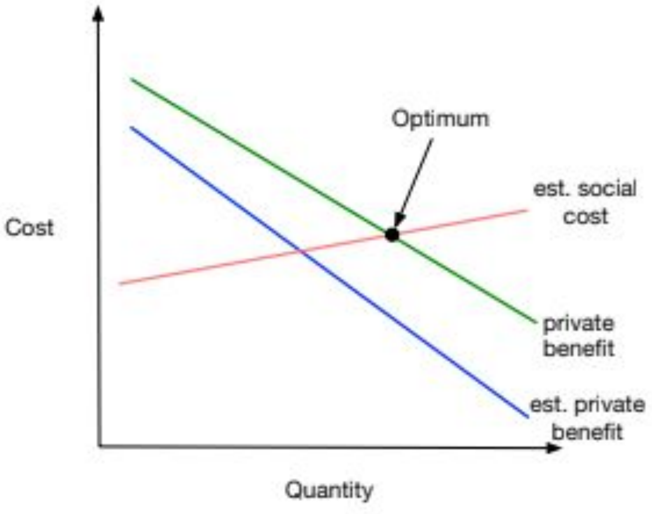
I. INTRODUCTION

The setting for the problem under consideration is a large economic organization or system which in some cases is best thought of as the entire economy. Within this large economic organization resources are allocated by some combination of commands and prices (the exact mixture is inessential) or even by some other unspecified mechanism. The following question arises. For one particular isolated economic variable that needs to be regulated,[3] what is the best way to implement control for the benefit of the organization as a whole? Is it better to directly administer the activity under scrutiny or to fix transfer prices and rely on self-interested profit or utility maximization to achieve the same ends in decentralized fashion? This issue is taken as the prototype problem of central control which is studied in the present paper. There are a great many specific examples which fit nicely into such a framework. One of current interest is the question of whether it would be better to control certain forms of pollution by setting emission standards or by charging the appropriate pollution taxes.

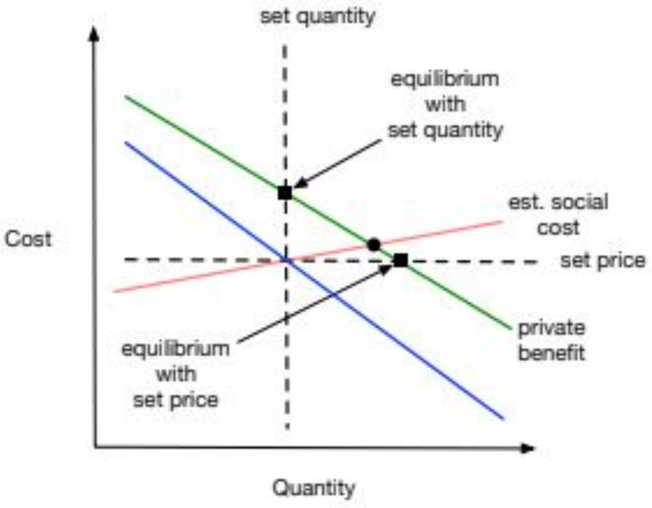
Prices vs Quantities (Weitzman 1974)

- Under perfect information, prices and quantity limits are equivalent
- Under imperfect information, the optimal policy depends on whether the social cost or the private benefit are more quickly increasing
- The *really* optimal policy is rarely a “purely” horizontal or vertical supply curve

Prices vs Quantities (Weitzman 1974)

