Block Size Increase

BitFury Group

Sep 06, 2015 (Version 1.1)

Abstract

Plans of block size increase are a subject of a heated debate in the Bitcoin community. The subject has gained increasing attention since the beginning of 2015, when the size of blocks started to approach the current hard limit of one megabyte. We study arguments for and against block size increase, and we analyze existing proposals by influential Bitcoin developers to increase the block size limit.

Version History

Version	Date	Change description
1.0	Aug 31, 2015	Initial version
1.1	Sep 06, 2015	Clarified statements concerning Bitcoin XT (Section 2.3); added a copyright notice

Blocks in the Bitcoin blockchain currently have a hardcoded upper size limit of 1 MB (more precisely, 1 million bytes). Since the beginning of 2015, there have been various proposals to increase this limit or abandon it altogether. The commonly cited reason behind such proposals is the rising throughput of Bitcoin transactions. Currently, Bitcoin processes up to 150,000 transactions per day, or less than 2 transactions per second (tps) [1]. During the major flood attack on the Bitcoin network that took place on July 2015 [2], the peak throughput was approximately 214,000 transactions per day, or 2.5 tps. The widespread estimate of maximum transaction throughput for the Bitcoin network is 7 tps. This number is dwarfed by more than several thousand tps peak throughput of Visa and other major payment processing systems. If Bitcoin positions itself as a replacement for other payment services, it must handle increased throughput. Thus, there are two interconnected problems concerning block size:

- What would be the impact of the block size limit increase have if the transaction throughput remains roughly the same?
- What would be the consequences of the rising number of transactions in the Bitcoin ecosystem?

We study both of these questions in the following sections.

1 Block Size Mathematics

There is currently a hard limit of block size of 1 MB; any block larger than 1 MB is considered invalid by all major Bitcoin clients [3]. This limit corresponds to the maximum of roughly 4,000 transactions per block (assuming the average transaction size is around 200–250 bytes). As blocks are mined every 10 minutes on average, this gives us the maximum throughput of approximately 7 transactions per second (tps). Given practical considerations of elongated transaction queue times, some studies estimate the throughput limit is realistically 3.5 tps [4]. Thus, the block size increase is necessary for expanding Bitcoin's transaction processing maximum throughput, especially as it could become as an alternative to existing payment systems which can handle thousands of transactions per second.

Mathematical modeling [5] suggests that the increasing strain on the Bitcoin network negatively influences the transaction confirmation time. The influence becomes more pronounced as the transaction throughput approaches the maximum. For example, at 2.8 transactions per second (80% of the maximum), the median time to the first transaction confirmation is 18.5 minutes; half of transactions is confirmed slower. To provide a comparison, if the throughput is 1 tps, the median time to the first confirmation is 7 min.

Increasing the block size could alleviate the problem, as throughput is inversely proportional to the size of a block, and thus confirmation delay decreases with growing block size (Table 1). As seen from the table, a 2 MB block size would be enough to solve the problems with confirmation delay; the further increase would not give a significant boost to the confirmation speed based on the current network limits. On the other hand, it is difficult to predict how the size increase will influence transaction throughput of the Bitcoin network. A good approach therefore should be able to adapt depending on

Table 1: Processing time of transactions depending on block size and network load

Throughput, tps	Block size, MB	Network load, %	Median processing time, min	Processing of 90% tx, min
1.75	1	50.0	8.5	29.0
(normal)	2	25.0	7.0	23.5
	3	16.7	7.0	23.0
	4	12.5	7.0	23.0
	8	6.3	7.0	23.0
	20	2.5	7.0	23.0
2.5	1	70.0	13.2	42.8
(peak)	2	35.0	7.4	24.7
	3	23.3	7.0	23.0
	4	17.5	7.0	23.0
	8	8.8	7.0	23.0
	20	3.5	7.0	23.0
3.5	1	100.0	129.1	380.0
(maximum)	2	50.0	8.5	29.0
	3	33.3	7.4	24.7
	4	25.0	7.0	23.5
	8	12.5	7.0	23.0
	20	5.0	7.0	23.0

Bitcoin network load. We also note the limitations of modeling altogether, as empirical data shows that even during flooding attack in July 2015, the actual median confirmation time didn't rise above 12 minutes [6].

There is another aspect to the throughput problem: one could argue that currently most of transactions with slow confirmation times are so-called *dust* – transactions with very low output values and lower than average transaction fees [7]; dust is commonly used in denial of service (DoS) attacks against Bitcoin, the largest of which took place in July 2015 [2]. Some types of dust transactions with extremely low fees are not relayed and not included in blocks. Imposing further restrictions on transactions acts as an alternative to a block size increase. Compared to a block size increase, transaction restrictions have an advantage: they constitute a soft fork (i.e., blocks mined by upgraded software with more restrictions would be accepted by non-upgraded software), whereas a block size increase implies a hard fork (i.e. change to the Bitcoin protocol that is not backward compatible). On the other hand, this solution to transaction throughput could cause a backlash among Bitcoin users and developers building applications on top of the blockchain.

