Miner Collusion and the BitCoin Protocol

Alfred Lehar[*]
Haskayne School of Business
University of Calgary

Christine A. Parlour[†]
Haas School of Business
UC Berkeley

October 28, 2019
Preliminary and incomplete
Comments Welcome

Abstract

We document high variation of bitcoin fees, not only over time, but also within blocks. The blockchain rarely runs at capacity, even though fees tend to be higher when blocks are fuller, so miners appear to be leaving 'money on the table.' We present a simple model of price discrimination to explain our results. We note that mining pools facilitate collusive equilibria, and estimate that they have extracted least 200 million USD a year in excess fees.

Keywords: Bitcoin, Transaction Costs, Blockchain, Decentralized Finance

[*] Corresponding author, email: `alfred.lehar@haskayne.ucalgary.ca`, Tel: (403) 220 4567.

[†] email:`parlour@berkeley.edu`. We thank Christina Atanasova and seminar participants at ISE and at IWF-SAS.

1 Introduction

On September 10, 2019 Nasdaq announced a new index, DeFix, to track firms organized along the principle of decentralized finance. The listing is consistent with the growing importance of financial entities that are designed to operate absent a trusted intermediary. While the production function of such entities is clearly novel, it is less obvious if their market conduct is also novel. In this paper, we try and understand whether the first successful decentralized financial product, Bitcoin, has evolved into a system that provides value transfer to consumers as it was promised to. Understanding how well the Bitcoin system works is important because it is the first widely-adopted fully decentralized finance system. It is also unregulated, and so provides us with insights into the sorts of economic structures that arise endogenously.

The original white paper by Nakamoto envisages a system which obviates the need for trusted intermediaries because "competitive miners" record and settle all BitCoin transactions. The code assigns BitCoin as an incentive for miners to do this. The first departure from the zero direct cost settlement, is that spontaneously, over time, consumers have added fees to their transactions to encourage speedy settlement. In this paper, we shed light on the determinants of Bitcoin transaction fees and provide suggestive evidence that miners act in a way consistent with fee maximization, as a monopolist intermediary might. We conclude that even a decentralized finance system can operate non-competitively.

We have three main results. First, we document that the BitCoin blockchain rarely operates at full capacity even in the presence of fees. Our sample comprises all Bitcoin transactions from the genesis block to November 4, 2018. There is no day over this period in which the BlockChain has run at full capacity. Fees began appearing in 2016, and amount to USD 844,537,503.30. The existence of excess capacity, in the presence of fees, suggest miners appear to leave "money on the table." Second, we illustrate that this behavior is observationally equivalent to collusive price-discrimination; mining pools, in as far as they restrict the number of independent players in the system, facilitate such price discrimination. Third, consistent with a conservative interpretation of our model, we define excess fees as those paid above the 10% quantile, which sum to USD 557,568,542.21, or approximately 200 million USD a year since the advent of mining pools.

Although the fees submitted by agents to the BlockChain appear akin to bids in a multi-unit first price private value auction, they arise from a different protocol. In a multiunit first price auction, higher bids are more likely to get the good (in this case, immediate settlement). Therefore, whether a particular bid results in the bidder winning the good depends on all the other bids because the seller commits to award the good to higher bids first. Effectively, the rationing rule is one of strict price priority. By contrast, under the BitCoin protocol, individual miners are not bound by any particular rationing rule, and may choose any set of transactions to work on. Indeed, given the decentralized system miners may observe different pools of potential transactions.

To understand how the miners' ability to choose affects the consumer cost of using the system, we present a simple two type framework that compares bidding behavior in two cases. First, we characterize bids if there are capacity constraints, but miners always execute the highest bidders first. In this case, the bidders who are most desperate for immediate settlement only have to bid slightly higher than the lowest type. In other words, the high types retain some surplus. If miners can pick and choose when and which bids to process, or discriminate, miners can ignore any bids that are less than the high types' maximal valuation. Under this rationing rule, the miners extract the high types' full valuation. The optimal rationing rule obviously depends on various parameters such as how high the high types' valuation is and how many there are of them. This is the familiar price-quantity tradeoff.

The BitCoin protocol is designed so that miners are competitive, and it is well known that in repeated games it is more difficult to sustain collusive equilibria as the number of players increases. This suggests that mining pools provide an economic role besides diversifying risk for individual participants. By acting collectively, each mining pool effectively reduces the set of strategic players and so makes it easier to sustain price-discrimination. This observation is consistent with mining pools' habit of "signing" the blocks that they mine. If they do this, other, unsuccessful, mining pools have a credible way of checking whether the block was mined at full capacity (the pool deviated) or under-capacity. Empirically, we find a relationship between excess fees and mining capacity concentration.

Our paper fits into a small and growing literature on transaction fees in blockchain systems. **?** explain the observed shift from no-fee to fee paying transactions and model the interactions of fee payments and waiting times. While the focus of their empirical analysis is on the time series of average transaction fees our paper documents a huge variation of Bitcoin transaction within blocks analyses the cross section of transaction fees. **?** compare Bitcoin to a traditional payment system and derive closed form solutions for equilibrium fees.

This research on fees fits into a larger body of literature that focuses on the economics and incentives in blockchain ecosystems (among others **?**, **?**, **?**) and the impact on financial markets (e.g. **?** or **?**). **?** analyze the incentives for miners to form pools. Other research focuses on the pricing of crypto-currencies in the market including frictions causing pricing differences (e.g. **?**, **?**, **?**) .

2 The Bitcoin Protocol: Block Capacity, Usage and Fees

A Bitcoin transaction comprises inputs and outputs. Once a transaction is submitted, it is broadcast to the network. Then, each node determines if it is valid (i.e., the Bitcoin have not already been spent elsewhere) and stores it, awaiting execution (also known as mining). Each node keeps an inventory of valid transactions that have not yet been mined called the "mempool." It is important to note that the mempool is not a centralized entity but rather is specific to each node (which typically have different capacity RAM). Each time a node competes to mine a new block, it selects valid transactions from its own mempool to work on. Once a block is mined, the output is then broadcast to the network and all nodes remove the relevant transactions from their own mempools. Output in a mined block consists of a Bitcoin amount that is deposited to an address using a public key.

Miners create blocks by being the first to solve a computational puzzle and are compensated in two ways. First, they receive a mining or block reward, which was set at Ƀ50 initially and is cut in half every 210,000 blocks (roughly every four years). The block reward appears as the first transaction in each block and is also called the coinbase. Second, and more germane to our analysis, participants who want their transaction to be included in a block can offer fees to miners. Fees are offered implicitly as the difference between inputs and outputs. For example, a submitted transaction might call ₿2.2 as input but only assign ₿2.18 as output. Miners retain the difference and pay it to themselves as part of the coinbase if they successfully mine a block.

2.1 The Data and Stylized Facts

Our sample comprises all blocks from the Genesis block (January 3, 2009) to block number 548,684 (Nov 4, 2018) and includes 903,976,764 inputs and 961,308,921 outputs. For each block we observe the coinbase, the inputs and outputs (and hence fees paid to the miners), and the size in bytes of each transaction.

There is a technological limit on the number of transactions that can be processed in a block. In the original design, Satoshi Nakamoto introduced a 1MB limit to Bitcoin blocks in 2010. However, for technical reasons, block size was limited by the number of database locks required to process one (at most 10,000). This limit translated to around 500-750k in serialized bytes, and was forgotten until March 11, 2013, when an upgrade to V0.8.0 with a switch of databases caused an unplanned fork in the blockchain.

After resolving the crisis, the community reached a consensus to remove this unknown limit and a hardfork was scheduled and cleanly activated on May 15, 2013. From then on, the 1 MB limit became the effective limiting factor of the block size for the first time.[1] Subsequently, after much debate, Segregated Witness (Segwit) was implemented on August 24, 2017. This implemented a way to store signatures and scripts outside of the block. The Segwit effectively increased the block size available to process transactions to 4MB under ideal circumstances. The first block exceeding 1MB limit was block 481,947 mined on Aug 25, 2017 with a size of 1032 KB.

Figure 1 shows the evolution of block size over time. Clearly, average block size increases steadily over time. Transaction capacity seems scarce mid 2017 just before the Segwit update.

In spite of the implicit capacity concern that lead to the Segwit, Figure 1 illustrates that there are empty blocks added to the blockchain. From a mining process perspective, there is no difference between mining a full and an empty block (empty spaces are also data). Further, for mining pools there is an advantage to mining a non-empty block as these prevent free riding. To prevent any one pool member from claiming the whole reward, the pool coordinator does not disclose a full list of all transactions that should be included in a candidate block. Instead, pool participants get aggregate transaction information. Once a miner finds the right solution she has an incentive to report it to the pool, as the the pool coordinator's information is required to claim the reward. Empty blocks do not allow the pool coordinator to withhold information and therefore increase the moral hazard problem. (We also note that empty blocks are sometimes mined with malicious intent to slow down transactions in Bitcoin, as in the fight between two

[1] Details of this system change are available at https://en.bitcoin.it/wiki/Block_size_limit_controversy,https://blog.bitmex.com/bitcoins-consensus-forks/

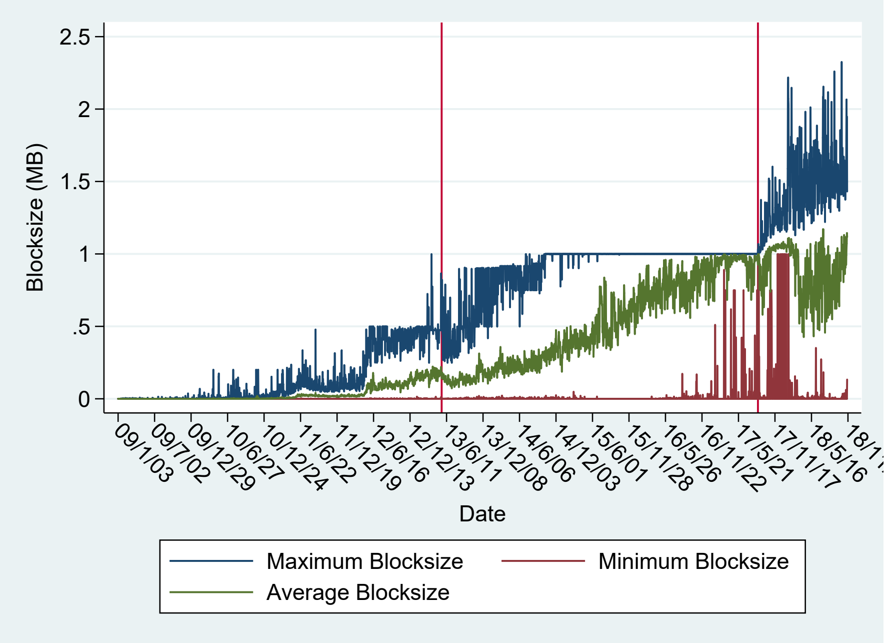

Figure 1. Minimum, average, and maximum daily block size. The red bars mark the safe implementation of the 1MB block size limit and the Segwit update, respectively. Days are defined over UTC.

competing Bitcoin Cash developer groups in November 2018, but these events are rare.[2])

Figure 2 illustrates the fraction of total blocks per day that are mined either completely full or empty. We note that the blockchain has never run at full capacity for a full day.

