

# Optimizing fee estimation via the mempool state

Karl-Johan Alm <karl@dglab.com>  
DG Lab

October 19, 2017

## Abstract

Scalability in Bitcoin is just as much about throughput as it is about transaction fees. By optimizing the fee estimation algorithm in Bitcoin Core, users are able to bump high priority transactions without paying an overly high fee, and regular users will not feel the fee pressure to as great an extent as they are now, since the medium fee average is lower. Sending small amounts of bitcoin with the recommended fees will in many cases result in a fee that is close to the amount being sent, causing frustration among users, in particular as a lot of users are not aware that they ultimately set the fees themselves.

The current fee estimation in Bitcoin Core (0.15) attempts to find a fee which will nearly guarantee that the transaction goes into a block after the given target count. The estimator disregards the mempool completely, and will as a consequence grossly over-estimate the fees required in cases where the mempool is experiencing a big drop in transactions. By using the mempool as the source for a secondary estimation and setting the resulting fees to the minimum of the two estimations, simulations indicate that an over 80% cut in “overpayment” of fees can be achieved in these low-traffic situations.

Experiments were made where the stream of transactions going into or out of the mempool are recorded, as well as each block being mined; periodically, estimations are done using various combinations of modes.

While the results will in some cases undershoot, proper use of RBF could alleviate a potentially never-confirming transaction, and as such making use of mempool optimizations has great potential.

A consequence of optimizing fees is a drop in revenue for miners. Currently, miner revenue comprises about 11% of user fees, and with optimizations this number would drop even lower. With subsidy halvings every four years, an important matter to address is how miner profits and user costs correlate, in particular related to transaction throughput.

This report shows that a block size increase is not an adequate solution, as it will only minimally affect the miner revenue at the cost of less security, more centralization, and the incentivization of a remine attack in low peak situations.

# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Block size dynamics</b>	<b>4</b>
2.1	Block size, miner profit, and user cost . . . . .	4
2.1.1	10 BTC per block assumption . . . . .	5
2.1.2	\$65,000 USD per block assumption . . . . .	6
2.2	Interruptions and consequences of mempool exhaustion . . . . .	8
2.2.1	The for-profit and essentially free reorganization attack . . . . .	8
2.2.2	Mining cost vs transaction (fee) throughput . . . . .	9
2.3	A note on subsidy halving . . . . .	10
<b>3</b>	<b>Transactions and the mempool as a market</b>	<b>11</b>
3.1	Transaction throughput and mempool market . . . . .	11
3.1.1	Overly conservative fee estimators . . . . .	12
3.1.2	User interface . . . . .	13
<b>4</b>	<b>Fee rate optimization via the mempool state</b>	<b>14</b>
4.1	Automatic transaction bumping via RBF . . . . .	14
4.2	Combining fee rate estimators . . . . .	15
<b>5</b>	<b>Conclusions</b>	<b>17</b>
<b>A</b>	<b>Derivation of fee function given in Equation 2.3</b>	<b>19</b>
<b>B</b>	<b>Impact of subsidy halving</b>	<b>20</b>

## INTRODUCTION

This report investigates the viability of using the mempool to optimize fee rates for transactions in Bitcoin, as well as the consequences of measures to handle an increase in transaction throughput as well as a decrease in base miner profit due to subsidy halving.

It begins in Chapter 2 by looking at how the block size would change, given certain assumptions, and how this would relate to various constraints (2.1), such as a minimum required bitcoin per block fee for the miners (2.1.1), or a minimum profit expressed in USD (2.1.2), as well as how this would change with time as the subsidy decreases. The chapter ends with an examination of the consequences of a blockchain system where all transactions are mined at all times, and some important security related problems are derived (2.2).

In Chapter 3, the opposite approach is examined, where the mempool is considered a market.

In Chapter 4, a fee rate optimization using the mempool is discussed, and conclusions are drawn in Chapter 5.

## BLOCK SIZE DYNAMICS

### 2.1 Block size, miner profit, and user cost

Let  $\wp$  be the collection of  $N$  transactions  $\tau_0, \tau_1, \dots, \tau_N$  in the mempool at some given point in time.  $\wp^* \subseteq \wp$  is the representation of  $\wp$  for some node, which is seeing a subset of all the unconfirmed transactions. The two functions  $\omega(\tau_i)$  and  $\text{FR}(\tau_i)$  give the weight<sup>1</sup> and fee rate of the transaction  $\tau_i$ .

$$y = \sum_i \omega(\tau_i) \tag{2.1}$$

is the aggregate weight of all transactions in the mempool, and

$$y_{\text{FR} \leq x} = \sum_i \omega(\tau_i) \mathbb{1}(\text{FR}(\tau_i) \leq x) \tag{2.2}$$

is the aggregate weight of all transactions  $\tau_i$  which satisfy  $\text{FR}(\tau_i) \leq x$  for some  $x$ , i.e. all transactions with a fee lower than or equal to  $x$ .

Using a naïve, perfect knowledge no-RBF universal fee estimator<sup>2</sup>, the fee rate becomes

---

<sup>1</sup>SegWit unit for size; it is often approximated so that 1 byte = 4 weight units, even though this is slightly wrong.