There is an opinion [8] that keeping the block size limit relatively low would help develop a market mechanism for transaction fees where users pay higher fees to confirm a transaction faster. Higher fees could also encourage developing applications pegged to the Bitcoin blockchain but not using it directly, such as sidechains [9] and micropayment channels [10]. This solution requires upgrading some

Table 2: Resource consumption by full nodes as the block size increases

Characteristic	Scale	Block size, MB ($= N/2$)						
	factor	0.5	1	2	4	8	16	32
Transaction throughput, tps	N	1.75	3.50	7.00	14.0	28.0	56.0	112
Number of txs in a block	N	1050	2100	4200	8400	16800	33600	67200
Blockchain storage per day, MB	N	72	144	288	576	1152	2304	4608
Blockchain storage per year, GB	N	26	51	103	205	411	821	1643
Transaction processing time, ms	1	0.33	0.33	0.33	0.33	0.33	0.33	0.33
Block verification time, s	$N + 0.09N \log_2 N$	0.07	0.15	0.33	0.71	1.51	3.23	6.86
Average bandwidth, kB/s	N	74	148	296	592	1184	2368	4736
Daily traffic, GB	N	6.2	12.4	24.8	49.6	99.2	198	397
Yearly traffic, TB	N	2.2	4.4	8.8	17.7	35.4	70.7	141
RAM usage, GB[1]	N	2	4	8	16	32	64	128
Immediately excluded nodes, %	n/a	0	20	40	75	90	95	95
Excluded nodes in 6 months, %	n/a	5	25	50	80	95	95	95

Bitcoin software, e.g., Bitcoin wallet applications that don't have variable transaction fees. However, high transaction fees could reduce the incentive to use Bitcoin for micropayments, remittances, and international payments.

Increasing the block size could lead to additional strain on the nodes of the Bitcoin network. The most computationally expensive operation during transaction relay is verification of the ECDSA signatures of the transaction inputs (each transaction is verified before it is relayed to other nodes in order to protect the network from DoS attacks). Modern CPUs are able to verify several thousand transactions per second [11]. Thus, current transaction throughput of several tps is not much of a burden for the network.

More concerning is that rising transaction throughput results in increase in memory consumption (both RAM and disk space) and Internet traffic in the network nodes. Please see Table 2 for characteristics of full nodes and our estimates of the impact of increased block size; we use *average* values of block size, which currently is ≈0.5 MB.

Current characteristics in Table 2 are obtained as follows:

- **Transaction throughput** is block size divided by the expected time interval between blocks (600 seconds) and by the current average transaction size (slightly less than 0.5 kilobytes). Transaction throughput can also be calculated using daily transaction statistics [1]. Both methods yield a result of approximately 1.75 tps.
- **Number of transactions in a block** is the block size divided by the average transaction size; alternatively, it is equal to transaction throughput times 600.

$$\langle \text{Transactions in a block} \rangle = 1.75 \cdot 600 = 1050.$$

[1] Approximate value to keep the same transaction processing time. Systems with less RAM are viable but would process transactions with delays

- **Blockchain storage** is a mean block size multiplied by the expected number of blocks (144 blocks per day; $144 \cdot 365 = 52560$ blocks per year).

$$\langle \text{Storage / day} \rangle = 0.5 \cdot 144 = 72 \text{ MB};$$

$$\langle \text{Storage / year} \rangle = 0.5 \cdot 144 \cdot 365 = 26280 \text{ MB}.$$

- **Transaction processing time** is based on an assumption that a node can process 3000 transactions per second [11]. Note that each transaction needs to be parsed, then the node needs to lookup the unspent transaction outputs corresponding to transaction inputs and check each of signatures embedded in inputs.
- **Block verification time** is transaction processing time multiplied by the mean number of transactions in a block times 0.2. The last factor corresponds to the notion that most transactions in a block are already verified when the block arrives; node only needs to calculate transaction hashes and look them up in the transaction cache.

$$\langle \text{Block verification time} \rangle = 0.2 \cdot 1050 \cdot 0.0003333 = 0.07 \text{ sec}.$$

- **Average bandwidth** of a node is 74 kilobytes per second [13].
- Currently, a node processes more than 6 gigabytes (GB) of **traffic per day** [13].
- **RAM usage** is the most difficult variable to estimate as it depends on many factors. Furthermore, RAM usage differs for various types of nodes. One of the major components contributing to RAM consumption is the unspent transaction output set, which currently takes more than 4 GB [14]. The set does not have to reside completely in RAM, although other storage methods lead to elevated transaction verification time. We use an empirical value of 2 GB based on recommendations [15] which is lower than what other measurements suggest [16]. Obviously, for specialized nodes such as mining nodes, the requirements on RAM are higher, as these nodes need to store the entire unspent transaction output set in RAM in order to rapidly verify incoming transactions and blocks.

We calculate key node parameter estimates for increased block sizes by multiplying our baseline values by scaling factors in the second column of the table, with N denoting the multiplier for block size. These scaling factors are established as follows:

- We assume that the average size of a transaction remains the same, so **transaction throughput** and the **number of transactions per block** scale linearly with block size. Similarly, it is obvious that **blockchain storage** linearly depends on the block size as well.
- **Transaction processing** requires searching in the unspent transaction output set. With an optimal implementation, average search time depends logarithmically on a size of the set; the latter should scale linearly (as it is implied by historic data [17]). As the number of unspent transaction outputs is currently well over 10^7, transaction processing time would remain nearly the same as the block size increases.

The above is true only if we define transaction processing as a local process that does not depend on the state of incoming and outgoing peer connections. Network-related operations such as relaying transactions are likely to lag for big blocks as they require proportionally more bandwidth. The dependency between the elapsed time of network operations and the block size can be derived using extensive modeling and is beyond the scope of this research.