We note that before Segwit, capacity seems to be constrained. However, on the "worst" day there was excess capacity for 1,444 transactions. Figure 3 illustrates the unused capacity based on the median transaction size.

To verify that space in mined blocks is not simply due to a lack of transaction demand, we collected minute by minute mempool data, which comprises summary statistics on the transactions that have been verified and are waiting to be included in a block.[3] Figure 4 contrasts the daily unused capacity (red line) with the size of the mempool (green line). When median fees (orange line) and bitcoin prices peaked, unused capacity exceeded the size of the mempool, meaning that it would have been easy to include all pending transactions in the empty capacity of the blocks on any one day. The graph also shows that excess capacity is not the result of

[2] On November 15, 2008 the Bitcoin Cash blockchain forked in two parts as competing developer groups fought over the future development of the crypto-currency. Craig Wright, representing one group threatened to halt transactions in the competing fork by constantly adding empty blocks to the competitor's blockchain, therefore increasing processing times for transactions, thus making the other crypto-currency worthless.

[3] We obtained the data from Jochen Hoenicke's website, `https://jochen-hoenicke.de/queue/#0,all`, which contains minute by minute snapshots of the mempool. Transactions are grouped into fee buckets based on sat/byte.

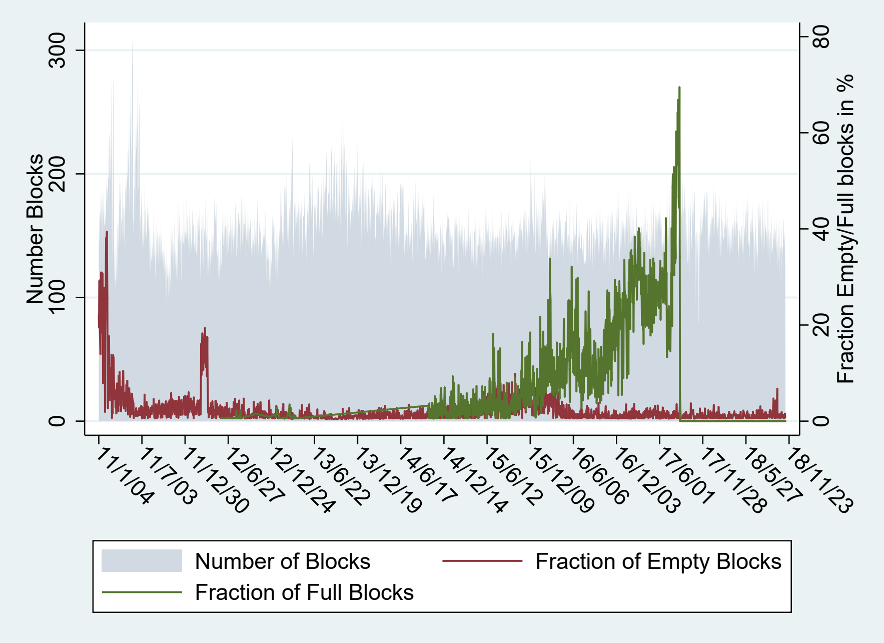

Figure 2. Number of blocks and fraction of empty and full blocks per day. Days are defined over UTC.

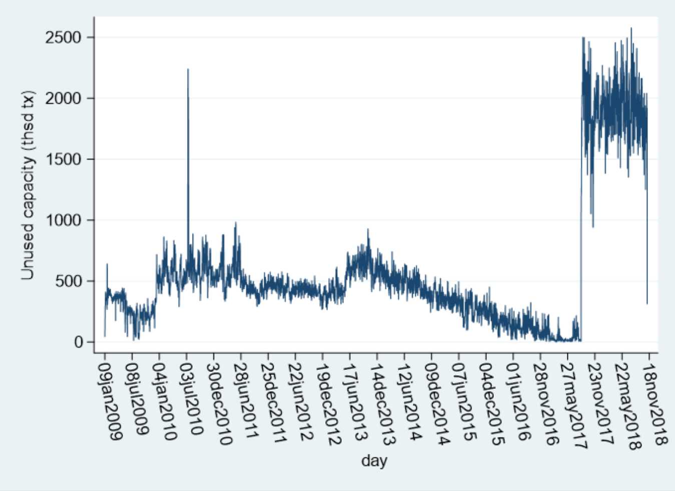

Figure 3. Unused Capacity denominated in thousands of the median transaction size.

an empty mempool. Our measure of mempool size is conservative as we only keep observations immediately after a block was mined. Our graph therefore represents the part of the mempool that was not picked up when miners confirm transactions in a new block.[4] In our sample the

[4] We go through the minute by minute snapshots of the mempool and look for instances where the size and the number of transactions in the pool drop and identify these drops as blocks being mined. We cannot reconcile

minimum size of the mempool was 112 KB, at any time there were at least 214 unconfirmed transactions in the mempool.

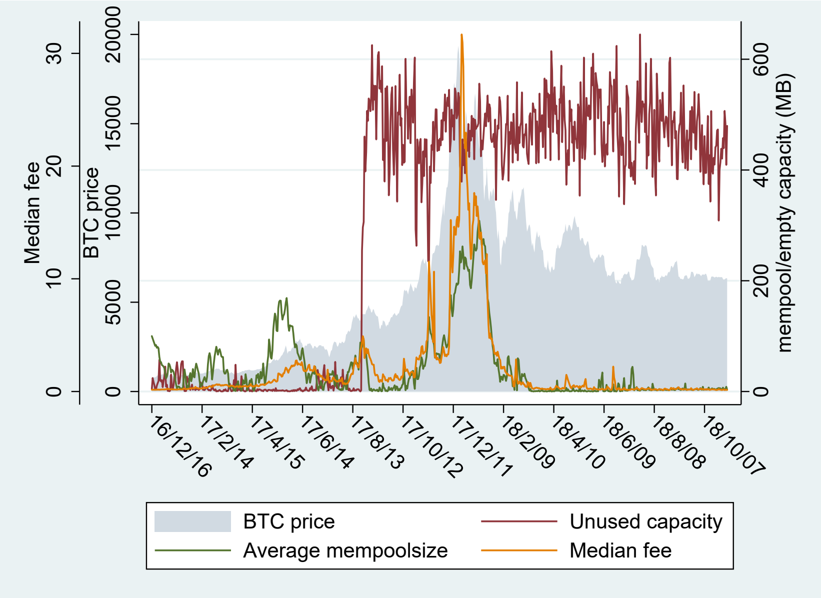

Figure 4. Unused Capacity denominated in thousands of the median transaction size.

In addition to the strategic reasons for mining a non-empty block, there is a financial motive. As we mentioned above, in addition to the coinbase or block reward, the successful miners may receive fees from agents who want to have their transaction prioritized. Thus, if there are transactions waiting in a miner's mempool with fees attached, the successful miner appears to be "leaving money on the table" by failing to put such transactions in his block. To get a sense of the importance of fees, Figure 5 indicates fee revenue as fraction of the block reward. The red lines indicate cuts in the block reward. Clearly, from 2016, fees gain importance as source of revenue for miners.

Figure 6 shows the total number of transactions over time as well as the number of transactions with zero fees over time. The number of transactions increases over time and peaks in mid December 2017 coinciding with the peak in Bitcoin prices. The fraction of transactions without fees declines over time. On the last full month of our sample, October 2018 only 67 out of 7,570,407 non-coinbase transactions were mined without a fee. High variation in Bitcoin's early days are possibly could arise from low transaction volume.

blocks based on the timestamp as timestamps of blocs are inaccurate. Because mempool data is specific to each node as transactions are shared via peer-to-peer communication our data might have at a given time more or less transactions in the mempool than other nodes.

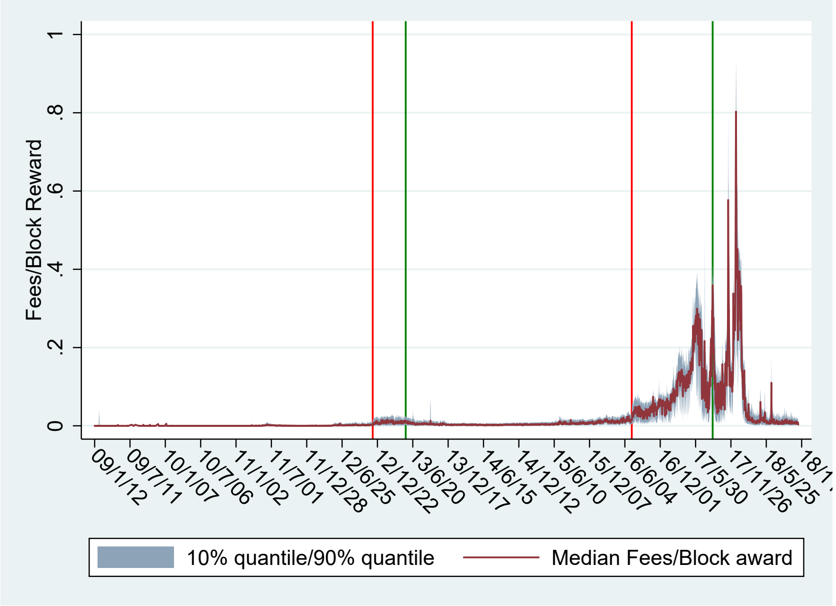

Figure 5. Fees and Block Rewards: Days are defined over UTC. Fees per block as a fraction of the block reward. Median and 10%/90% quantile are of the daily distribution. The red lines show when the block reward was cut in half, the green lines indicate when the block capacity was increased.