<sup>2</sup>Naïve because it assumes no competing transactions will appear; perfect knowledge because it sees the entire mempool of the network; no-RBF because it would otherwise bump its fee repeatedly rendering the algorithm essentially worthless; universal because all other transactors on the network are assumed to use exactly the same algorithm.

$$f(y, Z) = 1.1^{\lfloor \frac{y}{0.95Z} \rfloor} \quad (2.3)$$

where  $Z$  is the block capacity in weight units.<sup>3</sup>  
 The per-block profit without subsidy becomes

$$\frac{1}{4}f(y, Z)Z \text{ satoshi} \quad (2.4)$$

which, for a just-full mempool<sup>4</sup> simplifies to

$$\frac{1}{4}Z \text{ satoshi.} \quad (2.5)$$

In other words, a 1 MB block will on average get 1 million satoshi  $\approx$  (today<sup>5</sup>) \$55 USD. Generalized, the miner profit  $m(Z)$  as a function of the block size  $Z$  becomes

$$m(Z) \approx \frac{\$55}{4000000}Z \quad (2.6)$$

(at current price levels).

The current miner profit from subsidy alone (i.e. excluding all transaction fees) is roughly \$65,000 USD.

We will return to  $f(y, Z)$  but before we do, we will take a look at two cases related to miner profit: the first case is for a fixed BTC income per block, and the second case is for a fixed USD income per block.

### 2.1.1 10 BTC per block assumption

We will now explore how bitcoin will change when the mempool is optimal w.r.t. the blocks, i.e. no transaction backlog, and every block filled to capacity.

If we make the assumption that miners desire 10 bitcoins per block after the next halving in  $Y = 2020$  (they get 12.5 as of this writing), we begin by defining

$$r(s) = 10.0 - s \quad (2.7)$$

as the fee required per block after subsidy, where the subsidy

$$s = \frac{12.5}{2^{\lfloor \frac{Y-2016}{4} \rfloor}} \quad (2.8)$$

---

<sup>3</sup>The way this function is derived is given in Appendix A on page 19.

<sup>4</sup>Transactions are always mined at the next block, and blocks are always exactly full.

<sup>5</sup>Taking 1 BTC = \$5500 USD

( $Y$  being the current year) giving us  $r_s(s) = 10^8 r(s)$  as the fee required in satoshi.

The satoshi/weight (assuming a transaction takes up 300 bytes = 1200 weight) becomes

$$F(p, c) = \frac{p}{1200} \cdot \frac{10^8}{c} = \frac{10^8 p}{1200c} \quad (2.9)$$

where  $p$  is the transaction fee in US dollars, and  $c$  is the current USD/BTC price. E.g. for  $p = \$0.20$ ,  $c = \$5500$  (20 cents at current price of \$5500 USD/BTC),  $F(p, c) = \frac{10^8 \cdot 0.2}{1200 \cdot 5500} = 3.03$  satoshi per weight unit.

The weight required at the given fee  $F(p, c)$  to achieve the goal of  $r_s(s)$  satoshi for the block is expressed as

$$Z(s, p, c) = \frac{r_s(s)}{F(p, c)} = \frac{1200(10.0 - s)c}{p} \quad (2.10)$$

With  $s = 6.25$  (2020 through 2023),  $p = \$0.20$ ,  $c = \$5500$ , we get that the necessary block size in bytes is

$$\frac{1}{4} Z(6.25, 0.20, 5500) = \frac{1}{4} \left( \frac{1200(10.0 - 6.25)5500}{0.20} \right) = 30937500,$$

i.e. roughly 30 MB.

This hits a maximum when  $s = 0$  (no subsidy) at roughly 80 MB.

If the users aren't willing to pay the 20 cents, the block size increases proportionately, where if  $p$  is halved, the block size is doubled (and conversely, if  $p$  is doubled, the block size is halved). If the price of bitcoin relative to USD increases, the block size increases in the same fashion, and if it drops, the block size drops. E.g. at \$1,000,000 USD/BTC, the block size would cap at  $\frac{1}{4} \cdot \frac{1200 \cdot 10 \cdot 1000000}{0.20} = 14$  GB (but the miners would get \$10 million USD per block).

### 2.1.2 \$65,000 USD per block assumption

We approach the problem from the point of view of a given revenue in USD<sup>6</sup>. We begin by rewriting  $r(s)$  defined in Equation (2.7) as

$$r(s, c) = \frac{65000}{c} - s \quad (2.11)$$

---

<sup>6</sup>Here we picked the current subsidy of 12.5 BTC in USD at \$5500 USD/BTC.

Avg fee (USD)	BLOCK SIZES FOR $s = 6.25$					FOR $s = 0$
	Price USD/BTC					Any price
	\$6k	\$7k	\$8k	\$9k	\$10k	$\$x$
\$0.10	77 MB	61 MB	43 MB	25 MB	7.2 MB	186 MB
\$0.20	39 MB	30 MB	21 MB	13 MB	3.5 MB	93 MB
\$0.50	16 MB	12 MB	8.9 MB	5 MB	1.4 MB	37 MB
\$1.00	7.9 MB	6.1 MB	4.3 MB	2.5 MB	0.7 MB	19 MB

Table 2.1: The block size for various average transaction fees vs the price of bitcoin, with a fixed subsidy  $s = 6.25$  (2020-2023) and (right-most column) when  $s = 0$ , at which point the price no longer matters.

e.g. for  $c = \$5500$  USD/BTC,  $s = 6.25$  BTC, we get  $r(s, c) \approx 5.57$  BTC. The miner revenue would simply be  $\frac{65000}{c} \approx 11.82$  BTC.