- **Block verification time** exhibits the fastest growth rate as the block size increases; while most transactions in a block should already be verified when it arrives, a node still has to compute hashes for all transactions and look them up in the cache. As in the previous case, lookup times scale logarithmically depending on the size of the cache.

  Let S denote the current size of the transaction cache and t denote average time to look up a single transaction. If the block size increases N times, time to look up each transaction increases to $t\log_2(NS)$; thus, the block verification time would increase by a factor of

  $$N\frac{t\log_2(NS)}{t\log_2 S} = N(1+\log_2 N/\log_2 S).$$

  Currently, $S \approx 2000$ [18], which implies the scaling factor $\approx N(1+0.091\log_2 N)$.

- **Traffic** depends on the size of blocks passing through the node, as well as on the transaction throughput. Both of these values scale linearly.

- We consider that the nodes are likely to store the same percentage of the unspent transaction output set in RAM as they do now; as the set grows linearly with block size, **RAM usage** should grow linearly, too. Other factors contributing to RAM consumption (such as the size of the transaction cache) exhibit linear or sub-linear growth, so they are unlikely to overwhelm the unspent transaction output set in the long run. Unlike other factors, RAM usage is determined mainly by user settings; a node may run relatively well with lesser RAM, however, it would result in delays in transaction processing.

The table contains an estimate of how many full nodes would no longer function without hardware upgrades as average block size is increased. These estimates are based on the assumption that many users run full nodes on consumer-grade hardware, whether on personal computers or in the cloud. Characteristics of node hardware are based on a survey performed by Steam [19]; we assume PC gamers and Bitcoin enthusiasts have a similar amount of resources dedicated to their hardware. The exception is RAM: we assume that a typical computer supporting a node has no less than 3 GB RAM as a node requires at least 2 GB RAM to run with margin [15]. For example, if block size increases to 2 MB, a node would need to dedicate 8 GB RAM to the Bitcoin client, while more than a half of PCs in the survey have less RAM.

We also include an estimate of how many existing nodes would be excluded from the network in the next 6 months. While the immediate drop of the number of nodes is mainly related to RAM and CPU usage, in the long run there are other limiting factors such as disk space and Internet traffic. For example, with an 8 MB average block size, a node would require capacity for more than 34 GB of disk space to process approximately 3 terabytes (TB) of traffic each month.

Both estimates of a number of excluded nodes should not be taken for granted, as there is a multitude of other factors influencing the capability of Bitcoin nodes that could not be assessed without a large-scale modeling.

Block verification time implied by Table 2 has negative implications for large block sizes. Currently, relaying a block to all network nodes typically takes 10–15 seconds and depends chiefly on users' bandwidth. With a significant increase of block size, some owners of nodes may consider the purchasing of additional hardware to not be worth the expense. This could lead to elevated block relay times and increased orphan rates.

The increased requirements on hardware could also lead to more network centralization caused by nodes being excluded from the network. The drop in full node numbers is currently amplified by the transition to lightweight SPV (simplified payment verification) clients. For users who absolutely need full nodes, e.g. miners and Bitcoin exchanges, satisfying hardware requirements is less of a burden than for average users. On the other hand, hardware requirements for specialized nodes are several times higher than for regular nodes (e.g., a typical mining node currently consumes more than 8 GB RAM), so the problem of increased block size pertains not only to average nodes.

1.1 Block Size Increase Pros and Cons

In favor of an increased block size:

- More transactions per second, faster confirmation time. Note that block confirmation time also depends on mean block generation time (set to 10 minutes by the Bitcoin protocol). Thus, median confirmation time can never drop below several minutes unless the block generation aspect of the protocol is changed as well.
- More transactions for systems built on top of Bitcoin blockchain, such as colored coins platforms (e.g., Counterparty, OmniLayer).
- Increasing block size would allow keeping low transaction fees.

Against a block size increase:

- Increasing the block size would require a hard fork of the system, meaning newly solved blocks won't be recognized by non-upgraded software. This could lead to negative consequences for Bitcoin pricing and reputation.
- Larger blocks may propagate slower through the network, which could lead to an elevated orphan rate and increased probability of a successful double-spend. Because of poor bandwidth between China and the rest of the world, European and American miners are at a disadvantage compared to Chinese mining pools as long as the latter hold the majority of the mining power.
- Processing larger blocks requires more powerful hardware, which may cause a decrease in the number of full nodes in the Bitcoin network. This could lead to centralization of the network and a change of local and global peer-to-peer topology.

- Elevated hardware requirements and growing orphan rate could lead to unstable block generation and inaccurate evaluation of the difficulty target for the network.
- The Bitcoin blockchain was not originally designed for all kinds of transactions; some transactions are likely best handled with other technologies (such as sidechains and micropayment channels).
- Higher fees resulting from keeping the block size limit low may help develop a transaction fees market. This would help the Bitcoin ecosystem and encourage development of off-chain solutions.
- The Bitcoin blockchain should not be used for cheap permanent storage, as it is viable with big blocks and low transaction fees.

2 Proposals

2.1 Gavin Andresen's First Proposal

Historically, the first proposal to increase the block size limit was a momentary elevation of the limit up to 20 MB proposed by Bitcoin Core[2] developer Gavin Andresen. This proposal has serious drawbacks:

- The hard fork caused by the increase won't cause trouble only if there is a consensus among all major members of the Bitcoin ecosystem; otherwise, it would lead to the fork of the Bitcoin blockchain with unpredictable negative consequences for Bitcoin price and user trust.
- The chosen size of 20 MB is hardly substantiated from the theoretical standpoint, whereas such a fundamental decision must be accompanied by thorough modeling as well as mathematical and economic analysis.