3 Optimally "leaving money on the table"

To motivate our framework, we present a simple model of on-chain settlement. The population comprises two types of bidders, those with high valuations for settlement v_h, and those with low, v_ℓ, where $v_h > v_\ell > 0$. These agents also differ in the cost incurred for every block that the transaction is delayed, where $c_h > c_\ell$, and we normalize c_ℓ to zero. We refer to the pair $\{c_i, v_i\}$ as the type of the agent i, where $i \in \{\ell, h\}$. For compactness we sometime denote that $c_h - c_\ell = \Delta_c$ and $v_h - v_\ell = \Delta_v$.

Each agent arriving at the market has the same outside option, $\bar{v} > 0$, which determines his willingness to participate. This is the expected utility any agent participating in the Bitcoin system obtains if he chooses his own optimal alternative.

We consider an infinite horizon, discrete time model. Here, one period corresponds to one mining block, so the culmination of each period is the addition of another block to the chain. The sequence of events within a period is as follows: first, λ_i agents of type $i = h, \ell$ arrive, and then submit their transaction (normalized to 1) and bid b_i to the network. Transactions and bids are then submitted to a common mempool and the transaction servicer then picks the bids to process.

Each block has a fixed capacity of κ. We assume that $\lambda_h < \kappa < \lambda_h + \lambda_\ell$, so that there is congestion in the blockchain and some types will have to be rationed. (We note that if there is

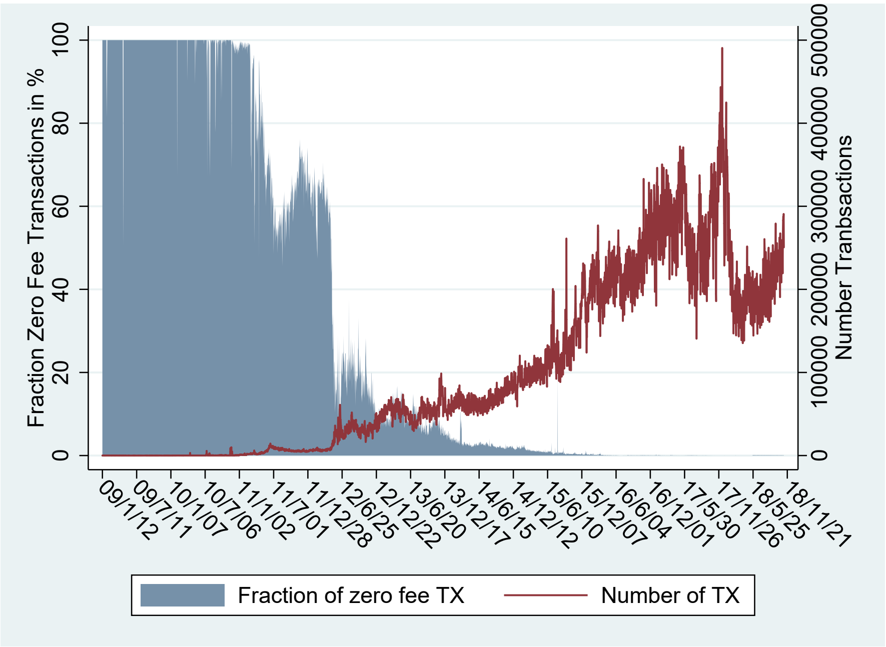

Figure 6. Number of transactions per day: Number of total transactions (red, right axis) and fraction of transaction without fees (blue area, left axis). Days are defined over UTC. Transactions exclude coinbase transactions

no congestion, in the case in which the servicer exhausts capacity, the optimal bid is always 0.)

In reality, each miner keeps track their own mempool. We abstract from this and consider a common mempool. Unprocessed bids remain in the mempool, as agents cannot revise or cancel their bids. However, as a practical matter, a mempool does not grow without bound because each mining pool flushes its mempool depending on its local conditions and capacity. For modeling purposes, we assume that the common mempool is flushed after ever period, but that agents whose transaction was not settled automatically resubmit their transaction at the same price. Thus, the arrival rate λ_i should be understood to include new transactions and those that were submitted earlier but have not been executed. The timeline for one period is presented in Figure 7 below.

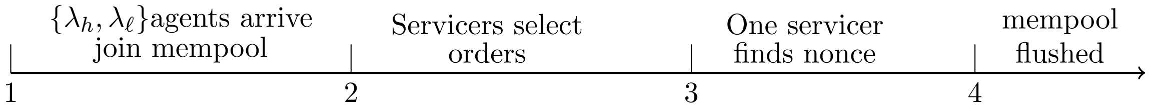

Figure 7. Sequence of Events within a block mining period

We characterize time invariant outcomes both because agents (and econometricians) do not observe the contents of the mempool, and the time between blocks – on average 10 minutes – probably does not admit time trends. We note, however, that in our empirical section, we use the random time between blocks as a proxy for congestion. In our base model, the arrival rates of different types are exogenous. In section 3.3, we consider the relationship between the stock of agents in the mempool and the in-flows and out-flows.

Let p_i be the time invariant probability that a transaction with bid b_i is executed in the current block. Then, an agent with cost c_i and valuation v_i, who submits a bid of b_i that executes after k periods, garners expected utility of $v_i - b_i - c_i(k-1)$. The costs of waiting are illustrated in Figure 8 below.

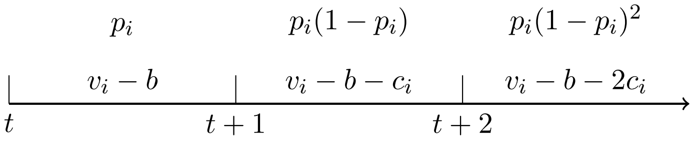

Figure 8. Probability of execution and payoff for an agent with type $\{c_i, v_i\}$. Here t refers to the block height.

Lemma 1 *The expected utility of an agent who submits a bid b_i and anticipates an execution probability p_i is*

$$V(b_i \mid c_i) \quad = \quad (v_i - b_i) - c_i \frac{1-p_i}{p_i}. \tag{1}$$

Agents take the execution probabilities as given. However, the execution probabilities depend on the actions of the miners, i.e., those who select transactions out of the mempool to process. We characterize two types of liquidity provision, first in which miners execute all orders by price priority and operate at full capacity, and second in which miners execute orders by price priority and ration strategically.

3.1 Different Rationing Rules

We know from the utility function of the consumer, and the participation constraint that the bid is

$$b_i^* \quad \leq \quad v_i - \bar{v} - c_i \frac{1-p_i}{p_i}.$$

Further, because $v_h > v_\ell$ and $c_h > c_\ell = 0$, we know that the type h has an incentive to bid higher than the type ℓ, and so we consider the minimum bid at which the type h agents execute with probability 1 and the type ℓ are rationed. Let the superscript c denote bids, and execution probabilities if the servicer exhausts all available capacity.

Proposition 2 (Competitive Servicing) *If the servicers execute by price priority and exhaust all capacity then* *i. Type h bids no more than $b_h^c = \lim_{\epsilon \to 0} v_\ell - \bar{v} + \epsilon$, and executes with probability $p_h^c = 1$.*

ii. Type ℓ bids at most $b_\ell^c = v_\ell - \bar{v}$ and executes with at least probability $p_\ell^c = \frac{\kappa - \lambda_h}{\lambda_\ell}$

iii. The profits of the servicer are $\pi^c = ((v_\ell - \bar{v}))\,\kappa$.

If the servicer commits to exhaust all capacity, then the execution probability of the lower bidder type is determined by the arrival of traders and block capacity. The high types bids just enough to ensure that their orders will always execute. We note in passing that although we have continuous prices, in a model with arbitrarily small prices, the high types would bid the smallest price increment above the lowest types. In short, the high type retains rents and enjoys a surplus by participating in the system.

By contrast, if the servicer strategically chooses the execution probabilities, he can induce higher bids. We establish that if a transaction servicer can commit to execution probabilities for certain bids, he can elicit higher bids from agents who face higher costs. Specifically, the servicer only executes b_h^d with probability 1, and all other bids with probability p_ℓ^d.

Proposition 3 (Discriminatory Servicing) *If the servicers execute by price priority and ration, then*

i. Type h bids $b_h^d = v_h - \bar{v}$, and executes with probability $p_h^d = 1$.

ii. Type ℓ bids $b_\ell^d = v_\ell - \bar{v} - c_\ell\left(\frac{\Delta_v}{\Delta_c}\right)$ and executes with probability $p_\ell^d = \min\left[\frac{\Delta_c}{\Delta_v + \Delta_c}, \kappa - \lambda_h\right]$

iii. The profits of the servicer are $\pi^d = (v_h - \bar{v})\lambda_h + \frac{(v_\ell - \bar{v})\Delta_c - c_\ell \Delta_v}{\Delta_v + \Delta_c}\lambda_\ell$

.

These two different types of servicing strategies will generate different observables. First, it follows immediately from inspection of the two propositions that the maximum bid observed under discriminatory servicing is higher than that under full capacity servicing.