Inserting this into Equation (2.10) we get

$$W(s, p, c) = \frac{r_s(s, c)}{F(p, c)} = \frac{10^8 \left( \frac{65000}{c} - s \right)}{\left( \frac{10^8 p}{1200c} \right)} = \frac{1200}{p} (65000 - sc) \quad (2.12)$$

The block size  $Z(s, p, c) = \frac{1}{4}W(s, p, c)$  grows inversely proportionately to  $p$  (the price users are willing to pay in fees for a transaction), and shrinks with subsidy  $s$  and price of bitcoin  $c$ ; however, as the subsidy drops, the impact of the price of bitcoin drops proportionately until it stops having any effect<sup>7</sup>. Table 2.1 shows a matrix for the next subsidy halving ( $s = 6.25$ ) and (right-most column) for the case  $s = 0$ .

We can also display the case  $s = 0$  with a variable profit value, so that  $r(0, c, \omega) = \frac{\omega}{c}$ , as shown in Table 2.2.

<sup>7</sup>This is not entirely obvious, but since we are defining user fee in terms of USD, and miner profits in USD, the amount of satoshi paid for a transaction will change, but proportionately to the miner profit. The equality  $65000 \cdot \frac{1}{c} - s = \frac{p}{1200} \cdot \frac{1}{c} \cdot Z$  has  $\frac{1}{c}$  (BTC per USD) on both sides, which cancel out when  $s = 0$ , leaving  $65000 = \frac{p}{1200} \cdot Z$ , and  $Z = \frac{65000 \cdot 1200}{p}$ , which is unaffected by changes in  $c$ .



BLOCK SIZES FOR $s = 0$					
Avg fee (USD)	Miner profit threshold				
	\$20k	\$50k	\$100k	\$250k	\$500k
\$0.10	57 MB	143 MB	286 MB	715 MB	1.4 GB
\$0.20	29 MB	72 MB	143 MB	358 MB	715 MB
\$0.50	11 MB	29 MB	57 MB	143 MB	286 MB
\$1.00	5.7 MB	14 MB	29 MB	72 MB	143 MB

Table 2.2: The block size for various average transaction fees and various miner profit thresholds, given a zero subsidy ( $s$ ).

## 2.2 Interruptions and consequences of mem-pool exhaustion

With no subsidy left, a miner will not attempt to find a block unless the fees in the publicly known unmined transactions are at minimum higher than the electricity costs of running the mining equipment.

As such, the higher the block size, the more likely it is that the chain will have *interruptions*, where no block is being mined by anyone. Miners would switch their equipment to some competing chain with the same proof-of-work (if any such chain existed) or put the mining equipment in “economy mode” to minimize costs.

This minimum requirement is additionally different per region, because electricity cost is different. As such, this state will lead to centralization of mining power, to where a region with lower electricity costs will represent a disproportionately high portion of the hashpower.

### 2.2.1 The for-profit and essentially free reorganization attack

This presents a high security problem. It would be profitable for a miner to purposefully drop the last block (if it was mined by someone else) and remine it, to gain the transaction fees from it, while the rest of the network is waiting for transactions to satisfy the above requirement. The miner could then mine an extra block on top of his remined one and the two blocks would replace the current tip as the chain with the highest amount of work.

He could do this for an arbitrary number of blocks if he has enough

hashpower, because everyone else will be waiting for transactions to cover their expenses after every “honest” block.

The difficulty of the chain would drop as well, because miners were not mining blocks every ten minutes, further reducing its security, and further making the above attack more easy to pull off.<sup>8</sup>

To counter this attack, miners of a block would need to keep mining an empty block so that their chain tip remained protected. This would only work if no other malicious miner had more hashpower, and even if, the miner would be mining at a loss. To prevent this, miners would be enticed to form coalitions or enter into contracts with other miners to promise to protect each others’ blocks, leading to more centralization.

### 2.2.2 Mining cost vs transaction (fee) throughput

One big argument against the above scenario is that the cost to mine a block grows with time, and this growth only needs to be slower than or equal to the transaction (fee) throughput for the miner to be profitable. I.e. if a miner finds a block after a few seconds, the revenue would only need to cover the few seconds worth of electricity usage. Because electricity cost varies, this would mean some miners would profit from mining earlier than others. That aside, this depends on the arguably unstable condition that transaction (fee) throughput is *constant* over time, which it is not; it is experiencing highs and lows and this trend will most likely grow in scale with adoption. In other words, miners will still benefit from *not* mining during low peak periods, for the case where the blocks have sufficient capacity to include all transactions.

One might assume this is an argument for dynamic block sizes, but this would change nothing. A static block size that would fit all transactions at peak time would be just as effective even if only a fraction of its capacity were used during the low peaks.

