An Examination of Single-Transaction Blocks and Their Effect on Network Throughput and Block Size

Andrew Stone
g.andrew.stone@gmail.com
www.bitcoinunlimited.info

Abstract. The Bitcoin network's creation of normal blocks and single (coinbase-only) transaction blocks is analyzed using multiple empirical approaches. A maximum Bitcoin network transaction commitment throughput of 60KB/sec is derived. The significant role of single-transaction blocks in limiting this throughput is then shown. The miner revenue equation in an unlimited-block environment is derived, and it is shown that the optimum strategy for mining pools is to mine competing small blocks when presented with a block that is so large that its validation time will affect fee revenue. It is shown how this strategy naturally discourages large block sizes as a function of transaction throughput, coinbase reward and average transaction fees, and how is encourages larger blocks as fees increase, but in an asymptotic manner. In fact given today's network metrics, typical transaction fees of 0.1 to 0.4 BTC/MB actually discourage block growth, and the optimum-profit block size will not exceed about 30MB regardless of fee spent.Therefore, the choice of block size is a de-facto hash-power weighted "vote" controlling average block size and can replace proposed schemes that use explicit voting and/or flexible capacity.

1 Introduction: The mystery of the 1 transaction block

To produce a valid block, a miner must include the hash of the prior block in the blockchain, a coinbase transaction, and may include any transactions that are not in any ancestor blocks. Since miners gain fees from included transactions, at first glance it makes sense to include as many transactions as possible. However, miners sometimes only include the coinbase transaction even though there are uncommitted transactions available in the network. Understanding why this occurs offers a technique to analyze the network.

When a new block is discovered by a miner on the network, five steps must be undertaken before another miner can mine on top of it:

- The block must be propagated to the mining pool.
- The pool must validate the proof-of-work and each transaction in the block
- The pool must update his mempool by removing the transactions included in the block
- The pool can then create a new block candidate using the block's hash and the remaining transactions in its memory pool.
- This candidate block must be submitted to the pool's hashing infrastructure to begin the mining process.

This process is time consuming; there is network latency and bandwidth limiting propagation speed, and validation requires intense CPU use and possible disk access.

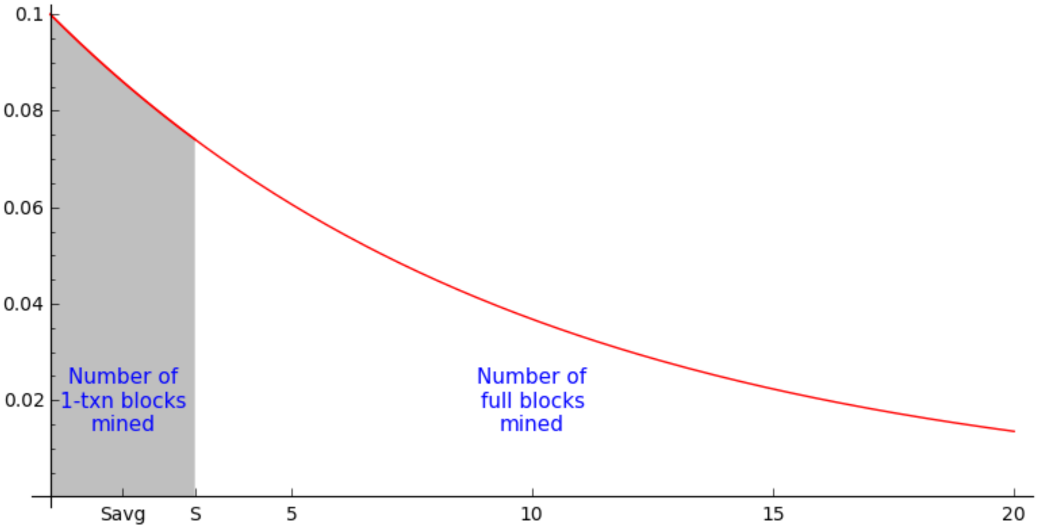
However, there is a shortcut. If a mining pool constructs a block candidate containing only the coinbase transaction (and transactions that it is certain cannot exist in the newly mined block[1]), it only needs the hash of the prior block. At 256 bits, this hash can be transmitted across the network extremely quickly. Since mining pools maximize profit by maximizing the time their ASICs are hashing blocks likely to be added to the chain, some pools use this technique to construct a single-transaction block candidate to mine while they are waiting to fully receive and validate the newly found block. When the block is validated, mining pools typically use the available transactions to construct a block candidate that maximizes profitability[1] and then switch their ASICs to mine that candidate2.

Therefore, by examining relationships between the 1 transaction and many-transaction blocks, we can discover properties of the Bitcoin network.

2 Observations

Data was collected and analyzed from approximately the middle of October 2014 until 19 November 2015. Figure 1 will help to illustrate the observation methodologies:

Figure 1: Block discovery interval showing 1-txn and full block regions



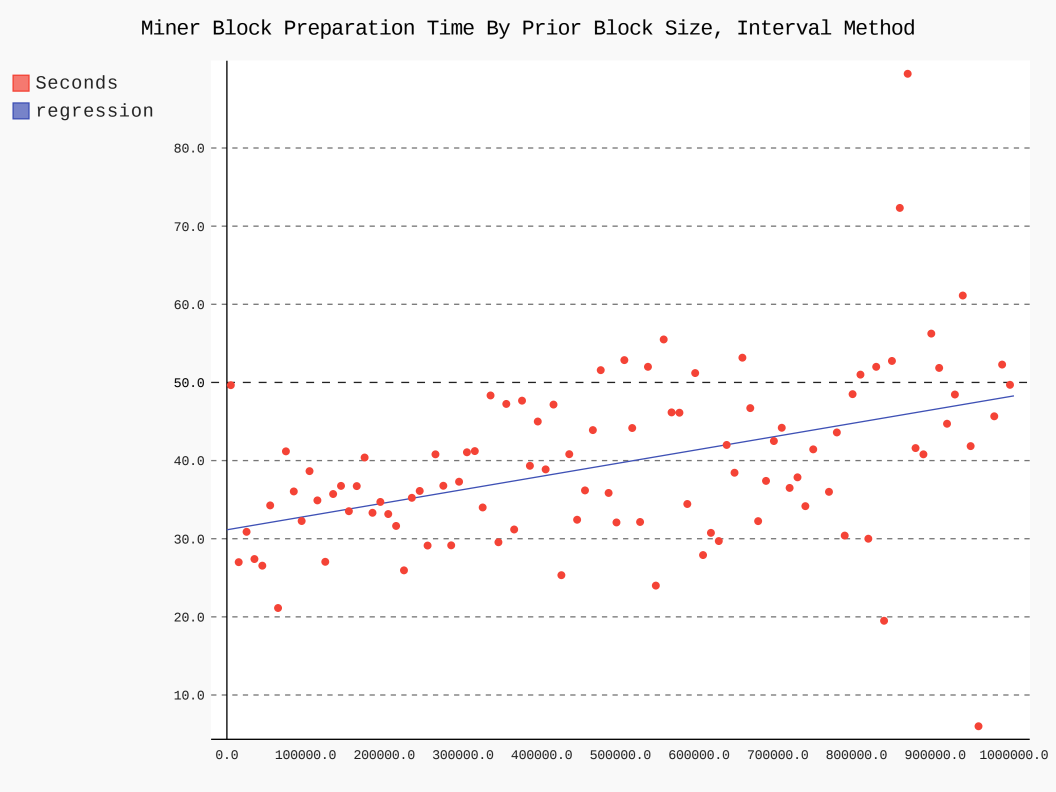
The red line marks the probability that a block is found at a particular time (shown in minutes on the horizontal axis). That this is not constant can be unintuitive to people who know that the likelihood of a block being found at any random time t (given constant hash rate) is the same as any other time. However, this is not what is being measured. We are measuring the likelihood that NO block was found during the interval [0, *t*) combined with the likelihood of finding a block at time *t*. This is the Poisson process equation:

$$f = \lambda e^{-\lambda t} \qquad \textbf{(1)}$$

λ refers to the inverse of the expected time for an event occurence. In this paper we will ignore difficulty adjustments and so λ is 1/the average block discovery time or 1/10 minutes.

In the first data set, the time interval between mining a 1 transaction block and its predecessor was compared to the size of the predecessor block. It is hypothesized that larger blocks will create a longer time for empty blocks to be mined (*S* in the figure). Therefore the average time interval (Savg in the figure) will increase as the switchover time *S* increases.

Figure 2:



In this figure, the X axis is the block size, ranging from 0 to 1MB and the y axis is seconds

Unfortunately blocks are time-stamped by their creator, and CPU clocks are not synchronized across the Bitcoin network so this dataset has significant errors. For example, there are many blocks that claim they were mined before their predecessor, which is causally impossible. To clean up the data, negative time intervals and intervals greater than 120 seconds were ignored. If time errors follow a symmetric distribution, this clean up effectively shifts the reported average time higher because all the negative errors are eliminated, yet positive errors remain.

Another methodology exists that does not rely on the block time stamp. The second graph was generated by observing the ratio (R) of the grey area in figure 1 to the total number of samples for particular block sizes. This can be used to determine S. Since the chance of finding a block at a particular time follows the poisson process equation R is:

$$R = \frac{\int_{t=0}^{S} \lambda e^{-\lambda t}\,\mathrm{dt}}{\int_{t=0}^{\infty} \lambda e^{-\lambda t}\,\mathrm{dt}} \qquad \textbf{(2)}$$

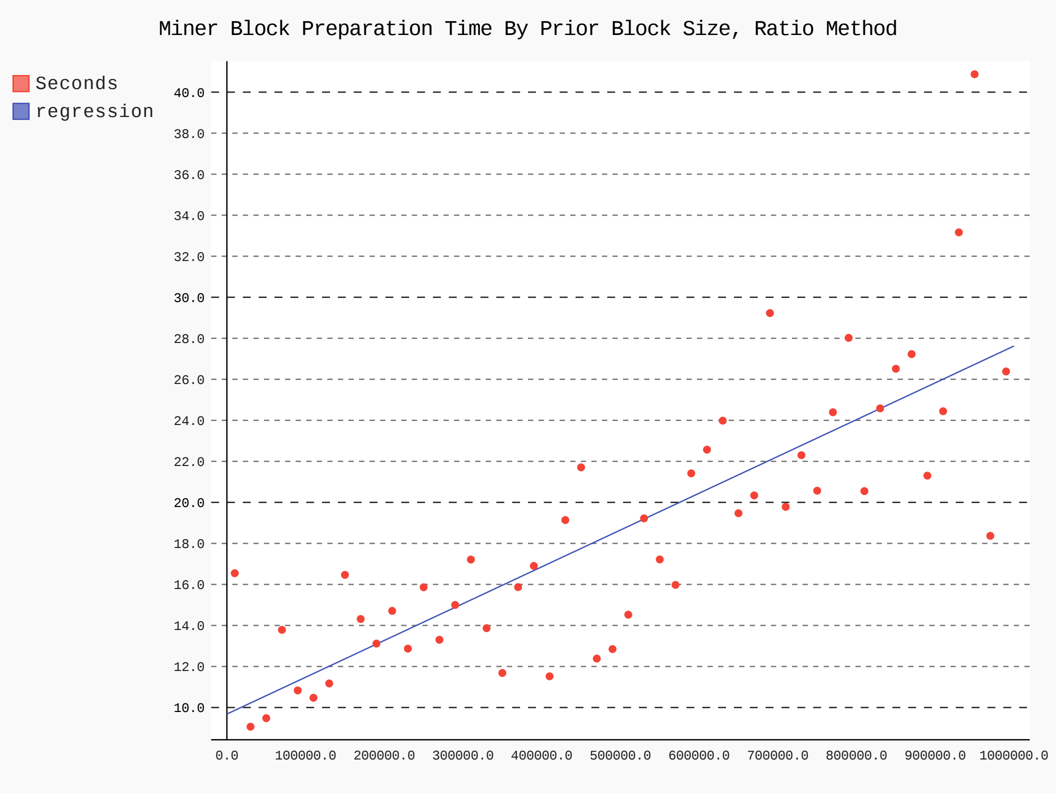
The denominator of this equation is simply 1 because $\int_{t=0}^{\infty} \lambda e^{-\lambda t}\,\mathrm{dt} = 1$ for all $\lambda > 0$. This makes intuitive sense; the sum of all the probabilities of finding a block must be 1 because given infinite time a block will always be found. Solving equation 2 for S yields:

$$S = \frac{1}{\lambda} * \log\frac{1}{1-R} \qquad \textbf{(3)}$$

To understand this equation, note that $\log\frac{1}{1-R}$ is approximately R for R < 0.1. One can therefore see that this equation is approximately equal (given Bitcoin's expected block discovery of 10 minutes or λ = 1/600) to the much more intuitive equation $S = 600\text{sec} * \frac{\text{number of 1 transaction blocks}}{\text{total samples}}$ that could be used if the block discovery probability was constant. This approximation was NOT used in the following charts, it is only included here expository reasons.