Corollary 1 *The maximum bid observed under discriminatory pricing is higher than the maximum bid under capacity servicing.*

Further, changes in capacity will have differential effects on the observed bids. Under full capacity servicing, the higher the capacity, the higher the utility agents get from using bitcoin. They are therefore willing to pay more for the system. Thus, after the Segewit, bids should increase. By contrast, in the discriminatory environment, bids are insensitive to capacity, so there is no effect of the Segewit.

Corollary 2 *In full capacity servicing, bids are increasing in capacity, whereas under discriminatory servicing, changes in capacity will have no effect on bids.*

3.2 Sustaining Discriminatory Pricing with Competitive Miners

Suppose that discriminatory pricing is optimal. It is straight forward to show that even a decentralized financial system such as the BitCoin protocol can lead to discriminatory pricing equilibrium. Suppose that there are $i = 1, \dots N$ servicers of varying size. Let χ_i denote a servicer's hash rate as the proportion of the total Bitcoin system hash capacity. We assume that this is also the probability that a servicer wins the nonce.

Consider the payoff to servicer i. Mining garners a block reward in addition to fees. For simplicity, we normalize the block reward to zero as this is awarded mechanically. Under pricing regime $j = c, d$, a servicer obtains a per block profit at time t of π_t^j. Thus, under either regime $j = d, c$, the servicer anticipates from the next block onwards of

$$\Pi_i^j = \sum_{t=1}^{\infty} \delta^{t-1} \chi_i \pi_t^d \qquad j = d, c. \tag{2}$$

Thus, a servicer who is playing adhering to discriminatory pricing and wins the nonce receives a payoff of:

$$\Pi_i^{coop} = (v_h - \bar{v})\lambda_h + \Big((v_\ell - \bar{v}) - c_\ell \frac{1 - p_\ell^d}{p_\ell^d}\Big)\lambda_\ell p_\ell^d + \delta \Pi_i^d$$

Discriminatory pricing requires each miner to refrain from executing low bids to ensure that their expected execution probability is sufficiently low that the agents who value a rapid settlement are willing to bid aggressively up to their valuation. A miner who deviates would include too many transactions from the low bid types. If he finds the nonce, this would become public. Full capacity mining in perpetuity is a natural punishment strategy. The maximum profit to deviating is thus:

$$\Pi_i^{dev} = (v_h - \bar{v})\lambda_h + (\kappa - \lambda_h)\Big((v_\ell - \bar{v}) - c_\ell \frac{1 - p_\ell^d}{p_\ell^d}\Big) + \delta \Pi_i^c \tag{3}$$

It is immediate in this framework that if the payoff to discriminating is higher than the payoff to full capacity mining, then full capacity as a punishment strategy will sustain the discriminatory pricing outcome. The condition to sustain collusive outcome is

$$\Pi_i^{coop} \geq \Pi_i^{dev} \tag{4}$$

$$\delta(\Pi_i^d - \Pi_i^c) \geq \Big((v_\ell - \bar{v}) - c_\ell \frac{1 - p_\ell^d}{p_\ell^d}\Big)\Big(\kappa - \lambda_h - \lambda_\ell p_\ell^d\Big) \tag{5}$$

As with most collusive equilibria, the fewer the participants, the easier it is to sustain collusive equilibria. In this framework, fewer participants is equivalent to a higher probability that a particular miner or mining pool finds the nonce and is successful. This suggests that an effect of mining pools is to make collusive equilibria much easier to sustain. To see this note, the continuation payoffs in equation 5) are scaled by the probability that a miner wins the nonce. Thus, to sustain collusion, a minimum mining capacity as a function of the total capacity is required. Finally, we note that the miners' habit of signing the blocks they mine ensures that other pools can easily verify that a pool did not exhaust capacity.

3.3 Long and Short Run

The analysis so far takes as given the arrival rate of the different types of traders as given, thus the analysis is essentially "short run." However, we note that the arrival rate will be determined by investors' participation constraint. In what follows, we consider what effect these strategies have on the arrival rates under the various regimes. This illustrates that, given a fixed participation constraint, the volume of trade will depend on the agents' execution probabilities which in turn depend on the servicers' strategies. To clarify the way in which different arrival rates may be affected by different servicing strategies, we use the notation λ_i^j, and m_i^j to represent the stock of a particular type in the mempool where $j = c, d$ and $i = h, \ell$ to distinguish between the different regimes.

Proposition 4 *In long run steady steady state,*

i. Under full capacity servicing, the arrival rate of the high types, $\lambda_h^c = \kappa$.

ii. Under discriminatory servicing, the arrival rate of high types, $\lambda_h^d \subseteq [0, \kappa]$

Clearly, $\lambda_h^d \leq \lambda_h^c$, so under full capacity servicing, the arrival rate of agents with extreme values is be higher. Further, if the size of the mempool does not grow, then in both cases, $\lambda_\ell^j = \kappa - \lambda_h^j$, $j = c, d$.

Corollary 3 *The distribution of types that arrive is driven by the liquidity supply strategy. With the discriminatory equilibrium, there are fewer high (extreme) types than the capacity constrained equilibrium.*

4 Empirical Analysis

4.1 Fee Heterogeneity

Figure 9 plots the time series of the median as well as the 25 and 75 percentiles of transaction fees starting in March 2015 when reliable price data exists. The graphs for USD and KRW are similar.

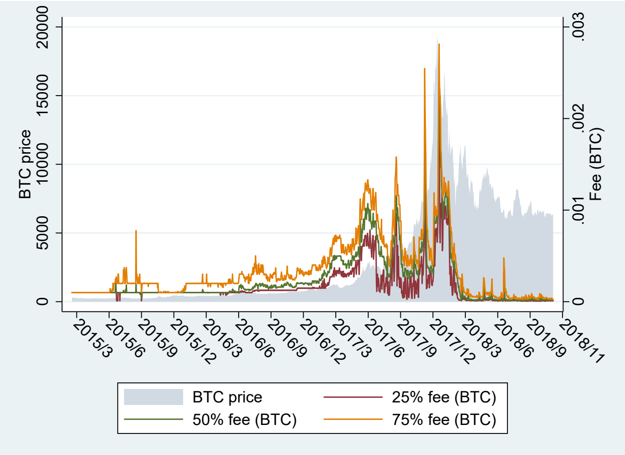

Figure 9. Bitcoin prices (gray area, left axis) and median, 25 and 75 percentile of daily fees (right axis) in BTC. Days are defined over UTC. Transactions exclude coinbase transactions

One of the key predictions of discriminatory service is a high dispersion in fees. Figure 10 illustrates the heterogeneity of the Bitcoin fee distribution in December 2017, which was the peak of Bitcoin prices in our sample. The left panel shows the lower end of the fee distribution documenting that a substantial fraction of transactions occurred at low or modest fees. The right panel details the upper end of the distribution documenting that as Bitcoin fell from its peak price some people were willing to pay enormous fees to have their transactions processed. On Dec 24, 2017, for example, the 99 percentile fee was USD 433.23 while the largest fee paid was USD 14,174.64. We note that there are 80 transactions in our sample with fees greater than USD 10,000, of which 51 occur between Dec 20, 2017 and Dec 24, 2017. However over these same five days 1,674,141 transactions were processed out of which 752 had no fee and 16,191 transactions are mined with fees less than USD 5.

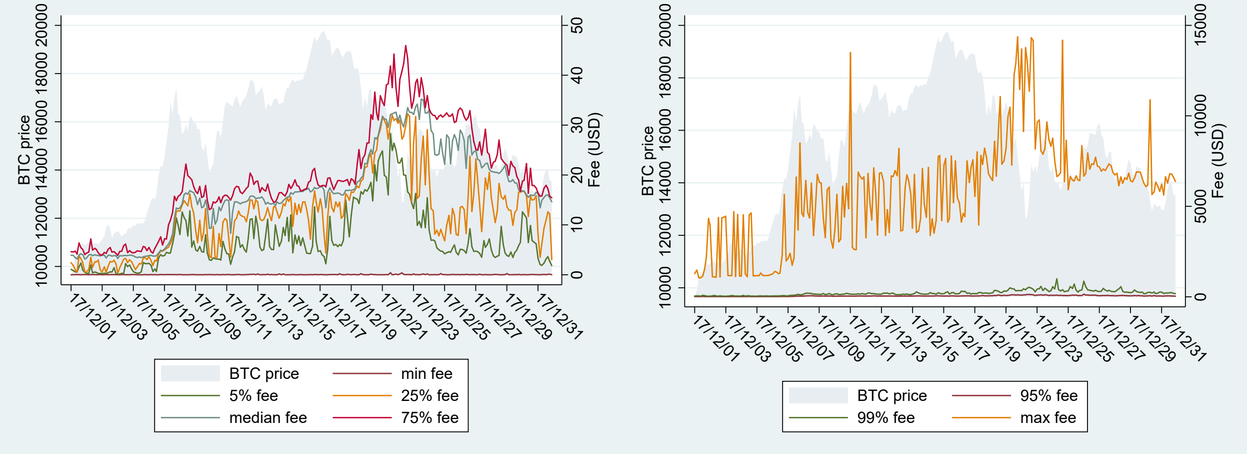

Figure 10. Fees in USD (right axis) and Bitcoin price in USD (left axis) in December 2017.

4.2 Cross-sectional determinants

Our model predicts that under discriminatory pricing fees increase with the valuation of the high type, while under competitive pricing the valuation of the low type determines the fees. In this section we first analyze how transaction and blockchain specific variables drive fees and then provide evidence that is consistent with the idea that high value types pay higher fees. Recall, that we posit two dimensions to liquidity, the cost of waiting per block and the valuation of having a transaction processed.

We define minimum outtime as the time (in blocks) until the first output of this transaction is spent again. Recipient wallet owners that spend their funds very quickly have a more immediate need for funds and might thus put pressure on their debtors to send funds quickly. The Op-ret dummy is set to one for all data insertion transactions as discussed in Section **??**. The Sum Inputs variable adds up all input values to a transaction. We choose inputs rather than outputs because some transactions, most of them data-insertion transactions, have almost all their inputs dedicated to fees and only have negligible output. Transaction size is the physical size of all inputs and outputs in bytes. The transaction size represents an opportunity cost for miners as block space is limited. Blocksize is the absolute size of the block in bytes. Larger blocks have less space for additional transactions and are an indication for transaction demand.