Assuming Bitcoin is successful, one might argue that none of this would ever pose a problem, but if the blocks are large enough to include *all transactions at all times*, there will inevitably be periods of high-peak activity and periods of low-peak activity (VISA today averages roughly 2,000 transactions

---

<sup>8</sup>Assume that most miners need to wait 60 seconds before they are able to mine at a profit. The average time to find a block would then be 9 minutes, which would be a reduction of 10% in hashpower, and consequently a 10% reduction in network difficulty compared to the available hashpower.

per second, but has a capacity of 24,000<sup>9</sup>). It would only take one low-peak period to encourage the voluntary reorganization attack mentioned above by a miner, simply to maximize profit, the primary incentive that motivates them in the first place.

## 2.3 A note on subsidy halving

A perhaps unusual amount of emphasis is placed on the subsidy halving and its effect on the dynamics of Bitcoin scaling in this report. Since the subsidy will not vanish entirely until roughly the year 2143, this may seem like making a mountain out of a pebble. However, it should be noted that due to the nature of the subsidy halving process, the highest impact will be seen earlier — for instance, the subsidy will drop by a total of 596 satoshi in the entire final 43 year period 2100 - 2143; it will be lower than 0.001 bitcoin from 2072, and it will drop below 1 bitcoin per block as soon as 2032, 15 years from now. See Appendix B for additional details on how subsidy halving and fees correlate.

---

<sup>9</sup><https://usa.visa.com/run-your-business/small-business-tools/retail.html>

## TRANSACTIONS AND THE MEMPOOL AS A MARKET

This chapter takes the approach of looking at the mempool as a market, and describes issues with this approach, such as overly conservative fee estimators (3.1.1), and issues related to the user interface and user experience in software (3.1.2).

### 3.1 Transaction throughput and mempool market

We return to  $f(y, Z)$  originally defined in Equation (2.3) on page 5.

$$f(y, Z) = 1.1^{\lfloor \frac{y}{0.95Z} \rfloor} \quad (3.1)$$

We define a transaction throughput  $\Delta y$  given in weight per block<sup>10</sup>. Assuming no mempool backlog,

- If  $\Delta y < Z$ , the next block will not be full. It will be at  $\frac{\Delta y}{Z}$  capacity where  $y = \Delta y$ , and the miner will get  $\frac{1}{4}\Delta y$  satoshi according to Equation (2.4).
- If  $\Delta y = Z$ , the next block will be full and miner simply receives  $\frac{1}{4}Z$  satoshi.

---

<sup>10</sup>Strictly speaking, it is weight per 10 minute interval, disregarding discrepancies in the time between blocks, which varies quite a bit.

- If  $\Delta y > Z$ , the next block will be full. There will be a growing mempool backlog for as long as this remains the case, where  $y_{t+1} = y_t + \Delta y_t - Z$ .

In the last case above, we have appended a time subscript  $t$  to the mempool size  $y$  and the corresponding transaction throughput  $\Delta y$ . In the simplest case,  $\Delta y$  remains fixed for all  $t$ , and  $y_{t+1}$  simply becomes  $(\Delta y - Z)t$ , after  $t$  blocks.

As  $t \rightarrow \infty$ , so does  $y \rightarrow \infty$ , and consequently  $f(y, Z) = 1.1^{\lfloor \frac{y}{0.95Z} \rfloor} \rightarrow \infty$ . In other words, we approach infinitely high fees if the transaction throughput per block exceeds the block capacity.

We note here that, if RBF was enabled, every participant would repeatedly bump the fee of their transaction as soon as  $y = Z$  in order to beat the bottom 5% threshold. Even at  $t = 0$ , this bumping would continue without bounds moving toward  $\infty$  in the same fashion, only faster. The only real reason why  $t$  matters at all is because each block has a number of ever growing “losing” transactions in the mempool that were created too early to catch the threshold.

As the fee rate increases, our willingness to transact decreases. It becomes clear that  $f(y_t, Z)$  breaks down as  $\Delta y_t \geq Z$ .

We redefine this to  $f(x, y, Z) = \min(x, f(y, Z))$ , where  $x$  is the maximum fee rate the user is willing to pay. We get bundles of transactions sorted by  $x$  in descending order, so that  $f(x, y, Z) \in \mathbf{B}(x)$ . It could be expected that rational miners will iterate through each bundle in  $\mathbf{B}$  and add transactions until the block is full, or until they exhaust  $\mathbf{B}$ .

We can imagine grouping  $x$  into ranges like “emergency”, “high priority”, “normal priority”, and “low priority”, where the ranges vary depending on what people are paying in fees right now.

This works in most cases, but has its problems.

### 3.1.1 Overly conservative fee estimators

One main problem is when fee estimators are suboptimal and/or overly conservative. With RBF support, the ability to fine tune fee rate is better than ever, but this is not taken advantage of by most, if any, wallets at this time.<sup>11</sup>

Let us define a function  $\Delta f_t(y_t, Z)$  given in satoshi / weight / time unit  $t$ , for some fee rate estimator  $f(y, Z)$  not necessarily equal to that in Equation

---

<sup>11</sup>Bitcoin Core 0.15 will use economic fee estimation for transactions which have RBF set, but that’s the only case we’ve seen so far.