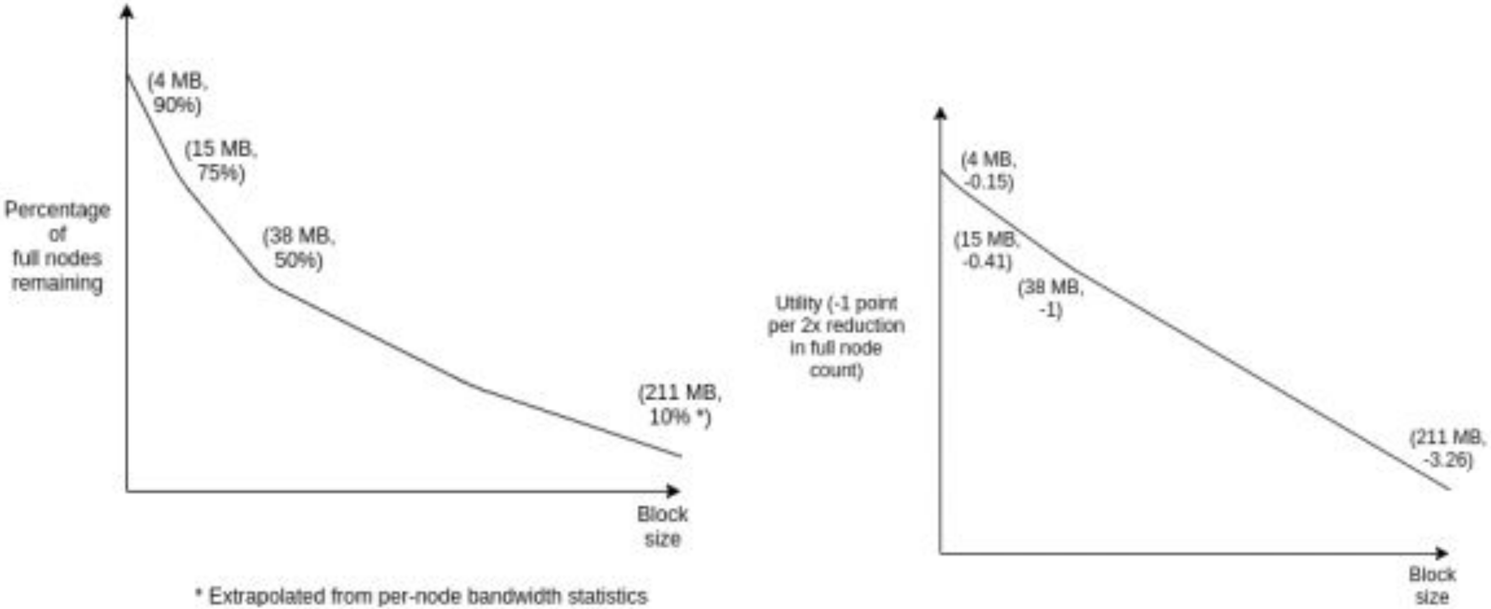


(a)



(b)

Measuring social cost

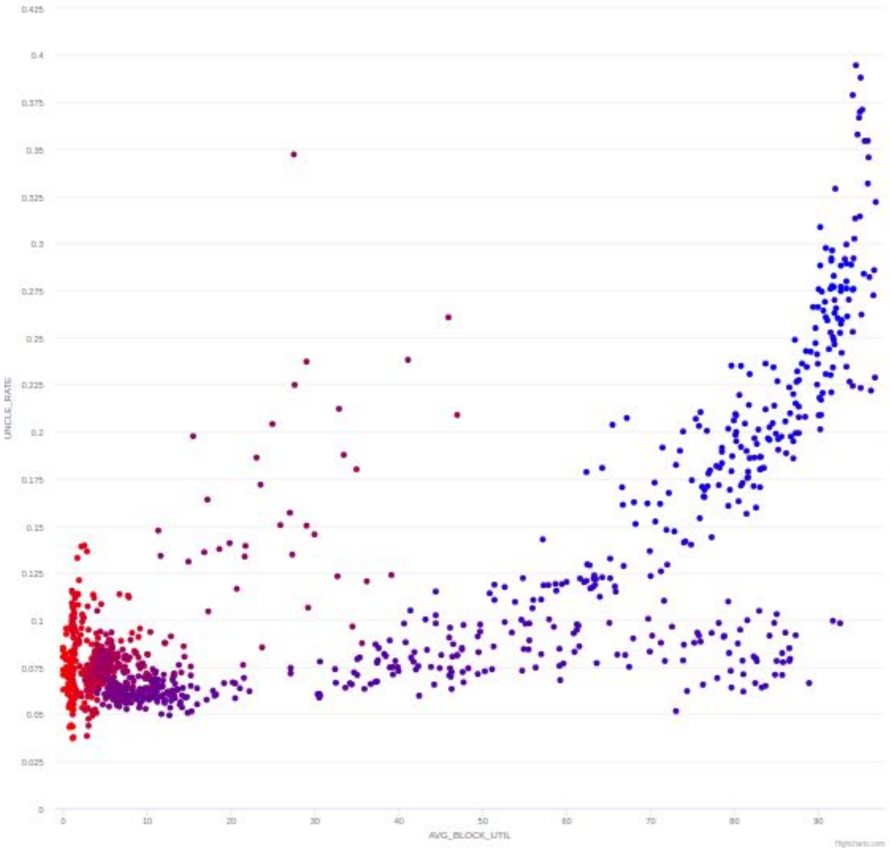


(a) Functional full nodes as function of blocksize from [8]