In the following chart, samples were divided into 50 bins ranging from 0 to 1MB corresponding to the size of the prior block (block sizes in each bin can vary by 20000 bytes) so that the ratio of 1-txn blocks to all blocks could be taken. Without some kind of binning (or more accurately if the bin size was 1 byte), millions of samples would be necessary to calculate ratios. Sequential 1-txn blocks were handled specially, please see Appendix 1 for details. Given the samples in a particular block size range, R was calculated and then equation 3 was applied to find S. This results in the following chart:

Figure 3:



3 Analysis

Figures 2 and 3 appear to indicate a relationship between the length of time miners generate empty blocks and the size of the preceding block. To determine whether the relationship was statistically significant, we calculated the probability that the *no* relationship existed and that the resulting data points were a result of randomness (the null hypothesis). The p-values for each method were 6.2 x 10^-15 and 1.9 x 10^-12, respectively, allowing us to confidently reject the null hypothesis. Indeed, there is a significant relationship between prior block size and the frequency of empty blocks

Therefore the slope of the regression lines in these two graphs indicate the relationship between prior block size and the length of time 1-txn blocks are generated. The slope in Figure 2 is 17.14 sec/MB, and in Figure 3 is 17.94 seconds/MB. Figure 2 shows a ~30 second baseline cost to move network-wide mining infrastructure from one block to the next one. However, this result is likely positively biased because negative time intervals were dropped. Figure 3 shows an approximate 10 second cost to move network-wide mining infrastructure from one block to the next one. It cannot be determined from the data why this is the case.

While this data shows a clear relationship between block size and generation of 1-txn blocks, unfortunately the quality of the data does not allow us to eliminate the possibility that it may actually be nonlinear[3]. Given the data collection errors, modifications to the Bitcoin client to produce more accurate data is necessary to fully investigate this relationship.

Effect on Network Throughput

Since the network cannot typically produce transaction-committing blocks until the prior blocks are validated, the network's transaction bandwidth is limited in an upper bound by its ability to validate these blocks. This idea was first proposed in [8]. But it is a common conceptual error to imagine that discovery of a 1-txn block influences the discovery time of subsequent blocks because the network is often described as finding blocks "every 10 minutes". In fact, mining is simply a independent probabilistic operation, similar to rolling dice multiple times.

So with validation rates of 17 seconds/MB, the maximum theoretical network bandwidth (propagation impedence "z" in [1]) is therefore naturally limited to 1/17 or ~60KB/sec of transactions (if a single infinitely sized block was submitted and so all available time was spent validating it). Actual bandwidth will be significantly lower and varies with maximum block size as shown next.

Given a block size Q, a propagation impedence z, and an average block discovery time T, the time to produce a "useful" (not a 1-txn) block Tu is going to be the block validation time plus the average time to discover a block or:

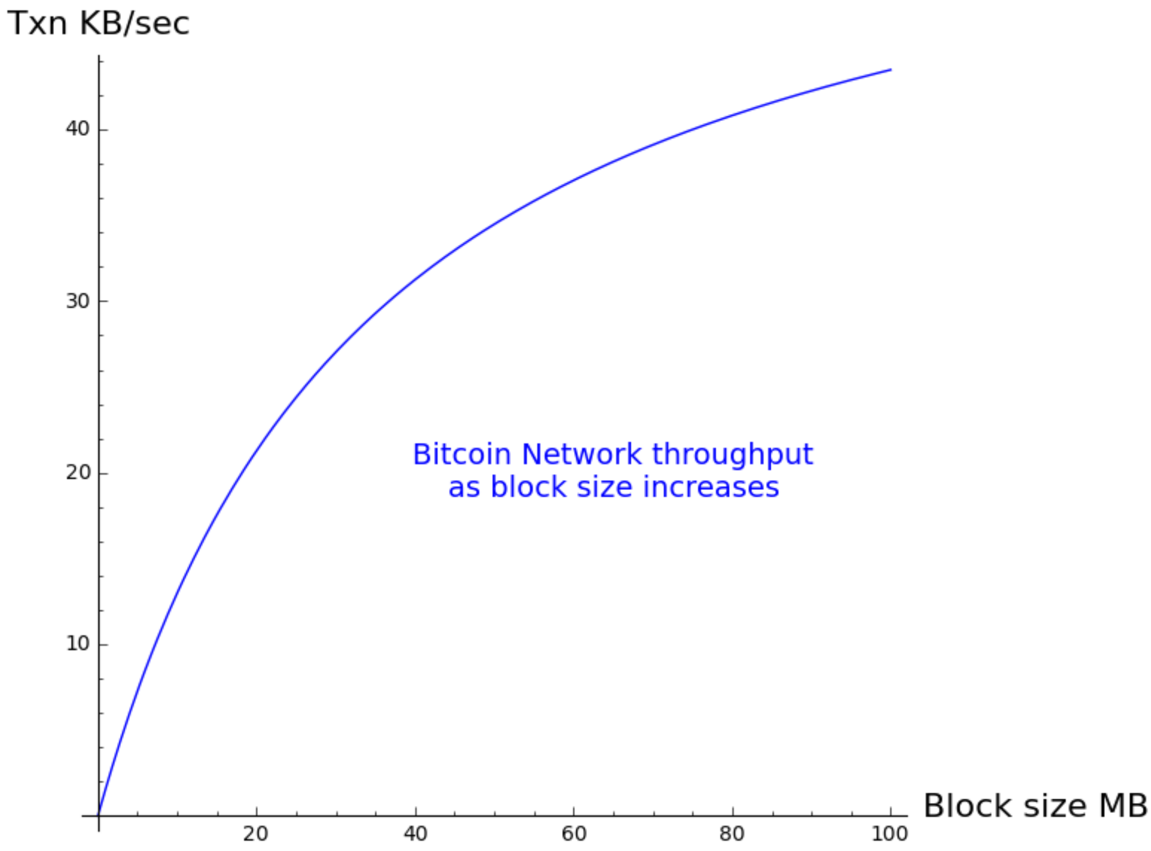
$$\mathrm{Tu} = Q/z + T \qquad \textbf{(4)}$$

So the maximum network throughput (Th) is the block size (Q) divided by the useful block discovery time (Q/Tu):

$$\mathrm{Th} = \frac{Q}{Q/z + T} \qquad \textbf{(5)}$$

With today's rate of z=1/17 MB/sec and T = 10 minutes the following plot can be drawn:

Figure 4: Bitcoin network Throughput as block size increases



For example, 32MB blocks would require approximately 10 minutes of validation (1-txn block generation) time. Since it subsequently takes (on average) 10 minutes to find a new block, we end up with a network with a 50% duty cycle (i.e. half the blocks are 1-txn half are carrying useful data), and ~30KB/sec of transaction throughput.