## 2.4.

Assuming perfect knowledge, a fee estimator may pick a fee rate close to, but not below, the 5% bottom threshold of the next block. Given  $y_t = \sum_{i=0}^t \Delta y_i$ , we will see a fee increase of

$$\Delta f_t(y_t, Z) = f(y_t, Z) - f(y_{t-1}, Z) \quad (3.2)$$

at time  $t$ . However, we have a maximum fee rate  $x$  on a per (user) transaction basis, so the fee rate is no longer unbounded. The rate at which it grows determines the resulting fee rate that a user ends up paying for a given transaction. As such, users want  $\Delta f_t(y_t, Z)$  to be as low as possible, whereas rational miners want it to be as high as possible. A lower  $\Delta f_t(y_t, Z)$  means an overall lower fee, and fewer users hit their cap  $x$  before their transactions are mined.

### 3.1.2 User interface

Another problem is that a lot of users don't realize they actually set the fee rate themselves. This ties in with the previous problem, in where wallets tend to be overly conservative in their estimates<sup>12</sup>. Better education of users and better UI in wallet software would possibly remedy this problem, especially if wallets turned RBF on by default<sup>13</sup>.

We will expand on this in the next chapter.

---

<sup>12</sup>Possibly to prevent a storm of angry users asking why their transactions are not being confirmed.

<sup>13</sup>This would allow users to fix "too low fee" mistakes, but they could still end up paying a very high fee by mistake, in which case RBF would not help.

## FEE RATE OPTIMIZATION VIA THE MEMPOOL STATE

This chapter evolves on the concept of using RBF (replace-by-fee) in wallet software in a more automatic and user friendly way, where users set a maximum fee, rather than an explicit one (4.1), then goes into detail on how to combine the current fee rate estimators which use block statistics, with a fee rate estimator using the mempool (4.2).

### 4.1 Automatic transaction bumping via RBF

A natural extension to enabling RBF by default would be for wallet software to periodically recheck the mempool and bump the transaction fee automatically. A drawback of this is that the wallet software needs to remain active even after sending the transaction, but it may be sufficient for it to do the recheck-and-bump whenever it is brought to the foreground, as users will most likely bring the application up to check on the transaction, especially if it is urgent enough to warrant fine-tuning.

$$\text{SHOULDBUMP}(T, t, Z) = tZ < y_{\text{FR} \geq \text{FR}(T)} \quad (4.1)$$

For instance, if a user wants a transaction to confirm within the next  $t$  blocks (“target”), the wallet software may choose to bump the fee if the accumulated weight of all transactions above itself in the mempool exceed

$t$  blocks worth of weight (Equation (4.1), where  $T$  is the transaction being bumped,  $t$  is the target (in blocks to confirm), and  $Z$  is the block size).

Using Equation (4.1) it is trivial to derive a method for finding an optimally minimal fee bump (Equation (4.2), where  $T_i$  represents a bumped form of the transaction  $T$  where the fee has been increased by  $i$ ).

$$\begin{aligned} \text{FEEDELTA}(T, t, Z, x) &= \min(i) && (4.2) \\ \text{provided } \neg\text{SHOULDBUMP}(T'_i, t, Z) \vee \text{FR}(T'_i) &\geq x \\ \text{where } T'_i &= \{T; \text{FR}(T'_i) = \text{FR}(T) + i\} \end{aligned}$$

A wallet would make periodic checks at appropriate times by calling FEEDELTA on each unconfirmed transaction; if the resulting value is greater than 0, the transaction is bumped. Since the user has given a maximum acceptable fee, the transaction will never exceed acceptable values, no matter what happens (unless the user increases the maximum).

Note that even a full node will not see the entire mempool, but a subset of it. This affects the accuracy of FEEDELTA and SHOULDBUMP proportionately.

## 4.2 Combining fee rate estimators

In reality, at least in the case of Bitcoin Core, wallets do not use the mempool state *at all* when estimating fees, contrary to this report which has *only* been using the mempool up until this point. Instead, most wallets look at past blocks and use statistics to determine a fairly reliable fee rate that would put the transaction in the next block. By using the mempool and “pretending” to make a block from the observable transactions, a wallet can get a rough representation of what the next block will look like, depending on how much time passes until the next block is found.

This assumes that miners are rational and open, but there is no basis for such an assumption. It is highly probable that miners take out-of-band payments to include specific transactions, which would bypass the mempool market.

The mempool state for a given node is just a local state of the actual mempool, sort of like an approximation of what everyone else is seeing. There is some delay in transaction relay, and as such, the mempool is not “perfect knowledge”.



The mempool can be manipulated, and is by nature highly volatile<sup>14</sup>. Using only the mempool to estimate fees is as such not recommended. Instead, the fee estimation from the mempool state should be given as a lower boundary to the regular fee estimation function(s) used, as in Equation (4.3).

$$f(\cdot) = \min(f_{\text{MPOPT}}(\cdot), f_{\text{SMARTEST}}(\cdot)) \quad (4.3)$$

In Bitcoin Core, the fee rate estimation (here called  $f_{\text{SMARTEST}}(\cdot)$ ) was improved<sup>15,16</sup> in version 0.15. It currently uses three horizons ( $\frac{t}{2}$ ,  $t$ , and  $2t$ , where  $t$  is the number of blocks until the transaction should be confirmed), each with a different success rate (60%, 85%, and 95% respectively).