	Nobs	Mean	Std.Dev.	Median	Min	Max
Fee (Satoshi)	353,306,421	52,406	2,826,555	18,802	0	29,124,090,000
Fee (USD)	291,702,649	2.90	36.86	.20	0	137,186.10
Sumin (Satoshi)	353,306,421	1,370,000,000	40,500,000,000	19,800,000	0	55,000,000,000,000
Sumin(USD)	291,702,649	25,690.01	556,020.30	256.64	0	947,000,000.00
Blocksize (bytes)	353,306,421	834,072	325,063	998,126	267	2,324,736
Tx-Size (bytes)	355,725,559	537	2,210	250	62	999,657
dum-OPRet	353,306,421	.02	.15	.00	.00	1.00
Minouttime (blocks)	350,031,634	547.54	4,950.34	4.00	.00	474,195.00
Priceusd (USD)	291,702,649	3,604.66	4,162.71	1,188.28	204.84	19,828.76

Table 1. Summary statistics

Table 1 contains summary statistics. Some transactions contain unusually high fees which are either errors or payments miners make to themselves to consolidate their bitcoin balances together with the block award they earned under a new address. For example on April 26, 2016 someone paid 291.241 BTC to an output with 0.0001 BTC, leaving fees of 291.2409 (or USD 137,186.07 at the time) for the miner.[5] We found overall 80 transactions with fees over USD 10,000. These transactions have an average input of 140.89 BTC, and an average fee of 7.55 BTC or (USD 16,974.42 at the time). These transactions can be found between April 2016 and October 2018, but most of them are concentrated from Dec 21-14 in 2017. Bitcoin prices peaked on Dec 16 2017 at USD 19,200 and feel back to USD 15,191 on Dec 21. One potential explanation for these high fees is that investors wanted to move Bitcoin from their private wallet to an exchange quickly in a market panic. Transactions from the earlier days of Bitcoin may also show unusually high fees denominated in bitcoin. In December 2011 a transaction can be

[5] this is the highest fee transaction in BTC and USD terms in our sample, see transaction cc455ae816e6cdafdb58d54e35d4f46d860047458eacf1c7405dc634631c570d.

found with inputs of over 207 BTC and outputs of 36 BTC, leaving a huge fee for the miner. At that time bitcoin were worth very little, yet since there was no shortage of transaction space in the blockchain, there is no obvious rationale for paying such a high fee.

Many variables show some extreme observations. The average Bitcoin transaction was for 13.7 BTC (one bitcoin equals 100 million Satoshi) while the largest transaction was for 550,000 BTC on Nov 16, 2011 at a zero fee.[6] The largest transaction in dollar terms occurred on Dec 17th, 2017 at the peak of the bitcoin price when 48,500 BTC valued at over 946 million USD changed wallet for a fee of 80,908 Satoshi or USD 15.8.[7] In our sample we find 54,907 transactions with a value of more than USD 10 million. Out of those 20,956 were processed with a fee of less than USD 5 and the average fee of those transactions was USD 90.54.

Most transactions are small in size with a mean of 537 and a median of 250 bytes. The largest transaction in our sample consumes the entire block 364,292 with a size of 999,657 bytes. The transaction was mined on July 7th, 2015 and consolidates 5,569 inputs of 1,000 Satoshi each in a single output.[8] Minouttime measures the time (in number of blocks) until the first output of a transaction gets spent again. In some cases a transaction output gets spent in the same block that it is received, showing a minouttime of zero. The mining difficulty is set such that on average one block is mined every 10 minutes. We chose blocks and not calendar time as measure to make our findings robust to smaller variations in the time between blocks especially when the output is spent shortly after it is being received. For longer timespans the difference between calendar time and block time is irrelevant as blocks are mined every 10 minutes.

To purge our sample from extreme outliers we windsorize our fee data, the transaction size, the inputs, and the outtime at the 99.9% level. We find our results to be robust to different levels of windsorizing. Bitcoin fees also vary tremendously over time. To control for this time variation we include day-fixed effects in our regression analysis. To control for variation of fees across miners we cluster standard errors per block.

We first examine how transaction and blockchain specific explanatory variables influence transaction fees. Table 2 presents our findings for the fees in term of Satoshis (1 BTC is 100 million Satoshis) and Table 3 for fees in USD. In line with our intuition fees are higher when transactions space is scarce (larger blocksize) and when the transaction is larger in terms of bytes (TXSize) and value (Sum Inputs). In the context of our model, this arises because more high fee transactions are in the mempool and therefore the block is larger.

Fees are also higher when the funds are spent sooner (lower min. outtime). One possible explanation is that receivers who are in need for liquidity pressure senders to accelerate payments by

[6] see transaction 29a3efd3ef04f9153d47a990bd7b048a4b2d213daaa5fb8ed670fb85f13bdbcf

[7] see transaction 261d69b25896034325d8ad3e0668f963346fd79baefb6a73b4eabd68c58c81ff

[8] According to some forum posts an unknown spammer created all these wallets to test the limits of the UTXO database. Somebody created many small outputs locked up under different addresses. To speed up the time it takes to verify the validity of a transaction each node holds in RAM memory a complete lust of all unspent BTC amounts per address. This list is called the Unspent Transaction Output (UTXO) database. The attack of the spammer forced each of the nodes to expand the UTXO list thereby increasing the memory requirement for each node. The Intentions of the spammer are unclear as all the transactions were protected by very weak private keys such as *'passowrd'* or *'cat'*. Somebody eventually guessed the private keys and consolidated all the small inputs in block 364,292, freeing up space in the UTXO list.

Blocksize	0.0024***					0.0012***
	(0.0001)					(0.0001)
TXSize		49.4070***				49.5693***
		(0.1380)				(0.1382)
Sum Inputs			1.04e-06***			5.75e-07***
			(8.65e-09)			(5.65e-09)
OP-Ret				-10371.4000***		-502.8570**
				(232.5050)		(243.3099)
Min. Outtime					-1.2260***	-2.1106***
					(0.0144)	(0.0199)
constant	44750.4000***	21403.3000***	45713.6000***	46993.6000***	47288.5000***	20521.8000***
	(74.5277)	(67.0622)	(39.8744)	(40.5487)	(40.2871)	(115.3693)
R^2	0.0951	0.3688	0.0973	0.0952	0.0952	0.3708
N	353,306,421	353,281,736	353,306,421	353,306,421	350,031,634	350,008,404

Table 2. Regression results of fees in Satoshis (1 BTC is 100 million Satoshis). Regressions include day fixed effects. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

offering higher fees. Data insertion transactions pay on average lower fees in BTC other users of the blockchain. While being statistically significant some results are economically insignificant. The value of the transaction, for example, only has a small impact on the fees, which is consistent with the fact that the miner's opportunity cost for including a transaction in a block of limited size is determined by the transaction size in bytes and independent of the value. In Appendix A we provide more details on the determinants of transaction size. Transaction size has therefore an economically significant impact on fees. Adding one input (with an average size of 180 bytes) to a transaction (the average transaction has 2.5 inputs in our sample) increases the fee by 8,893 Satoshi or USD 0.41. Given that the median fee over the whole sample is USD 0.20 this effect is economically large. Results are mixed for the data-insertion (OP-Ret) dummy. In terms of BTC we see that data-insertion transactions post lower fees, yet when controlling for other variables we see that they pay a higher fee in USD. We attribute that to the increasing volume of data insertion transactions towards the end of our sample, where BTC prices were generally higher than at the beginning.

Outtime is measured in blocks, which are mined every 10 minutes on average. People spending their funds a day (∼144 blocks) earlier pay on average USD 0.014 more in fees. Receivers that are keen to spend their coins sooner put a higher value on the execution. Consistent with discriminatory service those users pay higher fees as miners are able to price discriminate by delaying users that put a low priority on execution.

We first analyze if fees are higher at certain times. We argue that more sophisticated, high value, investors are active during the week and when the Bitcoin futures at CME are trading.[9] The results in Table 4 are consistent with discriminatory service. Transaction fees are lower on weekends by almost USD 0.196, which is close to median transaction fee for the whole sample.

[9] BTC futures at CME are trading Sunday to Friday from 6pm to 5pm EDT with a daily one hour break between 5pm and 6pm EDT. Futures were first traded on December 18, 2017. Settlement is on the last Friday of the contract month. For this analysis we convert all block timestamps from UTC to Eastern Time considering summer daylight savings time when appropriate.

Blocksize	9.06e-08***					5.03e-08***
	(5.89e-09)					(6.87e-09)
TXSize		0.0023***				0.0023***
		(9.90e-06)				(9.76e-06)
Sum Inputs			7.69e-06***			6.15e-06***
			(6.65e-08)			(5.85e-08)
OP-Ret				-0.3681***		0.1247***
				(0.0168)		(0.0157)
Min. Outtime					-0.0001***	-0.0001***
					(1.30e-06)	(1.45e-06)
constant	2.3626***	1.2527***	2.2953***	2.4579***	2.4814***	1.1323***
	(0.0068)	(0.0054)	(0.0034)	(0.0035)	(0.0035)	(0.0088)
R^2	0.2741	0.3997	0.2858	0.2741	0.2747	0.4079
N	291,702,649	291,678,054	291,702,649	291,702,649	288,555,299	288,532,152

Table 3. Regression results of fees in USD. Regressions include day fixed effects. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

Looking at individual days of the week it seems that fees gradually increase during the work week with the highest observed fees on Fridays (which is also the settlement day for futures). Fees are also higher by about half a dollar, or twice the median fee, whenever the futures market is open.