2.2 BIP 101

On June 22, 2015, Gavin Andresen submitted a similar proposal as BIP 101 [20]. According to the proposal, the block size limit should increase to 8 MB in 2016 and then double every two years until reaching 8,192 MB in 2036; after this, the block size limit would remain constant. The deployment date of the increase would be decided by a supermajority rule: two weeks after 750 of 1,000 consecutive blocks in the blockchain have a specified version number, but not before January 11, 2016. The rationale behind the proposal is that CPU power, storage capacity, bandwidth all grow exponentially according to Moore's law [21]; this growth, however, would eventually slow down.

This proposal came after Andresen received negative feedback on 20 MB blocks from the Chinese mining pools and countered with 8 MB. The main negative points about BIP 101 are the same as in the prior proposal. While the block size limit in BIP 101 grows at a predictable rate, it is hard to estimate whether this rate would correspond to the growth of the Bitcoin network in the future.

[2] Bitcoin Core is a de-facto reference implementation of the Bitcoin protocol that runs on most full nodes

2.3 Bitcoin XT Fork

Not having gained support from the Bitcoin Core developers, Andresen decided to implement BIP 101 and some other patches that were not included into Bitcoin Core in a separate Bitcoin client Bitcoin XT, co-developed with Mike Hearn.

Negative points about Bitcoin XT:

- Split between client software installed on full nodes could cause a negative impact on Bitcoin's image and decrease trust in the Bitcoin blockchain.
- Transition to Bitcoin XT could cause a fork of the blockchain: there is a concern that small differences between Bitcoin XT and Bitcoin Core could result in Bitcoin XT accepting some transactions that are considered invalid in Bitcoin Core.
- Trustworthiness of Bitcoin XT is lower, as code reviews and formal verification are utilized to a lesser degree than in case of Bitcoin Core.
- Unlike Bitcoin Core, Bitcoin XT developers don't plan to use a consensus mechanism to approve protocol changes that necessitate a hard fork.
- The developer team of Bitcoin XT is significantly smaller than the team working on Bitcoin Core. It raises concerns as to whether they are able to objectively assess changes being adopted into Bitcoin XT.

One could argue that the development process used in Bitcoin XT would allow for faster changes than Bitcoin Core; however, in our view, this advantage does not outweigh the drawbacks.

2.4 BIP 100

One of the more moderate approaches to the block size increase was proposed by Bitcoin Core developer Jeff Garzik in BIP 100 [22]. According to the proposal, the hard limit on the block size is lifted; instead, a floating limit is introduced. The fork is scheduled for January 11, 2016. The changes to the floating limit are decided by miner voting (similarly to BIP 34 [23]):

- Block size shall not be over the historical limit of 32 MB imposed by the peer-to-peer protocol utilized by Bitcoin.
- To introduce the floating limit, 90% of 12,000 consecutive blocks (which corresponds to 3 months) should indicate support of the proposal.
- In order to change the floating limit, miners would need to include a proposed block size limit into the coinbase signatures of mined blocks. Votes are evaluated by dropping the top 20% and bottom 20%, and then selecting the most common floor (minimum) of the remaining values.
- The increase or decrease of the block size limit may not exceed 2x in one voting round.

Compared to other variants, this proposal has the following advantages:

- Block size limit is determined by consensus
- Block size limit isn't regulated in a one-time decision but rather with a constant mechanism that allows miners to adapt to changes in transaction throughput
- The proposal solves the problem of a potential blockchain fork, as the new block is automatically supported by the majority of miners
- The proposal does not require transitioning to new software.

While miners do not represent the entire Bitcoin community, it is in miners' best interest to maintain the healthy overall state of the network.

The disadvantage of BIP 100 is the vague wording used in the description of vote counting. If we assume that the floating block size limit is decided by getting the minimum of votes remaining after trimming top and bottom 20%, a party with 21% of hashing power could effectively delay all attempts to rise the limit indefinitely. Another drawback (common with other proposals) is the remaining upper limit of 32 MB which is caused by the restriction on message size for the peer-to-peer protocol utilized by Bitcoin. In order to lift this limit, the protocol must be amended to allow relaying blocks in multiple messages.

We address the problem of a 21% attack with a mathematical formalism for the voting process in BIP 100 (Appendix A). Our approach is aimed to clarify vagueness of the voting process; it can be used to algorithmically determine the best block size limit which would be considered appropriate by all miners.

2.5 BIP 102

As an alternative to BIP 100, Jeff Garzik proposed BIP 102 [24]. It is a minimalistic solution that increases the block size limit to 2 MB on November 11, 2015. The author positions BIP 102 as a fallback solution if consensus is otherwise not reached.

The immediate drawback of BIP 102 is the lack of further plans on block size regulation; at best, it could be used as an intermediate solution.

2.6 Wuille's Proposal

Pieter Wuille, another Bitcoin Core developer, drafted his own proposal to increase the block size [25]. Wuille's proposal amounts to changing the hard block size limit with an exponentially growing function that depends on timestamp of the block, starting from January 2017. If implemented, the maximum block size would increase approximately 17.7% per year.