Each of these estimations is based on fee rate buckets, which are used to keep track of when transactions in a given fee rate range were mined. There are two modes; conservative, which has an additional requirement of 95% success rate over a longer time horizon, and economic, which is relieved of this requirement.

This works well under normal circumstances, but tends to result in overly high estimations when the transaction throughput drops rapidly, i.e. when the last couple of blocks have relatively high fees, but the mempool is relatively empty.

Using Equation (4.3) would in many cases dampen this effect.

---

<sup>14</sup>See e.g. [https://www.reddit.com/r/Bitcoin/comments/764nt7/be\\_warned\\_ledger\\_nano\\_s\\_transaction\\_accelerator/?st=J8QFCAAU&sh=3e5e642a](https://www.reddit.com/r/Bitcoin/comments/764nt7/be_warned_ledger_nano_s_transaction_accelerator/?st=J8QFCAAU&sh=3e5e642a) in which a user is recommended a \$7k fee by his wallet due to mempool optimizations gone awry.

<sup>15</sup><https://github.com/bitcoin/bitcoin/pull/10199>

<sup>16</sup><https://gist.github.com/morcos/d3637f015bc4e607e1fd10d8351e9f41>

## CONCLUSIONS

The assumption that all transactions are mined at all times leads to a number of problems:

- We end up with all transactions using the minimum fee rate<sup>17</sup>. At the current price levels, miners would get \$55 USD per MB of block size per block, assuming there were enough users to fill the block.
- To retain the equilibrium in terms of bitcoins per block (e.g 10 bitcoin per block), we would need 30 MB blocks by 2020 and 80 MB blocks when subsidy halving is completely replaced by fees. As the price of bitcoin rises, the minimum fee rate would decrease, and as a result the block size requirement would increase; at \$1 mln USD/BTC, each block would take 14 GB. Even the most optimistic (and compact) 30 MB size would severely increase centralization and decrease stability and accessibility to the Bitcoin network.
- To retain the equilibrium in terms of miner profit (roughly \$65,000 USD per block), we would need 77 MB blocks at the next halving if users paid 10 cents per transaction, although this would drop with the rising price of bitcoin, e.g. to 7.2 MB for \$10k USD/BTC. This drop is tied to the subsidy halving, however, and would ultimately fall off completely,

---

<sup>17</sup>There is no reason for users to pick a fee higher than the minimum, if there is no risk of the transaction being “bumped” by a higher-paying transaction. For sufficiently large blocks, everybody ends up paying the minimum fees until such a time as the transaction throughput beginning to exceed the block size.

leaving blocks at 186 MB, regardless of USD/BTC price. If miners required a higher profit, the block size would increase proportionately; e.g. at a \$100k USD/block profit, the block size would be nearly 300 MB.

- Arguably most problematic of all is, for the case with a low enough subsidy, if there are not enough transactions in the mempool at the time, there is no reason for miners to find a new block, and the blockchain will effectively grind to a halt until enough value has accumulated in the mempool. This results not only in higher centralization and unpredictability, but also renders the premise of “most work secures the chain” ineffective, because miners would now profit from remaining the chain tip block(s) while waiting, and it would decrease the difficulty, lowering the overall security of the chain.

Removing this assumption alleviates most of these issues, but results in a competitive mempool market with “winners” and “losers”:

- Transactions will approach a fee rate individually defined by their respective creators. The miner profit would vary, depending on mempool size, fee rate estimator precision and accuracy, and value of transacting. Users would weigh speed to be confirmed vs cost, just as they are now.
- The bitcoins per block received by miners would reach an equilibrium related to the aggregate value of transacting, as well as the transaction throughput. It would not directly satisfy miner profit requirements, but it would be far better than “minimum fee rate  $\times$  weight”.
- The mempool would very seldomly be empty, if ever, removing the problem with miners “pausing” their equipment due to a lack of profitable transactions once the subsidy has become a less significant part of the miner profits.

A full mempool results in a number of issues, some of which can be alleviated to a certain extent by optimizing fee rate estimators, and educating users on how fees work, e.g. via improved UI elements in wallet software.

A high fee rate would lead to a decrease in adoption rate, but with layer 2 technology like Lightning Network, higher fees would be more acceptable, as they go from per-transaction fees to per-charge fees over many transactions.

## DERIVATION OF FEE FUNCTION GIVEN IN EQUATION 2.3

$$f(y, Z) = 1.1^{\lfloor \frac{y}{0.95Z} \rfloor}$$

- When a user creates a new transaction, they look at the mempool and finds the fee that would put their transaction at the bottom 5% point to minimize their cost, with a minimum 10% increase in fees compared to the closest competing transaction at the threshold.
- If no transaction exists ( $y = 0$ ) or if the mempool is smaller than 95% of a block ( $y < 0.95Z$ ), the transaction uses a 1 satoshi per byte fee (the minimum required to be relayed by nodes, at this point in time).
- The above condition applies for the first  $0.95Z$  weight units, after which the user will pick a 10% higher fee than the competing transaction (which is initially at 1.0 sat/b), i.e.  $1.0 \cdot 1.1 = 1.1$  ( $y = 0.95Z$ ).
- After an additional  $0.95Z$  weight units ( $y = 2 \cdot 0.95Z$ ), we have filled the available 95% with 1.1 fee rate transactions, and as such, a new transactor will pick a 10% higher fee, i.e.  $1.1 \cdot 1.1 = 1.1^2$ .
- We apply induction to derive the generalized  $f(y, Z) = 1.1^{\lfloor \frac{y}{0.95Z} \rfloor}$ .