Bitcoin is a pseudo-anonymous system, all wallet addresses can be identified and payments can be tracked moving from one wallet to another but the identity of the wallet owner is usually unknown. In some cases, e.g. voluntary disclosure, court proceedings, the owner of some addresses gets known. Gambling sites often use vanity addresses, which start with a suggestive word like '1dice....' or '1Lucky....' which are easily identified. Such vanity addresses have to be found by trial and error in a time consuming process and therefore get often reused.[10] Once an address is known other addresses controlled by the same walled can be inferred in a process commonly known as address clustering see e.g. **?** or **?**. The idea is that if multiple addresses are used as inputs in the same transaction these addresses most likely belong to the same person because the private key has to be used to sign the transaction.[11]

We use data from lists of known addresses such as wallet explorer and are able to identify the identity of the sender for 18,123,498 transactions, out of which 7,250,374 (or 2.05% of all transactions) were initiated by an exchange and 10,873,124 (or 3.07% of all transactions) were initiated by a gambling site. Similarly we are able to identify 32,771.960 payments to an exchange and 23,099,021 payments to a gambling site. Table 5 presents our findings for fees in USD. Flows to and from exchanges transact at higher than average fees. Since we control for day fixed effects our results cannot be driven by more exchange flows occurring on days when fees are generally

[10] Addresses are encoded in a Base58 alphabet (i.e. there are 58 possible 'letters' consisting of upper case, lower case letters and numbers with some combinations dropped that are often mixed up when printed on paper, e.g. capital i and lower case L) and start with 1. To get an address with '1Lucky' one has to try $58^5 \approx 656 million$ combinations. Vanity address companies offer computing resources for custom bitcoin addresses.

[11] One notable exception are anonymizing services that for a fee combine transactions of several users into one large transaction so that it is not that clear who paid whom. See e.g. **?**.

Blocksize	1.88e-08**	3.90e-08***	4.47e-08***
	(8.34e-09)	(8.13e-09)	(8.27e-08)
TXSize	0.0023***	0.0022***	0.0023***
	(9.76e-06)	(9.76e-06)	(9.76e-06)
Sum Inputs	6.13e-06***	6.13e-06***	6.13e-06***
	(5.91e-08)	(5.91e-08)	(5.91e-08)
OP-Ret dummy	0.0832***	0.0848***	0.08056***
	(0.01870)	(0.01859)	(0.01859)
Min. Outtime	-0.0001***	-0.0001***	-0.0001***
	(1.51e-06)	(1.51e-06)	(1.51e-06)
Monday	0.2683***		
	(0.02050)		
Tuesday	0.3749***		
	(0.01963)		
Wednesday	0.4388***		
	(0.01873)		
Thursday	0.5236***		
	(0.01961)		
Friday	0.6379***		
	(0.02209)		
Saturday	0.4847***		
	(0.02119)		
Weekend		-0.1962***	
		(0.01197)	
Open Futures Mkt			0.4932***
			(0.0369)
Constant	0.7662***	1.1966***	1.0571***
	(0.01649)	(0.01089)	(0.01045)
R^2	0.3952	0.3949	0.3950
N	288,532,152	288,532,152	288,532,152

Table 4. Regression results of fees in USD - day of the week and opening hours of the futures market. Regressions include week fixed effects. Days are defined in UTC. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

higher. Our findings are also not driven by outliers since the data is windsorized. Transactions flowing out of exchanges pay USD 3.15 more, which is remarkable given that the median fee for the whole sample is USD 0.19. Fees into exchanges are paying USD 1.24 more than average. We argue that traders moving funds in and out of exchanges and gamblers put a high value on immediate execution. Consistent with discriminatory service we observe that high types pay higher fees.

We investigate how block capacity or "congestion" affects the fees that are submitted. First, we examine how many blocks have been recently mined. Notice that the time between blocks is a random variable. The Bitcoin protocol calibrates the difficulty, i.e. the number of leading zeros that a block hash has to have to qualify as valid, in such a way that on average a new block is added every ten minutes. Yet the times between blocks as they are recorded on the blockchain wary widely as illustrated in Figure 11. This is because mining a successful block is purely random and so sometimes blocks are found very quickly and sometimes it takes a long time. Another reason is that the time a block was mined is self reported by the miner and

	Identity sender		Identity Receiver	
Blocksize		7.00e-08***		5.76e-08***
		(0.0000)		(0.0000)
TXSize		0.0023***		0.0024***
		(0.0000)		(0.0000)
Sum Inputs		5.98e-06***		7.75e-06***
		(0.0000)		(0.0000)
OP-Ret dummy		-0.0874***		0.0884***
		(0.0154)		(0.0158)
Minimum Outtime		-0.0001***		-0.0001***
		(0.0000)		(0.0000)
Exchange	5.4745***	3.1496***	2.1978***	1.2460***
	(0.0478)	(0.0350)	(0.0140)	(0.0099)
Gambling	0.3528***	0.1914***	0.7724***	0.5259***
	(0.0039)	(0.0069)	(0.0060)	(0.0050)
constant	2.3485***	1.0723***	2.3281***	0.9826***
	(0.0033)	(0.0089)	(0.0036)	(0.0098)
R^2	0.2789	0.4095	0.2637	0.4062
N	291,702,649	288,532,152	294,586,994	291,416,436

Table 5. Regression results of fees in USD - day of the week. Regressions include week fixed effects. Days are defined in UTC. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

miners can have their local clocks not aligned with the world time. These clock mis-alignments can be substantial. Out of the 548,648 blocks in our sample we find 13,848 cases in which a block has an earlier time-stamp than its predecessor. This is technically impossible. Each block contains information from the previous block, which links the blocks together in a blockchain. The only rational explanations for the inconsistency in time-stamps is improper alignment of miners' clocks. To accommodate potential synchronization problems in miners' clocks the bitcoin protocol allows a block to have time-stamp up to two hours earlier than the previously mined block. Despite these problem cases for the vast majority of the sample the time-stamps seem to be properly recorded.

Under the assumption that the arrival rate of orders to the mempool is independent of the time between blocks, then fewer recently mined blocks should imply more congestion in the mempool. Second, we measure the capacity utilization of the recently mined blocks, which we define as the fraction of available space used by the miners. Table 6 includes our measures of congestion in the fee regression. The coefficient on both variables is negative. So, if more blocks have been mined recently the lower the fees in addition, the lower the capacity use, the higher the fees. At least the second observation is consistent with price discrimination. When miners successfully coordinate to leave more block-space empty by delaying users that put low priority on execution, the users that are keen on execution will bid more in equilibrium.

One group of users with high valuations on execution are arbitrageurs who take advantage of cross exchange pricing differences. The arbitrageurs' preference for immediacy will be proportional to the price differential between exchanges as they need to process transactions on the blockchain to move Bitcoin from one exchange to the next. Under competitive pricing we should find that fees are independent of the value that the high types put on immediacy. Under dis-

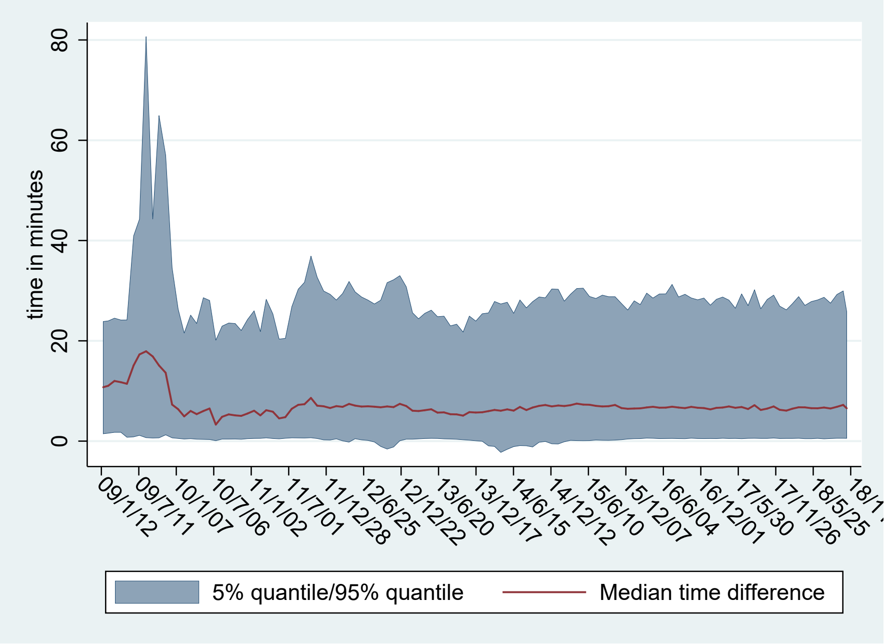

Figure 11. Time between blocks as recorded on the blockchain. Quantiles are computed from the monthly distribution.

criminatory pricing we should find that miners can extract at least part of that surplus and therefore charge arbitrageurs a higher fee when the price differential is high.

To test this prediction empirically we take high frequency prices from Korea and the US and compute the Kimchi premium as the relative price difference of Bitcoin in the US and Korea at the block level. We interact a dummy for payments being made to and from wallets that can be identified as exchanges with the absolute value of the Kimchi premium and and find that fees made to exchanges increase in the Kimchi premium for these users. Since arbitrage profits are increasing in the absolute value of the price difference between BTC markets our evidence is consistent with discriminatory pricing and the idea that miners can extract more if the value of the high type v_h increases.