It is possible that Wuille's growth rate proposal is too conservative. The proposal ties the block size limit growth to the growth rate of Internet bandwidth; however, if Bitcoin becomes a widespread payment system, transaction throughput would need to increase much faster.

2.7 Other Solutions to Bitcoin's Scalability

Block size limit increase is a near-to-intermediate term solution to the problem of Bitcoin's scalability (i.e., timely processing of transactions circulating in the network). There are other efforts within the community that are aimed to make Bitcoin more scalable in the long term:

- **Invertible bloom lookup tables** (IBLT) are aimed to propagate blocks much faster by minimizing the amount of data sent [26]. If implemented, miners would be able to increase block size dramatically without worrying about an elevation of the orphan rate.
- Off-chain solutions such as **sidechains** [9] and **micropayment channels** such as Lightning Network [10] can be used to decrease the number of transactions in the Bitcoin network. These solutions could be used to scale Bitcoin fast while maintaining a relatively small block size.
- **Treechains** [27] would allow organizing the blockchain in a tree, not as a linear sequence. This would help increase transaction throughput that the network could handle without increasing the block size limit.
- **Greedy heaviest observed sub-tree** (GHOST) protocol [28] can be utilized to reduce mean time between blocks from 10 minutes to several seconds without the associated risk of wasting a major part of the network's mining power on orphaned blocks.
- **Centralized off-chain ledgers** used by Coinbase [29], ChangeTip and other services. These solutions can help decrease transaction throughput; however, there are questions about their security and reliability.
- The problem of increased hardware requirements could be solved with enterprise-grade **supernodes** maintained by miners and other entities interested in stability of Bitcoin's ecosystem. By using specialized software and hardware, these nodes would be capable of processing thousands of transactions per second.

As our research studies block size regulation, the analysis of these scalability efforts is beyond its scope.

3 Conclusion

In order for the Bitcoin ecosystem to continue developing, the maximum block size needs to be increased. There is common understanding among Bitcoin developers that the current limit of one megabyte limits scalability of Bitcoin and could prevent its wider adoption. This opinion is further supported by mathematical modeling that shows long transaction confirmation delays as the block size approaches its limit.

Elevated hardware requirements could result in a substantial percentage of full nodes dropping from the network should the block size increase to 8 MB right now. This makes an argument against proposals to raise the maximum block size in a single big jump. Similarly, proposals implying a fork of client software could have negative consequences for the whole Bitcoin ecosystem. Jeff Garzik's BIP 100 and Pieter Wuille's proposal are sound in both these aspects.

The optimal way to solve the block size debate is by a consensus decision; the voting mechanism introduced in BIP 100 is a good way to implement such a consensus. Forecasts for the growth of the Bitcoin network made by other proposals don't have enough predictive power, and in this case, the cost of a mistake is high. As BIP 100 lets members of Bitcoin community decide the block size limit, it is the best solution in this regard.

Because of the stated reasons, we consider BIP 100 to be the most prudent choice to grow the block size limit in the near-to-intermediate term. At the same time, we recognize that block size regulation is only one of the steps in Bitcoin's evolution; other improvements must be made to the protocol in order to improve Bitcoin's scalability.

Appendix A Formalizing Voting in BIP 100

BIP 100 introduces a floating limit on block size, which is decided in voting rounds among miners. To vote, a miner must include a value of the limit he agrees with into the coinbase scripts. Voting rounds span 12,000 blocks (approximately 3 months). Allowed votes range between 0.5 and 2 times the current block size limit.

The weak point of BIP 100 is the mechanism how votes are tallied up. One can interpret it as trimming top 20% and bottom 20% and taking the minimum of the votes left [30]. In this case, a miner or a group of miners with slightly over 20% hash rate can hold back all attempts to increase block size limit (e.g., they can do so because they use outdated equipment that won't run with bigger block size).

We propose an enhancement to BIP 100 where the target block size limit is derived from the votes with relatively simple equations using optimization theory. Namely, we introduce a non-negative dissatisfaction function and minimize its value summed over all votes. The block size limit found this way would satisfy miners provided the dissatisfaction function is chosen appropriately. The model is not susceptible to 21% attack provided adequate parameters are chosen for it. Our enhancement should not be considered final; it needs to be verified by the Bitcoin community and modeled to highlight possible issues.

We can consider values v_i as a ratio of an actual size to the current block size limit. For votes defined this way, $0.5 \leqslant v_i \leqslant 2$.

Problem. We have block size limit votes $v_i > 0$, $i = 1, 2, \ldots, M$ read from blocks. (In BIP 100, $M = 12000$.) We need to determine the block size limit s that satisfies miner votes in a best way.

Solution. Consider a function of two variables $F : (0, +\infty)^2 \to [0, +\infty)$. The value $F(s, v)$ expresses dissatisfaction of a miner that voted for block size limit v, if the vote will result in limit s. Obviously, $F(v, v) = 0 \quad \forall v$. The problem is therefore reduced to an optimization problem

$$\hat{s} = \arg\min_{s} \sum_{i=1}^{M} F(s, v_i). \tag{1}$$

If we group votes for the same limit into blocks $v_{(1)} < v_{(2)} < \cdots < v_{(k)}$, with i-th block weight

$$w_i \in (0,1), \quad \sum_{i=1}^{k} w_i = 1,$$

then (1) is equivalent to

$$\hat{s} = \operatorname*{arg\,min}_s \sum_{i=1}^{k} w_i F(s, v_{(i)}) \overset{\text{def}}{=} \operatorname*{arg\,min}_s L(s), \tag{2}$$

where $L(s)$ is the target function we need to minimize.