## IMPACT OF SUBSIDY HALVING

Looking at miner revenue per block for the last year<sup>18</sup>, the revenue is around 14 bitcoin per block, with peaks around 16.5 and has a lowest point around 12.8.

Table B.1 shows how big a portion of miner fees are made up by the subsidy. It assumes that the miners will receive the same amount of bitcoin per block, i.e. 14 bitcoin. While this assumption is *most likely incorrect*, it provides us with an example of how things will evolve, if the miner revenue remains somewhat stable. The table shows that less than 11% of miner fees are from transaction fees. In the next subsidy halving in three years in 2020, this will jump by roughly 45%. At the next halving in 2024, it will be over 77%, and the subsidy will make up less than 10% come 2032.

Table B.2 removes the assumption of bitcoin revenue and instead assumes miners receive approximately \$65,000 USD per block instead, and also assumes that fees make up 11% of the total revenue (with a minimum cap of total profit set to 0.01 BTC). Again, it must be noted that these assumptions are arbitrary, and only serve to give an idea of how things would evolve during the given parameters. The bitcoin price that must follow for this to apply is listed as well in the right-most column. The lower limit and the price requirement ends up capping the bitcoin price at \$6.5 million USD. In this scenario, the user fees still comprise over 50% of the miner revenue, although this begins at a later date (2064, 47 years in the future); this is based on the 0.01 BTC minimum assumption, though, which if adjusted will move

---

<sup>18</sup><https://www.smartbit.com.au/charts/miner-revenue-per-block>

Subsidy	Year	Bitcoin in fees	% Fees
12.50000000	2016	1.50000000	10.71429%
6.25000000	2020	7.75000000	55.35714%
3.12500000	2024	10.87500000	77.67857%
1.56250000	2028	12.43750000	88.83929%
.78125000	2032	13.21875000	94.41964%
.39062500	2036	13.60937500	97.20982%
.19531250	2040	13.80468750	98.60491%
.09765625	2044	13.90234375	99.30246%
.04882813	2048	13.95117188	99.65123%
.02441406	2052	13.97558594	99.82561%
.01220703	2056	13.98779297	99.91281%
.00610352	2060	13.99389648	99.95640%
.00305176	2064	13.99694824	99.97820%
.00152588	2068	13.99847412	99.98910%
.00076294	2072	13.99923706	99.99455%
.00038147	2076	13.99961853	99.99728%
.00019073	2080	13.99980927	99.99864%
.00009537	2084	13.99990463	99.99932%
.00004768	2088	13.99995232	99.99966%
.00002384	2092	13.99997616	99.99983%
.00001192	2096	13.99998808	99.99991%
.00000596	2100	13.99999404	99.99996%
.00000298	2104	13.99999702	99.99998%
.00000149	2108	13.99999851	99.99999%
.00000075	2112	13.99999925	99.99999%
.00000037	2116	13.99999963	100.00000%
.00000019	2120	13.99999981	100.00000%
.00000009	2124	13.99999991	100.00000%
.00000005	2128	13.99999995	100.00000%
.00000002	2132	13.99999998	100.00000%
.00000001	2136	13.99999999	100.00000%
.00000001	2140	13.99999999	100.00000%
.00000000	2144	14.00000000	100.00000%

Table B.1: Portion of miner profit from subsidy vs fees, assuming a steady revenue of 14 bitcoin per block.