Our findings are in Table 7. Exchange is a dummy that identifies 35,163,580 payments to and from known exchange wallets, the Kimchi Premium is measured as relative price difference or the BTC price between Korbit in Korea and coinbase in the US, respecively. The interaction term of Exchange and Kimchi preium is statisically and economically significant. For an increase of the Kimchi premium by 10 percentage points users are willing to pay USD 3.22 more in fees. Our analysis most likely underestimate the importance that the demand for immediacy has for equilibrium fees for two reasons. First we can not identify all payments to exchanges. Observed high fees to exchanges might therefore coincide with similar high fee payments to unidentified exchanges making it harder to identify any effect in the data. Second, arbitrage between KRW and USD is only one of many potential trading strategies to exploit price differences, within one country, or between countries. We might therefore also observe high fees for payments to

Blocksize		-3.51e-07***
		(0.0000)
TXSize		0.0023***
		(0.0000)
Sum Inputs		6.11e-06***
		(0.0000)
OP-Ret dummy		0.1416***
		(0.0152)
Minimum Outtime		-0.0001***
		(0.0000)
Num blocks 1h	-0.0957***	-0.1343***
	(0.0018)	(0.0018)
Num Blocks 24h		-0.0107***
		(0.0006)
Capacity use 1h	-0.6112***	-0.8866***
	(0.0160)	(0.0172)
Capacity use 24h		-0.9620***
		(0.0999)
constant	3.3813***	5.0140***
	(0.0193)	(0.1433)
R^2	0.2789	0.4088
N	291,702,649	288,532,152

Table 6. Regression results of fees in USD including measures of congestion. Regressions include day fixed effects. Days are defined in UTC. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

exchanges at times when two different markets have a large price difference which would not be captured in our regression. The results are robust to the introduction of Segwit and other control variables. Our finding cannot be driven by general variation in fees over time as we include day fixed effects.

In sum, fees are higher when the recipient spends the output faster, consistent with some liquidity need of the receiver. We find that transaction fees are higher on workdays and when CME bitcoin futures are trading consistent with the idea that miners can extract a portion of the higher value institutional investors put on execution. Using a sub-sample of identifiable wallets we find that payments to and from exchanges and gambling sites have substantially higher fees. We also find that the fees miners can extract from users making payments to exchanges increase in the opportunities for cross country arbitrage as measured by the Kimchi premium.

4.3 Time Series evidence

In this section we analyze aggregated block level data to test some time series predictions of our model. Our model predicts that fee levels change in the value market participants put on execution but also on their opportunity cost and their costs of waiting. Since all of these parameters are unobservable we cannot use fee levels as explanatory variables.[12] Our model

[12] In the cross sectional regressions we control for time variation in values and costs by incorporating day fixed effects.

Kimchi Premium * exchange	35.05***	35.05***	32.22***
	(0.3815)	(0.3815)	(0.3366)
Exchange	0.93***	0.93***	-0.10***
	(0.0155)	(0.0155)	(0.0135)
Kimchi Premium	-1.52**	-1.52**	-0.58
	(0.6830)	(0.6830)	(0.6292)
Segwit		1.81***	1.11***
		(0.3888)	(0.3720)
Blocksize			-3.30***
			(0.0000)
TXSize			0.00***
			(0.0000)
Sum Inputs			7.62***
			(0.0000)
OP-Ret dummy			0.19***
			(0.0147)
Minimum Outtime			-0.33***
			(0.0054)
Min Outtime Zero			0.00***
			(0.0000)
Num blocks 1h			-0.14***
			(0.0019)
Num Blocks 24h			-0.01***
			(0.0006)
Capacity use 1h			-0.93***
			(0.0183)
Capacity use 24h			-0.93***
			(0.1068)
constant	2.36***	1.71***	4.62***
	(0.0275)	(0.1420)	(0.2043)
N	293,822,933	293,822,933	290,655,603
R^2	0.2706	0.2706	0.4124

Table 7. Regression results of fees in USD including payments to exchanges and the size of the Kimchi premium. Regressions include day fixed effects. Days are defined in UTC. Standard errors are clustered by block. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

predicts that the dispersion of fees is higher under discriminatory service. We define feespread as the difference between the 90% and the 10% quantile of fees in a given block, standardized by the average fee. We choose this measure of fee dispersion over, say, a standard deviation, to reduce the influence of outlyers.

The collusive discriminatory service is easier to maintain whenever mining concentration is high, which we measure by the market share of mining pools. We collect miners' signatures from each block's coinbase transaction. Unlike any other transaction on the bitcoin blockchain the bitcoin in the coinbase are newly created and therefore do not originate from another wallet. Miners use the space reserved for the input script to insert data into the blockchain. This space is used to send messages to the community like the miners' opinion on Bitcoin improvement proposals, it can contain other philosophical statements,[13] but most important for our purpose it usually

[13] e.g. 'Welcome to the real world.' in block 328465, or 'smile to life and life will smile back at you' in block

contains a signature identifying the mining pool. We automatically search for commonly used signatures and then manually examine unidentified blocks for reoccurring signature patterns.[14]

We compute the daily Hirschman Herfindahl index (HHI) of mining concentration as the sum of the squared shares of each mining pool computed over the day where the block is mined. We also compute the miningpools' daily share of all mining activity. Table 8 presents our findings. We find that the feespread increases in both, the HHI and the miningpool's aggregate share of mining activity. This finding is consistent with the idea that more mining by pools and more concentrated mining makes it easier to maintain the collusive discriminatory equilibrium in which the dispersion of fees increase within blocks.

HHI	3.1684***	3.1502***		
	(0.0373)	(0.0421)		
Agg Pool Share			0.8188***	1.1161***
			(0.0134)	(0.0163)
Mpool	0.0144***	0.0022	-0.0150***	-0.0115**
	(0.0047)	(0.0048)	(0.0049)	(0.0048)
Segwit	0.2025***	0.2943***	0.1859***	0.0899***
	(0.0027)	(0.0095)	(0.0027)	(0.0097)
Blocksize		-2.04e-09***		-2.02-07***
		(0.0000)		(0.0000)
TXSize		0.0000		0.0000
		(0.0000)		(0.0000)
Sum Inputs		-1.62e-07*		-1.51e-07*
		(0.0000)		(0.0000)
OP-Ret dummy		1.0195***		1.0220***
		(0.0459)		(0.0462)
Minimum Outtime		0.0002***		0.0002***
		(0.0000)		(0.0000)
Num blocks 1h		0.0117***		0.0119***
		(0.0011)		(0.0011)
Num Blocks 24h		-0.0005***		-0.0014***
		(0.0001)		(0.0001)
Capacity use 1h		0.2621***		0.2538***
		(0.0116)		(0.0119)
Capacity use 24h		-0.0323		-0.4090***
		(0.0212)		(0.0219)
constant	0.8455***	0.7578***	0.4995***	0.5419***
	(0.0057)	(0.0175)	(0.0117)	(0.0208)
R^2	0.0702	0.1095	0.0557	0.0992
N	198,111	198,105	198,111	198,105

Table 8. Regression results of fee spread per block defined as the difference of the 90% and 10% quantile over the average fee. HHI is the Hirschman Herfindahl index of mining pool activity and Agg Pool Share is the share of mining done by mining pools. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

328444, or 'the Lord of the harvest, that he send forth labourers into his harvest' in Block 143822.

[14] It is unclear why pools sign blocks. There is gain in expected revenue from joining a larger pool. While larger pools are expected to mine a block more often, the reward has to be shared among a larger group. In the context of our model mining pools find it optimal to disclose their identity to show that they do not deviate from the collusive equilibrium.

4.4 Overall impact of collusion

The total amount of fees paid in all transactions of our sample is USD 844,537,503.30. How much was extracted from miners via collusion is hard to estimate. According to our model the equilibrium fee under competitive mining should be below the value that the low type puts on execution. If there is no congestion, as it is for practically our whole sample, equilibrium fees should be zero without collusion. As conservative estimate of maximum fees under competitive mining we approximate the fees that the lowest type would be willing to bid with the 10%-quantile of the fee distribution per block. We then define excessive fees as the sum of all fees above the 10%-quantile. For the whole sample these excessive fees sum to USD 557,568,542.21 To make the result robust to outliers we re-compute excessive fees and winsorize fees per block at the 99% quantile. With winsorizing excessive fees for the whole sample add to USD 416,006,674.88. Overall we argue that excessive fees paid because of collusion are between half and two thirds of total fees paid. Figure 12 shows the daily ratio of excessive fees over total fees, bitcoin prices, and the mining concentration as measured by the HHI. Excess fees seem to be positively correlated with mining concentration.

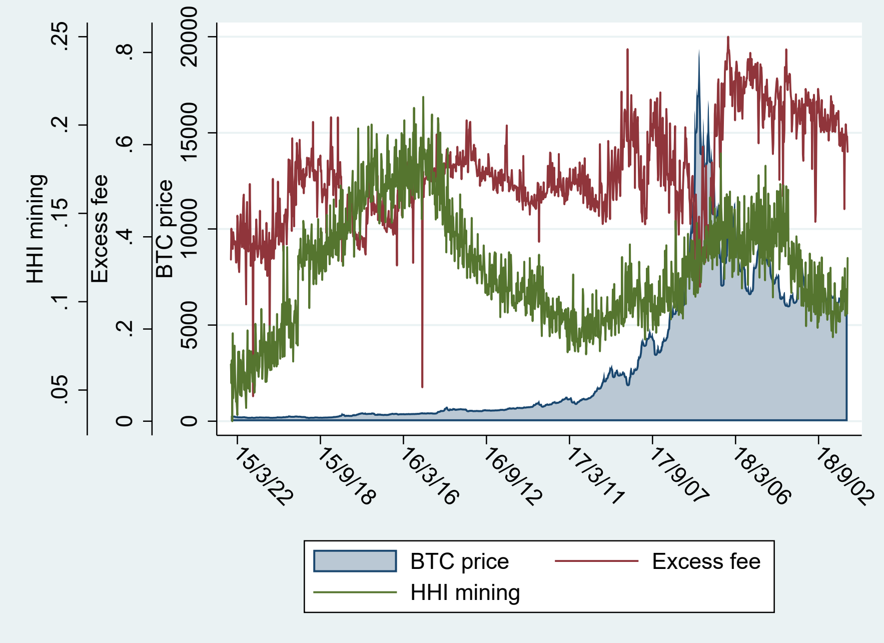

Figure 12. Daily excessive fees, mining concentration as measured by the HHI, and Bitcoin price. Days are defined over UTC.