A.1 Dissatisfaction Functions

What kind of a dissatisfaction function F should we consider? Consider two classes of functions that reduce F to a function of one argument:

- $F(s, v) = D(s - v)$, i.e., miner's dissatisfaction depends on an *absolute* difference between the target size and and his vote. In this case, target function

$$L(s) = \sum_{i=1}^{k} w_i D(s - v_{(i)}). \tag{3}$$

- $F(s, v) = D(s/v - 1)$, i.e., miner's dissatisfaction depends on a *relative* difference between the target size and and his vote. In this case,

$$L(s) = \sum_{i=1}^{k} w_i D(s/v_{(i)} - 1). \tag{4}$$

Function D should satisfy

$$D(0) = 0; \quad \forall x\ D(x) \geqslant 0; \quad \lim_{x \to \pm\infty} D(x) = +\infty; \tag{5}$$

Additionally, we will consider continuously differentiable functions D in order to find the minimum by solving $L'(s) = 0$; we don't lose much generality by considering this class of functions.

A.1.1 Quadratic Function

We can consider

$$D_{sq}(x) = \begin{cases} x^2, & \text{if } x \geqslant 0, \\ \alpha x^2, & \text{if } x < 0; \end{cases}$$

Logically, $\alpha < 1$ (a miner does not like situation where a target limit s is bigger than his vote v, but is partially satisfied with the situation where $s < v$). A couple of D_{sq} functions are plotted on Figure 1.

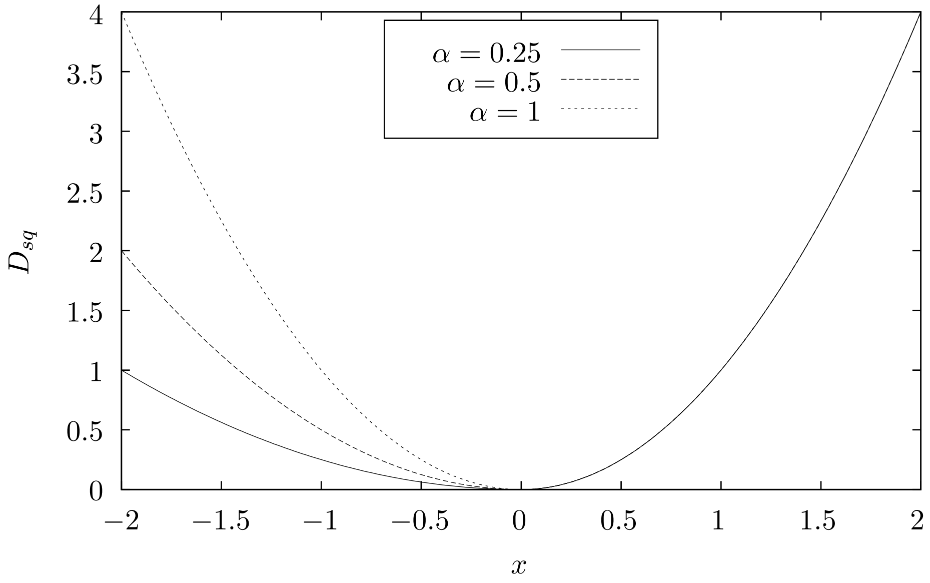

Figure 1: Function $D_{sq}(x)$ for various values of the parameter α.

Absolute difference: $F(s,v) = D_{sq}(s - v)$**.** If we assume $\alpha = 1$ in D_{sq}, the solution to (2) will be the weighted average for all votes:

$$\hat{s} = \sum_{i=1}^{k} w_i v_{(i)}.$$

In the general case, we should differentiate (2) for each interval $(v_{(i)}, v_{(i+1)})$, $1 \leqslant i < k$; plus test all margin points $v_{(i)}$. The optimal point on interval $(v_{(i)}, v_{(i+1)})$ can be found as

$$\frac{dL}{ds} = \sum_{j=1}^{i} 2w_j(s - v_{(j)}) + \sum_{j=i+1}^{k} 2\alpha w_j(s - v_j) = 0;$$

$$\hat{s}_i = \Big(\sum_{j=1}^{i} w_j v_{(j)} + \alpha \sum_{j=i+1}^{k} w_j v_{(j)}\Big) \Big/ \Big(\alpha + (1-\alpha)\sum_{j=1}^{i} w_j\Big). \tag{6}$$

The global optimum of the target function L

$$\hat{s} = \operatorname*{arg\,min}_{s \in S} L(s),$$

where a set of candidate points

$$S = \{\hat{s}_i\}_{i=1}^{k-1} \cup \{v_{(i)}\}_{i=1}^{k}.$$

Note that some $\hat{s}_i$ may not belong to the corresponding interval $(v_{(i)}, v_{(i+1)})$; we can exclude these points from the consideration. Furthermore, as the target function L is convex, we can skip checking marginal points $v_{(i)}$.