Subsidy	Year	Fees	Revenue	% Fees	BTC price
12.50000000	2016	1.54494375	14.04494375	11.00000%	\$4,628
6.25000000	2020	.77247188	7.02247188	11.00000%	\$9,256
3.12500000	2024	.38623594	3.51123594	11.00000%	\$18,512
1.56250000	2028	.19311797	1.75561797	11.00000%	\$37,024
.78125000	2032	.09655898	.87780898	11.00000%	\$74,048
.39062500	2036	.04827949	.43890449	11.00000%	\$148,096
.19531250	2040	.02413975	.21945225	11.00000%	\$296,192
.09765625	2044	.01206987	.10972612	11.00000%	\$592,384
.04882813	2048	.00603494	.05486306	11.00000%	\$1,184,768
.02441406	2052	.00301747	.02743153	11.00000%	\$2,369,536
.01220703	2056	.00150873	.01371577	11.00000%	\$4,739,072
.00610352	2060	.00389648	.01000000	38.96484%	\$6,500,000
.00305176	2064	.00694824	.01000000	69.48242%	\$6,500,000
.00152588	2068	.00847412	.01000000	84.74121%	\$6,500,000
.00076294	2072	.00923706	.01000000	92.37061%	\$6,500,000
.00038147	2076	.00961853	.01000000	96.18530%	\$6,500,000
.00019073	2080	.00980927	.01000000	98.09265%	\$6,500,000
.00009537	2084	.00990463	.01000000	99.04633%	\$6,500,000
.00004768	2088	.00995232	.01000000	99.52316%	\$6,500,000
.00002384	2092	.00997616	.01000000	99.76158%	\$6,500,000
.00001192	2096	.00998808	.01000000	99.88079%	\$6,500,000
.00000596	2100	.00999404	.01000000	99.94040%	\$6,500,000
.00000298	2104	.00999702	.01000000	99.97020%	\$6,500,000
.00000149	2108	.00999851	.01000000	99.98510%	\$6,500,000
.00000075	2112	.00999925	.01000000	99.99255%	\$6,500,000
.00000037	2116	.00999963	.01000000	99.99627%	\$6,500,000
.00000019	2120	.00999981	.01000000	99.99814%	\$6,500,000
.00000009	2124	.00999991	.01000000	99.99907%	\$6,500,000
.00000005	2128	.00999995	.01000000	99.99953%	\$6,500,000
.00000002	2132	.00999998	.01000000	99.99977%	\$6,500,000
.00000001	2136	.00999999	.01000000	99.99988%	\$6,500,000
.00000001	2140	.00999999	.01000000	99.99994%	\$6,500,000
.00000000	2144	.01000000	.01000000	100.00000%	\$6,500,000

Table B.2: Bitcoin fees and price given a set revenue of \$65,000 USD per block, where fees are locked at 11%, with a minimum revenue of 0.01 BTC (which is taken from fees, which is why they begin to rise in 2060).

the point in time at which this takes effect. E.g. with a 1 bitcoin minimum assumption, the fees would comprise over 50% of the miner revenue from the 2036 halving in 19 years.

Table B.3 shows the scenario where the fees grow by 5% each subsidy halving, and how this would affect e.g. the bitcoin price and revenue in terms of bitcoin. Again note that the \$65,000 USD per block and 5% growth assumptions are arbitrary. The price peaks, and the revenue hits its lowest point, in 2040, at which point the 5% increase in fees outpaces the revenue lost from subsidy halving.



Subsidy	Year	Fees	Revenue	% Fees	BTC price
12.50000000	2016	1.54494375	14.04494375	11.00000%	\$4,628.00
6.25000000	2020	1.62219094	7.87219094	20.60660%	\$8,256.91
3.12500000	2024	1.70330048	4.82830048	35.27743%	\$13,462.29
1.56250000	2028	1.78846551	3.35096551	53.37165%	\$19,397.39
.78125000	2032	1.87788878	2.65913878	70.62019%	\$24,444.00
.39062500	2036	1.97178322	2.36240822	83.46497%	\$27,514.30
.19531250	2040	2.07037238	2.26568488	91.37954%	\$28,688.90
.09765625	2044	2.17389100	2.27154725	95.70089%	\$28,614.86
.04882813	2048	2.28258555	2.33141368	97.90564%	\$27,880.08
.02441406	2052	2.39671483	2.42112889	98.99162%	\$26,846.98
.01220703	2056	2.51655057	2.52875760	99.51727%	\$25,704.32
.00610352	2060	2.64237810	2.64848162	99.76955%	\$24,542.36
.00305176	2064	2.77449701	2.77754876	99.89013%	\$23,401.93
.00152588	2068	2.91322186	2.91474774	99.94765%	\$22,300.39
.00076294	2072	3.05888295	3.05964589	99.97506%	\$21,244.29
.00038147	2076	3.21182710	3.21220857	99.98812%	\$20,235.30
.00019073	2080	3.37241845	3.37260919	99.99434%	\$19,272.91
.00009537	2084	3.54103937	3.54113474	99.99731%	\$18,355.70
.00004768	2088	3.71809134	3.71813903	99.99872%	\$17,481.86
.00002384	2092	3.90399591	3.90401975	99.99939%	\$16,649.51
.00001192	2096	4.09919571	4.09920763	99.99971%	\$15,856.72
.00000596	2100	4.30415549	4.30416145	99.99986%	\$15,101.66
.00000298	2104	4.51936327	4.51936625	99.99993%	\$14,382.55
.00000149	2108	4.74533143	4.74533292	99.99997%	\$13,697.67
.00000075	2112	4.98259800	4.98259875	99.99999%	\$13,045.40
.00000037	2116	5.23172790	5.23172827	99.99999%	\$12,424.19
.00000019	2120	5.49331430	5.49331448	100.00000%	\$11,832.56
.00000009	2124	5.76798001	5.76798010	100.00000%	\$11,269.11
.00000005	2128	6.05637901	6.05637906	100.00000%	\$10,732.49
.00000002	2132	6.35919796	6.35919799	100.00000%	\$10,221.41
.00000001	2136	6.67715786	6.67715787	100.00000%	\$9,734.68
.00000001	2140	7.01101575	7.01101576	100.00000%	\$9,271.12
.00000000	2144	7.36156654	7.36156654	100.00000%	\$8,829.64

Table B.3: Bitcoin fees and price given a set revenue of \$65,000 USD per block, where fees grow by 5% each subsidy halving.