5 Conclusion

We have documented stylized facts about the Bitcoin protocol. In particular, we observe that there appears to be excess capacity as a significant portion of blocks are empty or not at capacity. We note that this is consistent with collusive price discrimination. Indeed, the rise of fees coincided with the rise of mining pools. Given that the idea behind the bitcoin system was to provide a completely decentralized way of transferring value based on competitive mining, the possibility that there could be collusive equilibria raises questions about the viability of proof of work blockchains.

Our preliminary evidence suggests that one implementation of decentralized finance may operate in a way that is observational equivalent to traditional finance. Indeed, our analysis has highlighted the cost to the consumer of the higher fees they pay because of unused capacity. However, we note that there is a positive side to these rents. Higher profits are one way to ensure that miners will view participating as a valuable exercise which ensures the continuity and stability of this aspect of the BitCoin protocol. Similar to financial intermediaries, market power and the ability to extract rents provide an incentive to continue.

A Determinants of Btcoin Transaction Sizes

Transactions sizes are influenced by the number of inputs and outputs as well as by the type of the transaction. The Bitcoin eco-system allows several types of transactions from the simple pay to an address, to multi-signature transactions where, for example, two out of three people need to digitally sign a transaction for validation, to pay-to-script-hash transactions where the precise rules of redemption are not known at the time an output is locked but only the hash of the redemption script is recorded. Transaction size increases in the number of inputs and outputs with inputs contributing more since they have to hold the transaction signature and the public key. For basic transactions with i inputs and o outputs a common rule of thumb is that the transaction size in bytes is $180i + 34o \pm i$.[15]. To confirm this idea and to show that value does not drive transaction size we regress transaction size in bytes on the number of in, and outputs (numin and numout, respectively) as well as on the value of all inputs (sumin). The results in Table 9 document that the rough rule mentioned above is approximately true for our sample as well. We also find that that the value has no economically significant impact on the size of a transaction.

Numin	164.74***
	(0.06414)
Numout	37.781***
	(0.02734)
Sum Inputs	9.23e-12***
	(1.07e-12)
constant	19.617***
	(0.16118)
R^2	0.9394
N	350,008,404

Table 9. Regression of transaction size in bytes on the number of inputs, number of outputs, and the value of the inputs. One, two, and three stars indicate significance at the 10%, 5%, and 1% level, respectively.

[15] see e.g. https://bitcoin.stackexchange.com/questions/1195/how-to-calculate-transaction-size-before-sending-legacy-non-segwit-p2pkh-p2sh

Proof of Lemma 1

The utility of an agent with cost c_i who bids b and executes in period k is $v - b(v,c) - c(k-1)$. If the probability of execution is p_i, they obtain utility

$$\begin{aligned} V_i(b) &= \sum_{k=1}^{\infty} p_i[1-p_i]^{k-1}(v - b - c_i(k-1)) \\ &= (v-b) - c_i p_i \sum_{j=0}^{\infty}[1-p_i]^j j \\ &= (v-b) - c_i \frac{1-p_i}{p_i}. \end{aligned} \tag{6}$$

where

$$\begin{aligned} S &= \sum_{j=0}^{\infty}[1-p)]^j j \\ &= (1-p)^1 + (1-p)^2 2 + (1-p)^3 3 + \dots (1-p)^n n \\ &= (1-p)\left\{S + \frac{1}{p}\right\} \\ S &= \frac{1-p}{p^2} \end{aligned}$$

■

Proof of Proposition 2

As the servicer is assumed to exhaust capacity, and the type h agents all execute, we know that the probability any single type ℓ will execute is

$$p_\ell^c = \frac{\kappa - \lambda_h}{\lambda_\ell}.$$

Thus, the highest such a type will bid is

$$\begin{aligned} b_\ell^c &= v_\ell - \bar{v} - c_\ell \Big(\frac{1-p_\ell^c}{p_\ell^c}\Big) \\ b_\ell^c &= v_\ell - \bar{v} - c_\ell \Big(\frac{\lambda_\ell - (\kappa - \lambda_h)}{\lambda_\ell}\Big)\Big(\frac{\lambda_\ell}{\kappa - \lambda_h}\Big) \\ b_\ell^c &= v_\ell - \bar{v} - c_\ell \Big(\frac{\lambda_\ell - (\kappa - \lambda_h)}{\kappa - \lambda_h}\Big) \end{aligned}$$

The maximum bid for a low type is $b_\ell^c = v_\ell - \bar{v}$, as this is the highest bid which still generates his reservation utility.

The high type will bid no more than $b_h^c = lim_{\epsilon \to 0}(b_\ell^c + \epsilon)$ as at this price, he will obtain immediate execution.

Thus, if the servicer uses all capacity, the maximum bids of the agents is

Type	Bid	Execution Probability	Quantity
1	$v_\ell - \bar{v} + \epsilon$	1	λ_h
2	$v_\ell - \bar{v}$	$\frac{\kappa - \lambda_h}{\lambda_\ell}$	$(\lambda_\ell) p_\ell^c$

Table 10. Maximum Bids and quantity executed if the servicer exhausts capacity

Proof of Proposition 3

The high type bids $b_h^d = v_h - \bar{v}$ and is executed with probability 1.

The second highest type is executed with probability p_2^d, which satisfies incentive compatibility, that ensures that type 2 doesn't want to bid b_h^d.

$$\begin{aligned} v_\ell - b_h^d &\leq v_\ell - b_\ell^d - c_\ell \frac{1 - p_2^d}{p_2^d} \\ v_\ell - (v_h - \bar{v}) &\leq v_\ell - b_\ell^d - c_\ell \frac{1 - p_2^d}{p_2^d} \\ b_\ell^d &\leq (v_h - \bar{v}) - c_\ell \frac{1 - p_2^d}{p_2^d} \end{aligned}$$

Further, the highest type does not want to bid lower and experience waiting costs:

$$\bar{v} \geq v_h - b_\ell^d - c_h \frac{1 - p_2^d}{p_2^d} \quad (7)$$

$$b_\ell^d \geq v_h - \bar{v} - c_h \frac{1 - p_2^d}{p_2^d} \quad (8)$$

Subtracting these two equations from each other yields:

$$0 \leq (c_h - c_\ell)\Big(\frac{1 - p_2^d}{p_2^d}\Big)$$

Type	Bid	Execution Probability	Quantity
1	$v_h - \bar{v}$	1	λ_1
2	$v_\ell - \bar{v} - c_\ell\Big(\frac{\Delta_v}{\Delta_c}\Big)$	$\frac{\Delta_c}{\Delta_v+\Delta_c}$	$(m_2+\lambda_2)p_2^d$

Table 11. Bids and quantity executed if the servicer discriminates

As the RHS is strictly positive, these conditions are satisfied. To pin down b_ℓ^d, we note that

$$b_\ell^d \;=\; v_\ell - \bar{v} - c_\ell \frac{1-p_2^d}{p_2^d},$$

and use this in the incentive compatibility constraint Equation 7:

$$\begin{aligned}
\bar{v} &\geq v_h - b_\ell^d - c_h \frac{1-p_2^d}{p_2^d} \\
\bar{v} &\geq v_h - \Big(v_\ell - \bar{v} - c_\ell \frac{1-p_2^d}{p_2^d}\Big) - c_h \frac{1-p_2^d}{p_2^d} \\
0 &\geq \Delta_v - \Delta_c \frac{1-p_2^d}{p_2^d}
\end{aligned}$$

$$\begin{aligned}
\frac{1-p_2^d}{p_2^d} &\geq \frac{\Delta_v}{\Delta_c}, && (9) \\
\Longrightarrow p_2^d &= \frac{\Delta_c}{\Delta_v+\Delta_c} && (10) \\
1-p_2^d &= \frac{\Delta_v}{\Delta_v+\Delta_c} && (11)
\end{aligned}$$

In this case, the flow revenue of the servicer is:

$$\begin{aligned}
\pi^d &= b_h^d \lambda_h + b_\ell^d p_\ell^d(\lambda_\ell) \\
&= (v_h - \bar{v})\lambda_h + \Big(v_\ell - \bar{v} - c_\ell \frac{1-p_\ell}{p_\ell}\Big) p_\ell^d(\lambda_\ell) \\
&= (v_h - \bar{v})\lambda_h + \Big((v_\ell - \bar{v}) - c_\ell \frac{\Delta_v}{\Delta_c}\Big)\Big(\frac{\Delta_c}{\Delta_v + \Delta_c)}\Big)(\lambda_\ell) \\
&= (v_h - \bar{v})\lambda_h + \frac{(v_\ell - \bar{v})(\Delta_c) - c_\ell(\Delta_v)}{\Delta_v + \Delta_c}(\lambda_\ell)
\end{aligned}$$

■

Proof of Proposition 4

Consider the arrival rates of type h traders. Under discriminatory liquidity supply, the utility they receive is just $\bar{v}$, so they are indifferent between entering and not. Thus, any quantity $\lambda_h^d \subseteq [0, \kappa]$ is consistent with equilibrium. Whereas, for type h traders under the capacity liquidity supply, the traders make positive profits, so $\lambda_h^c = \kappa$.

For type ℓ traders under discriminatory liquidity supply, they receive zero utility, but the arrival rate that is consistent with a finite size of the mempool is determined by:

$$\begin{aligned}
m_\ell^d &= (m_\ell^d + \lambda_\ell^d)(1 - p_\ell^d) \\
m_\ell^d &= (m_\ell^d + \lambda_\ell^d)(1 - \frac{\Delta_c}{\Delta_v + \Delta_c}) \\
m_\ell^d &= (m_\ell^d + \lambda_\ell^d)\Big(\frac{\Delta_v}{\Delta_v + \Delta_c}\Big) \\
\\
\implies \lambda_\ell^d &= m_\ell^d \frac{\Delta_c}{\Delta_v}
\end{aligned}$$

Further, the total quantity executed must be consistent with the capacity of the system.

$$\begin{aligned}
(m_\ell^d + \lambda_\ell^d)\frac{\Delta_c}{\Delta_v + \Delta_c} &\leq \kappa - \lambda_h^d \\
\implies \lambda_\ell^d \leq \kappa - \lambda_h^d &
\end{aligned}$$

■