Relative difference: $F(s,v) = D_{sq}(s/v - 1)$**.** If we assume $\alpha = 1$ in D_{sq},

$$L(s) = \sum_{i=1}^{k} w_i (s/v_{(i)} - 1)^2;$$

$$\frac{dL}{ds} = \sum_{i=1}^{k} \frac{2w_i}{v_{(i)}} \Big(\frac{s}{v_{(i)}} - 1 \Big) = 0;$$

$$\hat{s} = \sum_{i=1}^{k} \frac{w_i}{v_{(i)}} \Big/ \sum_{i=1}^{k} \frac{w_i}{v_{(i)}^2}.$$

For $\alpha \neq 1$ we can get an analytical solution by differentiating L on each of intervals $(v_{(i)}, v_{(i+1)})$ and checking margin points, just as in the previous case.

Example. Consider the following vote distribution:

i	1	2	3	4
Vote $v_{(i)}$	0.5	1	1.5	2
Weight w_i	0.21	0.25	0.25	0.29

Examine the case (3) when dissatisfaction depends on the absolute difference between the vote and the target limit. First, when $\alpha = 1$, we get

$$\hat{s}(\alpha = 1) = 0.21 \cdot 0.5 + 0.25 \cdot 1 + 0.25 \cdot 1.5 + 0.29 \cdot 2 = 1.31.$$

Consider $\alpha = 0.5$. First, let's calculate target function at votes $v_{(i)}$:

i	1	2	3	4
Vote $v_{(i)}$	0.5	1	1.5	2
$L(v_{(i)})$	0.483	0.229	0.309	0.785

That is, vote for 1 is the optimal among all votes. Then, we calculate the optimal value according to (6) on three intervals:

Interval	0.5 to 1	1 to 1.5	1.5 to 2
$\hat{s}$	1.169	1.140	1.193

Among these three values, only the second one fits into the corresponding interval. As $L(1.140) = 0.214 < L(1)$, we can conclude that 1.140 is indeed an optimal block size limit: $\hat{s}(\alpha = 0.5) = 1.140$.

If we take $\alpha = 0.25$, we will obtain $\hat{s}(\alpha = 0.25) = 0.997$. Thus, the lower the value of α, the lower the optimal block size limit.

A.1.2 Exponential function

Consider a parameterized function

$$D_{exp}(x) = e^{\alpha x} + \alpha e^{-x} - \alpha - 1.$$

One can see that D_{exp} satisfies (5). $\alpha > 1$ corresponds to the case when a miner prefers a target limit less than his vote rather than an opposite situation. See Figure 2 for examples.

The derivative

$$D'_{exp}(x) = \alpha(e^{\alpha x} - e^{-x}).$$

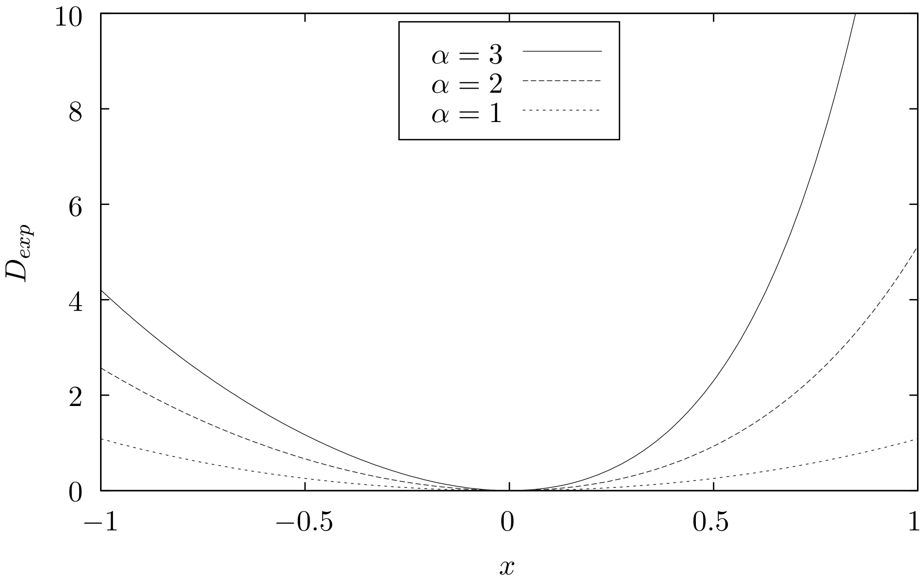

Figure 2: Function $D_{exp}(x)$ for various values of the parameter α.

Absolute difference: $F(s, v) = D_{exp}(s - v)$.

$$L'(s) = \sum_{i=1}^{k} w_i D'_{exp}(s - v_{(i)}) = 0;$$

$$\sum_{i=1}^{k} w_i \alpha \left[e^{\alpha(s - v_{(i)})} - e^{v_{(i)} - s} \right] = 0.$$

From here, we calculate the optimal block limit

$$\hat{s} = \frac{1}{\alpha + 1} \Big(\ln \sum_{i=1}^{k} w_i e^{v_{(i)}} - \ln \sum_{i=1}^{k} w_i e^{-\alpha v_{(i)}} \Big) \tag{7}$$

Relative difference: $F(s, v) = D_{exp}(s/v - 1)$. An explicit analytic expression for the optimal block limit is impossible in this case; one has to use computational methods to find it. As D_{exp} and consequently L are convex, calculating the optimal size limit is simple: one just has to find a single root of $L'(s) = 0$, e.g. by a binary search on $[v_{(1)}, v_{(k)}]$.