Therefore the average block validation time naturally limits the maximum rate that transactions can be committed to the block chain. Although the rate of transaction submission to the network is not directly related to the commitment rate, it is related for honest participants (participants will leave the network if their transactions are not being committed). Qualitatively, it becomes increasingly important to optimize block validation and creation times as blocks grow large, but is unnecessary today.

Effect on Block Discovery

Let us imagine the sequence of events that a miner/mining pool undergoes during network operation, noting event times:

T0: Block found by someone on the network
T1: Block header received
T2: Full block received
T3: Full block validated
T4: New block created
T5: ASICs start mining the new block
T6: A block that builds upon the T0 block is found by someone on the network. Note that T6 can actually occur at any point after T0.

Let us call the "full block mining interval" (T5,T6) even though the blocks mined may not actually be full (depending on the number of uncommitted transactions and the mining pool's preference). The time interval that "full" blocks cannot be created is

(T0,T5) since mining pools will not risk including already committed transactions into the new block. Let us call this interval the "validation time", because it constitutes the time required to receive and validate a new block. 1-txn blocks cannot be created during the time interval (T0,T1) because it is necessary to receive the prior block's header before creating them. So if 1-txn blocks are created, the orphan interval becomes (T0,T1) (the "propagation time"), and the 1-txn interval is (T1,T5).

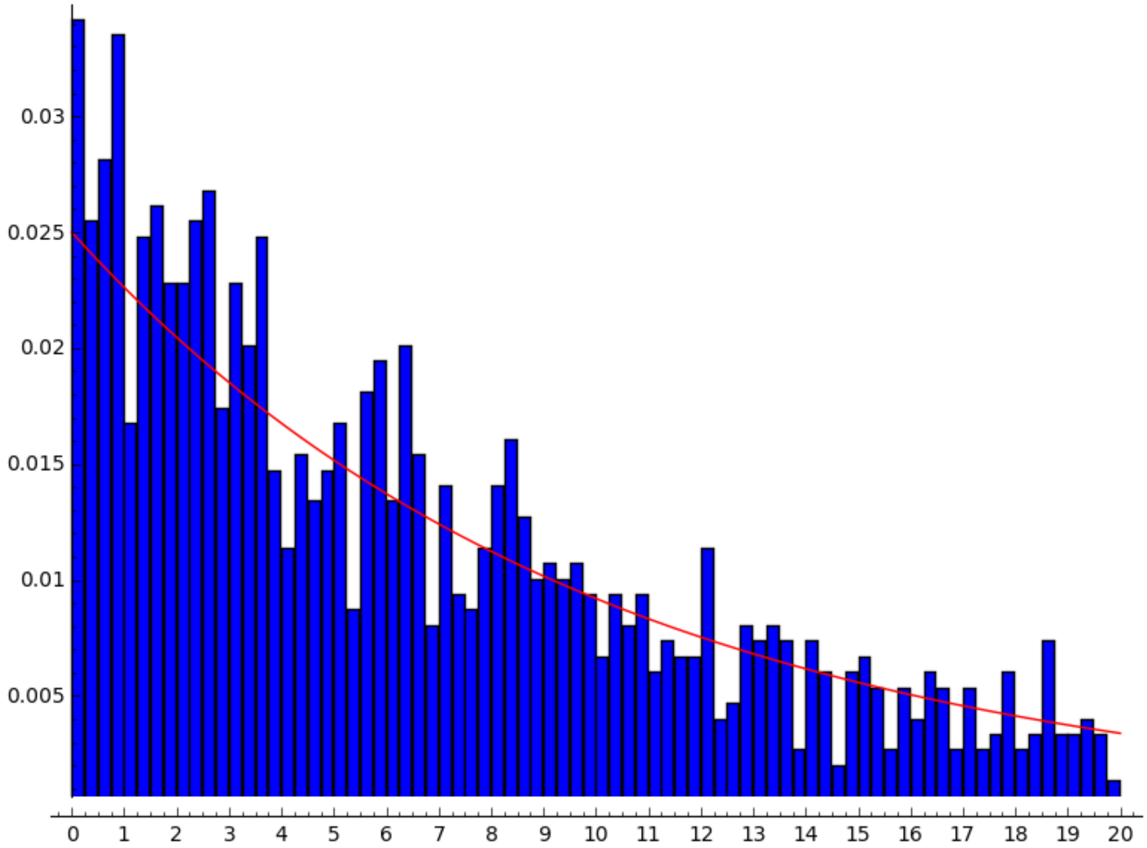
Let us propose that the 1-txn blocks observed in Figures 2 and 3 are caused by the inability to mine full blocks during the validation time.

During the propagation time (T0,T1) the network is effectively operating at significantly reduced capacity - the only effective miner is the block discoverer, since sibling blocks will get dropped from the blockchain data set. If this time is significant, it should be visible in a graph of the interval of time between block discovery as a lower-than-expected number of blocks with the shortest intervals. As described in chapter 2, The expected number of blocks found at a particular time can be modelled with the Poisson process equation (1).

Due the the inaccuracy of the self-reported block discovery times in the blockchain, another method was needed to examine block discovery intervals. The Bitcoin Relay Network[4] is a fast, centralized block solution distribution network. A Relay Network client was modified to log the time blocks were received. Given the architecture of the Relay Network, it is likely that the client received the blocks with sub-second latency and with a consistent error (that is, the time interval from when a block is discovered to when the client receives it is approximately equal for every block of similar size due to the architecture of the Relay Network)[5].

The following graph overlays both the theoretical and actual block discovery intervals, using data from the Relay Network for a period of N days from Dec 6,2015 to Dec 15,2015:

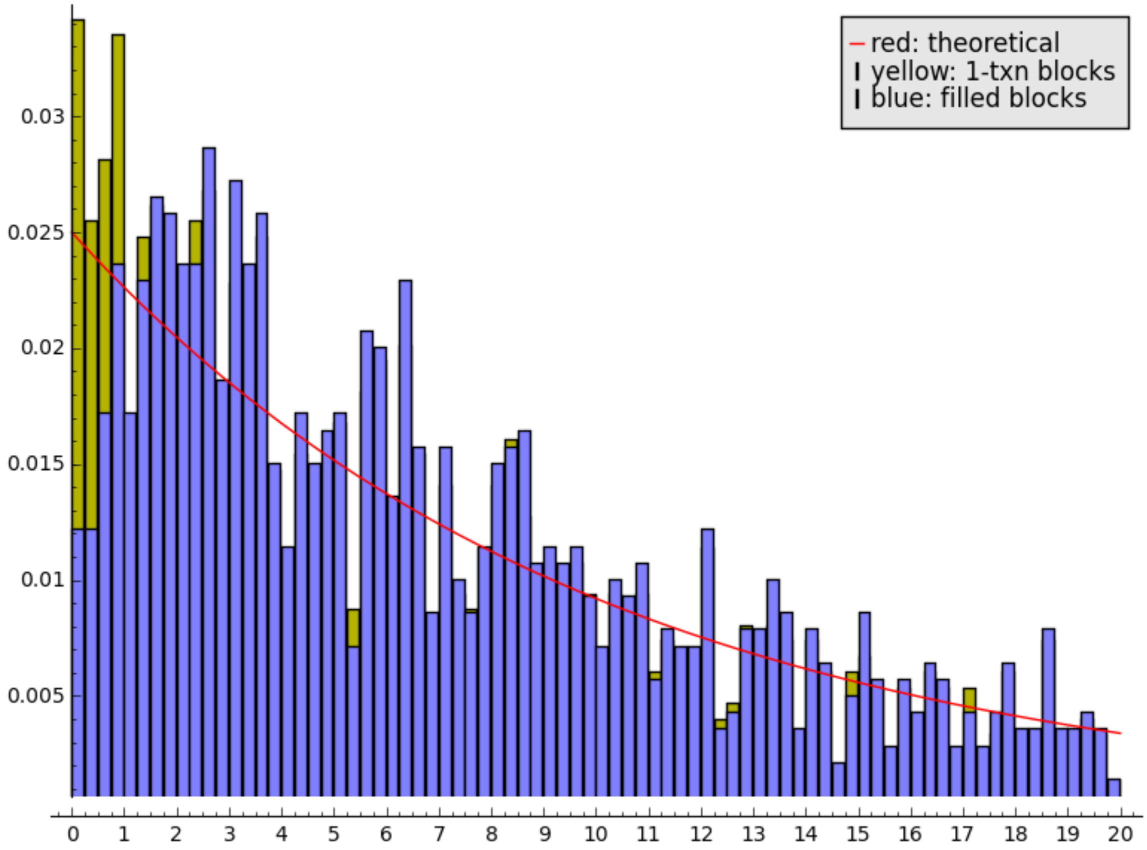
Figure 5: Block Discovery Intervals



Visually, there is no evidence of an unexpected decrease in block discovery due to block propagation time.

We can simulate a network where miners do not generate 1-txn blocks by ignoring intervals where the second block was a 1-txn block. In the next chart, light blue bars show the block discovery likelihood if no blocks were being generated during the validation time interval (T0,T5), and yellow shows the contribution of the 1-txn blocks.

Figure 6: Block Discovery Intervals with 1-txn blocks



This results in a graph whose first minute is significantly affected by block propagation and validation times. This result is consistent with the results of the other methods and our hypothesis that 1-txn blocks are created during the block validation interval.

Effect on Block Size

What would be the effect on the network if the 1 MB limit did not exist? Let us assume that there is an unlimited transaction pool so that every block can be completely filled. Every miner can mine its maximum block Mi. Furthermore, let us define the hash power of every miner producing blocks as a fraction of the total hash power Hi. So SUM(Hi) = 1. The network's average block production size B is therefore the sum of every mining pool's block size times the likelihood that that miner will find a block:

$$B = \sum_{\mathrm{i=pool}} \mathrm{Hi} * \mathrm{Mi} \qquad \textbf{(6)}$$

A mining pool can therefore reduce the overall network throughput by mining 1-txn blocks (Mi = 0 for one pool). This is in effect a hash power weighted vote for smaller blocks, but unlike ceasing to mine this vote does not carry significant financial penalties since the coinbase reward is much higher than the transaction fees. Given this model, a mining pool could also increase overall network throughput to any value D by mining blocks where (Mi = D/Hi), essentially by compensating for lack of hash power by mining a proportionally larger block. This means that a single mining pool can produce a block so large that the other pools never fully validate it and instead produce 1-txn block or orphans until the original pool is is able to produce another block so large that all other pools never fully validate, forever. This attack is easily defeated (see Appendix II), however, it shows that a pool can in theory spend the entire inter-block discovery time building/validating the next block and therefore drive the overall network's transaction processing capacity to this mining pool's maximum transaction processing capacity, regardless of his fraction of the hash rate. However, this is not the full story.

This transaction monopolization can only happen IF all the other mining pools choose to mine on top of that monopolizing mining pool's blocks. Yet, doing so will result in lower profitability for the other (the majority) of pools since they (having smaller validation capacity) must always mine 1-txn blocks and are therefore unable to reap transaction fees. This is an unstable situation - if a single mining pool chooses to ignore the large block and is able to find a small competing block while other pools are still validating a large block, it is in the other pools best interest to switch to this new sibling[6]. By switching, the other pools reduce the risk that they are mining on top of an invalid block, and can mine blocks with transactions. But if mining pools know that the majority will switch to a discovered sibling, it is rational for all pools except for the producer of the large block to search for a sibling rather than produce a 1-txn block.

In practice, other mining pools may not act in their own rational best interest for a variety of reasons beyond the capability of this game-theory based analysis. For example they may not have upgraded to software capable of doing so. However, mining pools can easily signal their intention and ability to switch via a tag in the coinbase transaction. This allows mining pools to communicate their intention and ability to competitively mine large blocks and also to only activate this strategy when a majority of the hashing power has also signaled its ability to do so.

How small does the competing block need to be? It should be shorter in length than what remains to validate on the large block for each validating miner. If this is the case, then the validating miner will be able to finish validating the small block before the large block and therefore switch to mining its own fee-paying block faster.[7]

So a miner's optimal strategy is to produce a block that the majority of the miners will switch to. Let us presume that the miner produces a block of size Bc to compete with the large block of size Bl. Let us define V sec/MB as the average miner validation rate. Therefore:

$$\begin{gathered} V * \mathrm{Bl} = \text{Average time to validate original block} \\ V * \mathrm{Bc} = \text{Average time to validate sibling (competing) block} \end{gathered} \tag{7}$$

So if t + V*Bc < V*Bl, mine the competitive block Bc. Otherwise mine on top of Bl. From the perspective of the large block mining pool with hashing fraction (1-h) (h is every other pool's fraction of the total hash power), his revenue (P) equation is:

$$P = (\text{reward} + \text{block size} * \text{fee per mb}) * (1 - \text{probability of competing block four} \tag{8}$$

In this situation, the "probability of a competing block being found" is not simply "h" (the static ratio of different mining pools' hashpower) because mining pools switch from mining Bc to mining Bl as validation completes. Instead we must sum the probability at every time instant t. Inserting this into our miner profitability equation and substitution symbols for the descriptions gets us:

$$P = (R + \mathrm{Bl} * \mathrm{Fm}) * \left(1 - \int_{t=0}^{\infty} (\text{prob\_finding\_block(t)} * \text{hash\_power(t)})\right) \tag{9}$$