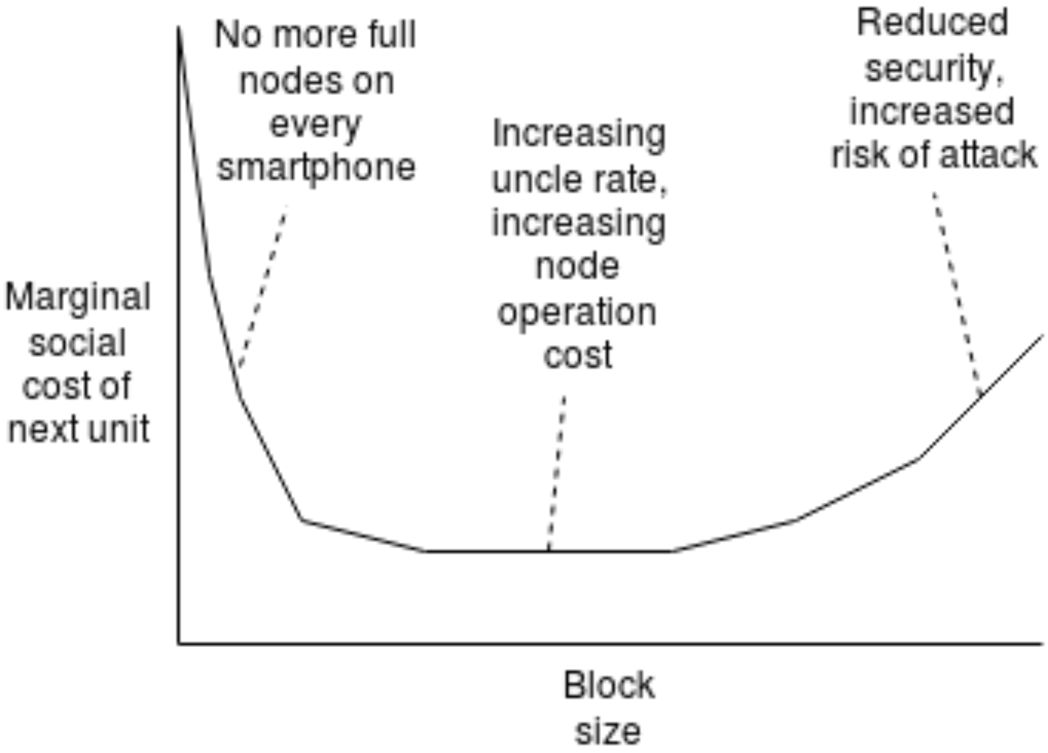
(b) Drop in utility

Figure 3: Estimating the utility

Measuring social cost



Measuring social cost



Auction mechanisms

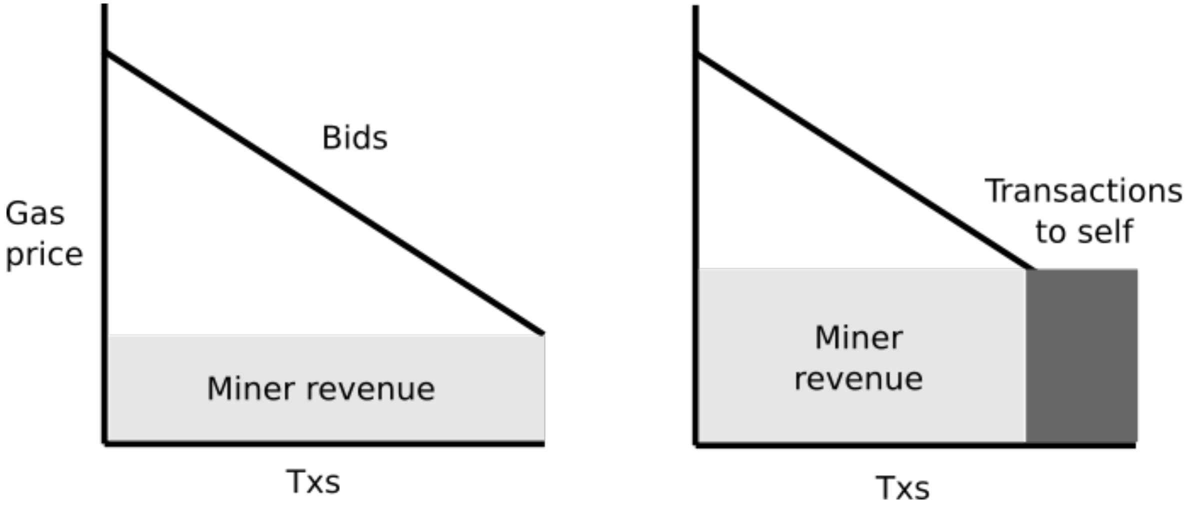
- Current transaction fee mechanism is similar to a **first price auction**
- Everyone submits their bid, auctioneer (block proposer) selects highest bids, everyone pays what they bid
- Problem: this mechanism is not “**truthful**”; bidding optimally requires complex strategies that involve taking into account other miners’ bids