Example. Consider the same vote distribution as in the previous example. Examine the case (3) when dissatisfaction depends on the absolute difference between the vote and the target limit. Using (7), we get the following results:

α	Optimal block size $\hat{s}(\alpha)$
1	1.307
1.5	1.235
2	1.168
2.5	1.109
4	0.969

One can see that the parameter α indeed can be used to regulate the results of the vote: the higher the value of α, the lower the optimal block size limit. Note that when $\alpha = 1$, we get a result close to the weighted average of votes (which is equal to 1.31).

A.2 Further Research

One could introduce other functions F or D that may reflect dissatisfaction better than quadratic and exponential functions we have explored. Additionally, modeling needs to be performed in order to find optimal parameters of dissatisfaction functions (α in case of D_{sq} and D_{exp}).

References

[1] Number of transactions per day. Blockchain.info charts
URL: `https://blockchain.info/charts/n-transactions`

[2] July 2015 flood attack. In: Bitcoin Wiki
URL: `https://en.bitcoin.it/wiki/July_2015_flood_attack`

[3] Block size limit controversy. In: Bitcoin Wiki
URL: `https://en.bitcoin.it/wiki/Block_size_limit_controversy`

[4] *David Hudson* (2014). 7 transactions per second? Really?
URL: `http://hashingit.com/analysis/33-7-transactions-per-second`

[5] *David Hudson* (2014). Bitcoin traffic bulletin
URL: `http://hashingit.com/analysis/34-bitcoin-traffic-bulletin`

[6] Median transaction confirmation time (with fee only). Blockchain.info charts
URL: `https://blockchain.info/charts/avg-confirmation-time`

[7] Transaction fees. In: Bitcoin Wiki
URL: `https://en.bitcoin.it/wiki/Transaction_fees`

[8] *Bram Cohen* (2015). Bitcoin's ironic crisis
URL: `https://medium.com/@bramcohen/bitcoin-s-ironic-crisis-32226a85e39f`

[9] *Adam Back, Matt Corallo, Luke Dashjr et al.* (2014). Enabling blockchain innovations with pegged sidechains
URL: `https://www.blockstream.com/sidechains.pdf`

[10] *Joseph Poon, Thaddeus Dryja* (2015). The Bitcoin Lighting Network: scalable off-chain instant payments
URL: `http://lightning.network/lightning-network-paper.pdf`

[11] Scalability. In: Bitcoin Wiki
URL: `https://en.bitcoin.it/wiki/Scalability`

[12] Protocol documentation. In: Bitcoin Wiki
URL: `https://en.bitcoin.it/wiki/Protocol_documentation`

[13] Bandwidth usage. Statoshi.info (Retrieved on Aug 27, 2015)
URL: `http://statoshi.info/dashboard/db/bandwidth-usage`

[14] *Gavin Andresen* (2015). UTXO uh-oh…
URL: `http://gavinandresen.ninja/utxo-uhoh`

[15] Running a full node. Bitcoin.org
URL: `https://bitcoin.org/en/full-node`

[16] System metrics. Statoshi.info (Retrieved on Aug 27, 2015)
URL: `http://statoshi.info/dashboard/db/system-metrics`

[17] Unspent transaction output set. Statoshi.info (Retrieved on Aug 27, 2015)
URL: `http://statoshi.info/dashboard/db/unspent-transaction-output-set`

[18] Transactions. Statoshi.info (Retrieved on Aug 27, 2015)
URL: `http://statoshi.info/dashboard/db/transactions`

[19] Hardware & software survey. Steam
URL: `http://store.steampowered.com/hwsurvey/?platform=pc`

[20] *Gavin Andresen* (2015). Increase maximum block size (BIP 101)
URL: `https://github.com/bitcoin/bips/blob/master/bip-0101.mediawiki`

[21] Moore's law. In: English Wikipedia
URL: `https://en.wikipedia.org/wiki/Moore's_law`

[22] *Jeff Garzik* (2015). Making decentralized economic policy
URL: `http://gtf.org/garzik/bitcoin/BIP100-blocksizechangeproposal.pdf`

[23] *Gavin Andresen* (2012). Block v2, height in coinbase (BIP 34)
URL: `https://github.com/bitcoin/bips/blob/master/bip-0034.mediawiki`

[24] *Jeff Garzik* (2015). Increase block size limit to 2MB on Nov 11, 2015 (BIP 102)
URL: `https://github.com/bitcoin/bitcoin/pull/6451`

[25] *Pieter Wuille* (2015). Block size following technological growth
URL: `https://gist.github.com/sipa/c65665fc360ca7a176a6`

[26] *Gavin Andresen* (2014). O(1) block propagation
URL: `https://gist.github.com/gavinandresen/e20c3b5a1d4b97f79ac2`

[27] *Peter Todd* (2014). Tree-chains preliminary summary
URL: `http://www.mail-archive.com/bitcoin-development@lists.sourceforge.net/msg04388.html`

[28] *Yonatan Sompolinsky, Aviv Zohar* (2013). Accelerating Bitcoin's transaction processing: fast money grows on trees, not chains
URL: `https://eprint.iacr.org/2013/881.pdf`

[29] *David Gilson* (2013). Coinbase implements zero-fee microtransactions off the block chain. In: CoinDesk
URL: `http://www.coindesk.com/coinbase-implements-zero-fee-microtransactions-off-the-block-chain/`

[30] 21% attack possible against BIP100? Reddit
URL: `https://www.reddit.com/r/Bitcoin/comments/3id7f9/21_attack_possible_against_bip100/`