As explained in equation 1, block discovery is a Poisson process, so the probability of finding any block at time t is:

$$\text{prob\_finding\_block(t)} = \lambda e^{-\lambda x} \tag{10}$$

where λ is 1/expected block discovery time = 1/10minutes = 1/600seconds

At first, the hash power is limited to the pool that produced the block, because everybody else is mining a competitive block. Eventually, the time to finish validating the large block is less then the time to validate a newly discovered sibling so everybody switches to the large block:

$$\text{hash\_power(t)} = \begin{cases} h \text{ if } t < (V*\text{Bl} - V*\text{Bc}) \\ 0 \text{ if } t >= (V*\text{Bl} - V*\text{Bc}) \end{cases} \tag{11}$$

Plugging these definitions into the revenue equation (9) yields:

$$P = (R + \text{Bl}*\text{Fm})*\left(1 - \int_{t=0}^{\infty}\left(\lambda e^{-\lambda x} * \begin{cases} h \text{ if } t < (V*\text{Bl} - V*\text{Bc}) \\ 0 \text{ if } t >= (V*\text{Bl} - V*\text{Bc}) \end{cases}\right)\right) \tag{12}$$

The integral of 0 (when in the large t case) is 0. This makes sense; we can drop the "long tail" of the Poisson because at that point everyone is mining the large block, yielding:

$$P = (R + \text{Bl}*\text{Fm})*\left(1 - \int_{t=0}^{(V*\text{Bl} - V*\text{Bc})}\left(\lambda e^{-\lambda x} * h\right)\right) \tag{13}$$

For simplicity, let us assume that the size of the competitive block Bc is approximately 0 (i.e. its a 1-txn block).

$$P = (R + \text{Bl}*\text{Fm})*\left(1 - \int_{t=0}^{V*\text{Bl}}\left(\lambda e^{-\lambda x} * h\right)\right) \tag{14}$$

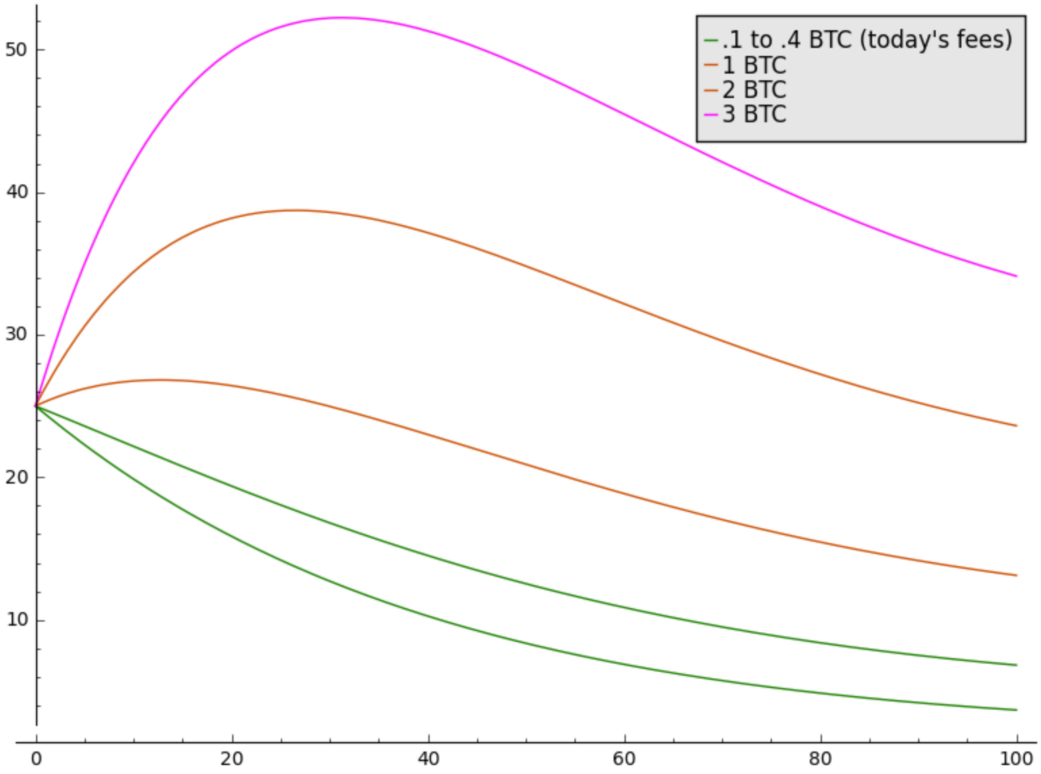
Note that this equation assumes that if the large block mining pool finds another block then all other pools switch to mining that block. This is a conservative and overly simplistic assumption. There is no reason for the competitive-block mining pools to ever switch to mining the large block so long as other pools have not switched -- and it is not in the interest of any mining pools to do so until the large block is validated. So mining pools have no reason to ever switch so long as they have accurate information about the intentions of the hashing power majority.

This assumption causes a rising tail in the equation (beyond the end of the included figures but addressed here in case you generate your own graphs) where the fees for gigantic blocks exceed all revenue lost by orphaned blocks which is unlikely to occur in practice.

Another unrealistic but conservative assumption is that the model assumes an infinite quantity of fee paying transactions. To be more accurate, the constant fee model in equation 8 (Bl*Fm) should be replaced by a function fee(b) that models the sum of all fees in a block of size b. Since rational miners will include more valuable fees first, the actual fee/MB must decrease as block sizes increase. Although it would be interesting to gather fee data and accurately model this function, the effect can only reduce block sizes and has been theoretically investigated in [1] so will not be addressed here.

The following chart graphs this equation given transaction fees per MB within the typical ranges we see today (0.1 to 0.4 BTC) and for 1, 2, and 3 BTC fees.