Auction mechanisms

- Professional “geek squad” offers to help to users to submit transactions with fees as low as possible and with maximum reliability

Auction mechanisms

- Possible alternative: second price auction
- Everyone pays the same bid as the lowest bidder
- Strategy for tx senders becomes simple: bid your valuation
- Problem: vulnerable to manipulation by proposer



Hybrid mechanism

- Idea: establish a "minimum fee" that gets **burned**, adjust this fee so that blocks are 50% full
- In the normal case, bidding (minfee + 1) gets you included
- More complex option: allow transaction fees of the form "I bid the minimum fee plus X, up to a maximum of Y, starting from block number B"
    - Allows expressing "I don't care when this tx gets included, I just want it to get included"

Storage fees

- Storage consumption (account balances, contract code, contract state….) is not like other kinds of expenses
- Storage must be held not just by presently online nodes, but also all future nodes

Storage fee issues

1. Storage is underpriced in absolute terms
2. Fees are very volatile day-to-day, but the social cost of storage is not that volatile day-to-day
3. Not enough incentive to clear storage
4. No incentive to clear storage earlier rather than later
5. Incentive to create second-layer markets (eg. GasToken)

Ongoing storage fees

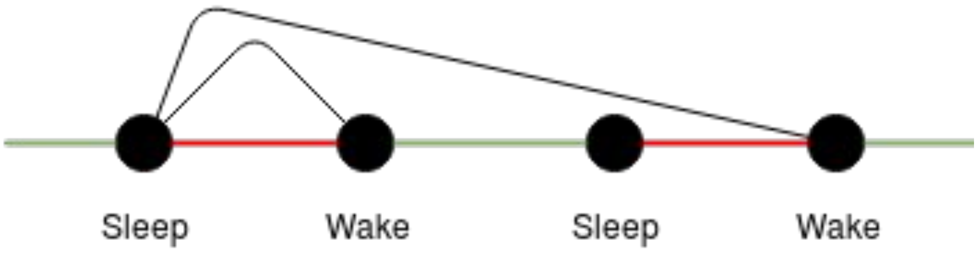
- Simple model: charge N wei per block per byte for every byte stored
- Tradeoff:
  - Fee predictability
  - Total storage size predictability
- What’s the social cost of storage size?
  - Possibly more linear than other social costs

Challenges with ongoing storage fees

- What if an app depends on a contract that disappears?
- How would you write a contract that today would allow anyone to save a storage key in the contract, and in this model would cause the contract to incur a permanent ongoing cost?
- Who pays for upkeep of “public good” contracts?

Mitigation: hibernation

- Idea: when a contract's balance goes below 0, it is deleted from the current state and "hibernated"
- Waking a hibernated contract requires a proof of the contract's previous state, plus proofs that the contract was not already woken up before
- Note: need to watch out for a possible attack:



Implementation

- Fee market changes: theoretically Ethereum 1.0 compatible
- Ongoing storage fees: Ethereum 2.0?
  - May require restructuring of storage mechanisms to work well
- Further research needed:
  - Trying to emergently derive gas / block size limits
    - Eg. non-outsourceable proofs from transaction senders
  - Better estimating social costs and changes from technology
  - Understanding how newer validation technologies (fraud proofs, STARKs, data availability proofs...) would change the analysis