Figure 7: Miner revenue in BTC as block size increases

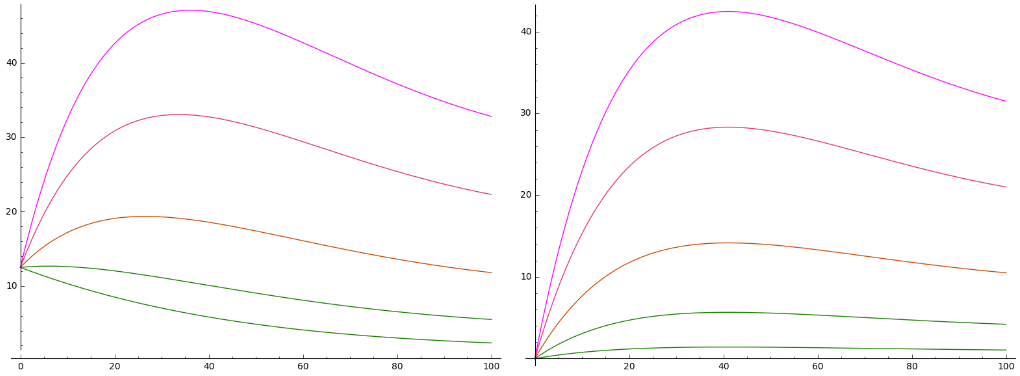


The maxima in these curves show that the network's average block validation rate will limit the size of a particular mining pool's blocks regardless of how much that pool's block validation capacity exceeds other pools. Therefore, an individual mining pool cannot increase average network block sizes to arbitrary size by "voting" with huge blocks. Given rational mining pools, attempting to do so will simply be an expensive lesson in how quickly a pool's blocks can be deliberately orphaned.

It is also interesting to note that as as transaction fees increase, the optimum block size grows, but in a non-linear manner. Proposals exist to artificially increase the maximum block size based ultimately on the fees users are willing to pay[5],[6],[7]. However, it is clear that the network contains a natural optimum capacity that maxes out at around 30MB irrespective of transaction fee size.

The situation is similar when the block reward halves to 12.5 BTC, and in the ultimate case 0 BTC.

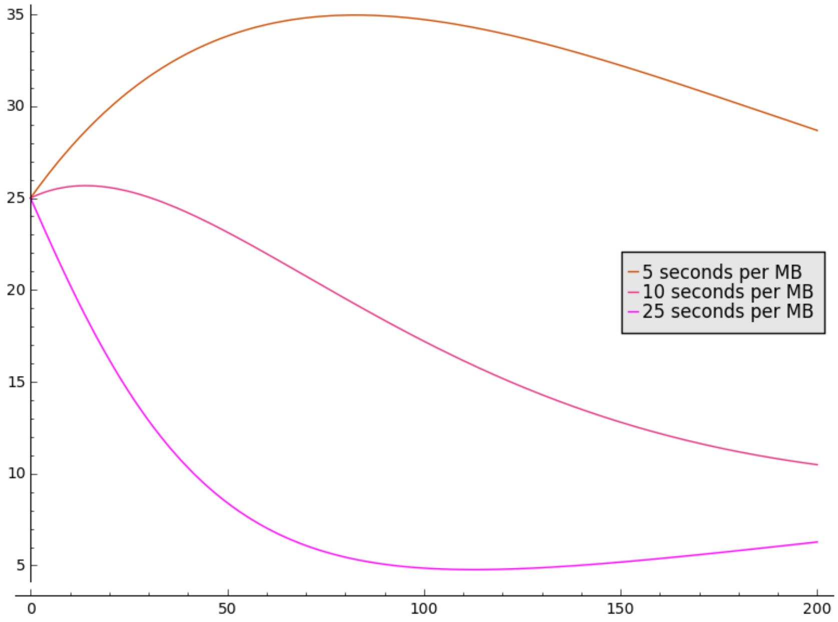
Figure 8 and 9: Miner revenue in BTC as block size increases, with block rewards of 12.5 and 0 BTC respectively



As shown by the green lines, there is actually a disincentive for miners to to include transactions in blocks, given today's 25 BTC reward and typical transaction fees (Figure 7). Yet as we move to 12.5 BTC reward (Figure 8), this incentive turns slightly positive at .4 BTC transaction fees/MB for block sizes up to about 5MB. While this small disincentive will serve to keep block sizes relatively small, it is unlikely to cause mining pools to consistently mine blocks without any transactions due to long term effects that are not captured in this model -- in particular, if transactions are not committed, Bitcoin ceases to be a value transfer network, and therefore the value of the Bitcoins mined would drop dramatically.

The final figure shows what will happen when the block validation times change. Since block validation includes the time to receive the full block and the time to check it for errors, this rate can be increased by more efficient software, faster CPUs and disks, and lower latency / higher bandwidth network connections. This chart was generated with a constant .5 BTC fee per MB and a 25 BTC coinbase reward. Note that the chart is denoted in seconds per megabyte so a smaller number is a faster validation time.

Figure 10: Miner revenue as validation speed changes



Effect on Network Security

In 2014 Yonatan Sompolinsky, Aviv Zohar showed how an increase in orphans reduces effective network-wide hash rate and how an increasing block size increases orphans[2]. They suggested the GHOST subtree weighting algorithm to resolve these issues. However, the market found another solution by mining 1-txn blocks on top of unverified blocks in order to optimize profitability. In this environment, increasing block size does not increase orphan rate since the quantity of data required to begin mining is constant and so propagates through the bitcoin network at the same rate independently of block size. Therefore, 1-txn blocks actually all network security to be maintained as the block size increases.

However, this solution is not as effective as GHOST. If an attacker could efficiently generate and inject invalid blocks into the Bitcoin Relay (fast-propagation) network, it could divert a significant amount of hash power onto an invalid fork (invalid blocks cannot be injected into the P2P network because they are validated before being forwarded). The attacker's fraction of the network's hash power increases by the fraction of miners it manages to divert onto the invalid fork. In this manner an attacker may temporarily gain the mining majority necessary to execute a double spend. However, it should be noted that over the short term diverting hash power will not allow the miner to discover blocks faster, it simply denies block discovery to other miners. This means that this attack cannot be used to mine more bitcoins, unless it is sustained for enough time to trigger a difficulty reduction (two weeks). But at these time scales it is easy to imagine that the network will respond by identifying and isolating the attacker, or creating block relay trust relationships between honest mining pools.

To defeat this attack, some entity should verify the hash and difficulty of a block before full block validation. This will force attackers to deliberately mine an invalid block which will make the attacker's cost significant (equal to the block reward and fees). Hash validation is extremely fast compared to transaction verification and could be done by each mining pool or as blocks enter the Bitcoin Relay network. This will limit time spent mining an invalid fork to the hash validation time.

4 Conclusion

The Bitcoin network is naturally limited by block validation and construction times. This puts an upper limit on the network bandwidth of 60KB/sec to transmit the block data to one other peer. Of course, the retransmission of transactions both in and outside of blocks doubles this rate (proposals exist to address this issue) and your node may be transmitting to multiple peers depending on its location within the peer to peer network and your node's capability and configuration. However, these multiples remain easily within the capabilities of corporate Internet connections and affordable hosting solutions, and is also within the capability of many residences.

The existence of 1-txn blocks allows a mining pool to slow down the average transaction throughput without significant financial impact. At the same time, if you believe that it benefits the network to increase or maintain current transaction commitment capacity, the unequal financial reward of a â€œsmallerâ€ vs â€œlargerâ€ block size "vote" (caused by transaction fees) is an incentive for mining pools to increase network capacity, so long as demand (fee-paying transactions) exist. As the network matures, the (coinbase) reward for merely being a participant declines and so mining pools must begin to accomplish valuable work (transaction validation and commitment) to profit.

If the maximum block size restriction was removed, rational mining pools would quickly adopt simple measures to discourage excessively large blocks found by other miners, since these blocks impact their own profitability. The size of a "excessively large block" emerges from maximizing the mining pools' revenue equation and depends on the network's average validation rate, the transaction fees available, and the block reward. Given today's transaction fees and today's 25 BTC coinbase reward, mining pools are actually disincentivized to include transactions. This will remain true given similar transaction fees even after the next coinbase reward halving. Therefore, it is unlikely that we will see significant block size increases until transaction fees increase or block transmission and validation times are dramatically improved, unless individual pool operators believe that doing so will increase the value of the Bitcoin token.

Proposals to insert explicit maximum block size voting (BIP-100) or an artificial variable maximum block size based on transaction-fees (flex-cap) are unnecessary. A mining pool's decision to create competitive siblings or 1-txn blocks rather than "full" blocks is a de-facto hash power weighted vote and simultaneous implementation of his choice to reduce or increase the average block size, based on his and the network's block validation capacity. Higher transaction fees naturally encourage mining pools to increase block size, despite the fact that these blocks have a higher chance of the block being orphaned in a manner similar to flex-cap schemes. At the same time, a single mining pool cannot "force the vote" by dramatically exceeding the rest of the network's transaction processing capacity without making it more profitable for the rest of the network to orphan this pool's blocks and effectively reject its vote.

5 Acknowledgements

We would like to thank Peter R and Zangelbert Bingledack (note these are pseudonyms on the www.bitco.in forums) for their careful review of this paper.

Notes

[1] This paper uses the term "1 transaction" or "1-txn" blocks to refer to blocks that predominantly contain only the coinbase transaction, but actually may contain a few more. For example, a miner may put private (unrelayed) transactions in a 1-txn block. We do not use the term "SPV mining" (a practice that also may produce 1-txn blocks) primarily because this term implies WHY these blocks are created, which is an assumption that an empirical analysis cannot determine. It seems that some miners who practice "SPV mining" never fully validate blocks, however the production of 1-txn blocks may occur for other reasons.

[2] This is just one hypothesis explaining the existence of these blocks, included here so the reader can understand how 1-txn blocks might reveal network function.

[3] Examination of the actual block verification algorithm could tell us, however this section of the paper focuses on empirical observations.

[4] Bitcoin Relay Network

[5] Bitcoin Relay Network Statistics

[6] From the perspective of a third miner, it is irrelevant when two sibling block solutions were found. To maximize profitability all that is relevant is when the miner can begin mining full-transaction blocks. Therefore the miner should stop verifying a large block and start verifying a small block if the work needed to verify the small block is smaller than the work left to complete the large block.

[7] This analysis makes the game-theoretic assumption that mining pools are rational and attempt to maximize their profitability. One caveat is of course that mining pools would need to be aware of this analysis and implement this behavior.

References

[1] Peter R: A Transaction Fee Market Exists Without a Block Size Limit (2015)

[2] Yonatan Sompolinsky, Aviv Zohar: Secure High-Rate Transaction Processing in Bitcoin (2014)

[3] Jeff Garzik: BIP-100 (2015)

[4] Christian Decker, Roger WattenHofer: Information Propagation in the Bitcoin network (2013)

[5] Andrew Stone: Flexible transaction space supply, based on fees(2015)

[6] Meni Rosenfeld: Elastic Block Cap(2015)

[7] BtcDrak: BIP-105 (2015)

[8] Peter R: Evidence of an effective blocksize limit: no protocol-enforced limit required (2015)

[9] TradeBlock: Bitcoin Network Capacity Analysis â€“ Part 6: Data Propagation (2015)

Appendix

I. A discussion of exceptional cases in the Ratio Method

The initial analysis yielded an outlier point in the first bin (not shown). Examination of the data show that this outlying point was caused by multiple 1 transaction blocks being mined after a single large block, for example:

timestamp	**size**	**# tx**	**hash**
1447931186	188	1	000000000000000008fa5a805bbbac0350c7c80361f2e1f01250d713ef133315
1447931006	188	1	00000000000000000276731fd3e4a20ee38f161480fd05b2b9c96405b3151454
1447930829	188	1	00000000000000000aad7ecd1699dedbf4158f20ccda65414f0eece4ab478802
1447930817	938872	2346	000000000000000008b267fe53f0ac3fe761ef1195121e068a393dae8996afa6

The analysis was modified to place sequences of 1 transaction blocks in the bin corresponding to the size of the first normal block. This did not have a significant effect on the findings but removed much of the visually obvious error, yielding the graph shown in Figure 2.

Another source of bin 1 error may be alternating sequences of 1 transaction and normal blocks, as shown here:

timestamp	**size**	**# tx**	**hash**
1447914783	266	1	00000000000000000c5f67748ec50fb445e85e96b1c7420ae56576d4374e3b00
1447914749	386140	801	0000000000000000047a9ffce1adb5abb75dd5ed675e4cdff4da7d7e195f7e3b
1447914373	209	1	00000000000000000ce7206d95bdd9737996fc12975a03eb8cbb68d47337c805
1447914397	784024	488	00000000000000000ce65df03a4d36fe3574cf0618709ddbe76e456cfc48b26a

In this case, is is conceivable that slower miner-validators may fall several blocks behind the blockchain tip. This would cause the 386KB block be more likely to be followed by a 1-txn block as compared to a situation where the 784KB block did not exist. In looking at the data above, please remember that the ~350 second gap between blocks 2 and 3 could be errors in miner clock settings. No attempt was made to correct this error source, whose effect would be to increase the apparent time to validate small blocks.

II. Never Ending Transaction Validation Attack Defeats

In this situation, a single miner controls all of the transactions committed to the blockchain which gives him a subset of the powers available to a miner with 51% of the hash power (such as censoring transactions). However:

1. Mining pools could modify their block creation logic to construct a block with new transactions without validating the incoming block. This could be done by comparing the transactions and spent TxOuts in the unvalidated block against candidate transactions for new blocks. Mining pools can then begin mining the new block while simultaneously validating the incoming one.
2. If miners could identify what transactions the â€œmonopoly minerâ€ is censoring, they can be safely added to a 1-txn block.