On Mitigating Scaling Issues of a Peer-to-Peer Electronic Cash System

Darren Tapp

May 4, 2017

Abstract

The year 2009 marked the launch of peer-to-peer electronic cash systems. These systems have proven to be resistant to attack. However, concerns about a potential denial-of-service attack (DoS attack) have prompted artificial limits on the transactions that such a system will process. We will explore the need for these limits. We introduce *economic mitigation* which may be more intuitive for end users. We also propose that economic mitigation can lead to a more organic way of moderating network growth.

1 Consequences of a DoS attack

We will divide DoS attacks against a peer-to-peer electronic cash system into two types.

DoS Attack 1 *A node or nodes broadcast many transactions in a bid to overwhelm the system.*

The effect of this attack is to increase the resource use of every node. Specifically, this attack increases bandwidth, processor, and memory use. This attack could also result in the economic transactions being pushed out by the attacking transactions, delaying or denying service.

DoS Attack 2 *A miner mines a block with an overwhelming number of transactions.*

This produces a block that is difficult for the network to process and store. The resource demands imposed by this attack are not as great as DoS Attack 1, however there is a need to store the large block in perpetuity. The costs of this attack are felt for all time as storage demands on a node increase. A sustained version of this attack will certainly require nodes to upgrade hardware, or shut down. In the case of one very large block, the time it takes to propagate around the network could cause a competing block to be created. This could lead to a large orphan rate for the attacker, but also could raise the orphan rate of non-attacking nodes.

2 Currently-applied mitigation of a DoS attack

In section 6 of [1] a fee on transactions was introduced. The introduction of a fee on transactions supplies a mining reward for blocks later in the progression of Bitcoin. It was also intended to reduce the economic incentive of a 51% attack. A fee also raises the cost of DoS Attack 1. If transactions generated by a node all have fees attached then there is a real cost to this attack. In the original implementation of Bitcoin the fee does not mitigate DoS Attack 2. However, if a version of the protocol is implemented whereby not all of a fee goes to the miner, then a fee also will increase the cost of this second attack.

The current implementation of the Bitcoin protocol also introduces a block size limit. This limit is enforced by the fact that all of the nodes in the system do not consider a block over a certain size to be valid. This indeed does mitigate the harm of DoS Attack 2. In fact, this makes DoS Attack 2 more or less impossible. This limit does reduce storage incurred by DoS Attack 1. However, this mitigating factor has the unintended consequence of making this first attack more successful as a denial-of-service attack.

The introduction of Xthin blocks or compact blocks makes the effectiveness of DoS Attack 2 less severe. If a miner does not publish transactions before a block is found then every node must download every transaction which will result in slower propagation, and a higher orphan rate for the attacker. The would-be attacker could broadcast transactions ahead of time, converting DoS attack 2 to DoS attack 1.

Current mitigation techniques have led to an unintuitive user experience. Users are unsure if their transaction will ever be confirmed. Many users do not understand how they should select a fee size. Even high-power users need to check the state of the network to understand what fee is required.

The current state of Bitcoin has also produced fees that are economically unintuitive. Many users price the fee in their local currency, and consider the fee in terms of a percentage of the amount they are sending. Sometimes this percentage is not considered to be competitive with traditional payment systems.

Current mitigation techniques have also led to the introduction of services that accelerate transactions in exchange for a traditional payment. The need for such a service completely undermines the usefulness of a peer-to-peer electronic cash system.

3 Alternate operating mode

It is useful to think through the extremes of possibilities. If we consider current Bitcoin to be on one extreme of an operating mode, then the other extreme would be Bitcoin without any cap on block size. Here we will explore the scaling of Bitcoin if there were no cap on block size. Let us explore actual risks of this scenario.

3.1 Methodology

We assume that Bitcoin has a working version of Xthin blocks which have been tested in production [2, 3, 4, 5, 6]. Bandwidth requirements of a node can be greatly reduced with Xthin blocks. Xthin blocks do not require a block to be downloaded, but rather transmit instructions to rebuild the block from existing transactions known by the node. We also assume that 144 blocks come out in a day, as targeted. However, Bitcoin blocks have come out at an average of 153.1 blocks per day since inception.

Without Xthin blocks our analysis would be different. With Xthin blocks we estimate overhead, including serving blocks, to account for 75% of bandwidth. Without Xthin blocks we would expect this to be over 90%. This analysis may not apply to Bitcoin during the time that it was growing, as Xthin blocks were still coming online. However, this analysis is intended for scenarios in the future, so we consider this assumption to be justified. Please note that Xthin blocks boost block propagation even more dramatically when dealing with censored environments, such as going through what is called the Great Firewall of China.

3.2 Nodes would shut down

Let us explore the idea that nodes would shut down if there were no cap. Currently, the cheapest desktop computer that Dell sells is $300 and has a terabyte of storage. This means that if the computer had 200 GB of overhead due to the operating system and pre-installed programs, downloaded the currently 110 GB blockchain, and continued to run a node, this drive with 590 GB of free space would only be full after

$$\frac{590,000}{6 * 24 * 365} \cong 11.62 \text{ years}$$

if every block is under one megabyte. Even if every block were 4 MB this node could record the entire blockchain over the three-year lifespan of a new computer. If there were no limit, it is reasonable to assume that transactions would have continued to have grown at an exponential rate. If we perform an exponential regression of the blockchain size from May 1, 2012 to Dec 31, 2015, we achieve a model

$$S = Ae^{bt} \quad \text{where} \quad A = 2223.51, \quad b = 0.0025886773, \tag{1}$$

t is the number of days since May 1, 2012, and S is the size of the blockchain in MB. We chose this date range because it is a time period after the blockchain started measurably growing every day, but we suspect early enough that the cap on block size did not influence our model. This model most likely may overestimate the blockchain size as it estimates the blockchain size to be 71372 MB at the end of 2015 when it was actually 53467 MB. The R^2 value obtained is 0.9507.

If this model continues, then our hypothetical computer could store the entire blockchain until late July 2018. If the original computer had a 2 TB drive available for $550 then the node could record the entire blockchain until June 1, 2019. However, a fully-functioning node could exist well past this time because of pruning. See [7] for more information.

The main requirement for running a node is bandwidth. A very low-end estimate for household internet speed is 3 Mbps. A household with 3 Mbps download speed can download 225 MB in 10 minutes. If this is reduced by a factor of 4 to account for overhead and rebroadcasting, a node could download 56.5 MB blocks every 10 minutes. When would our unrestricted Bitcoin model predict 50 MB blocks? To answer this question we take the derivative of (1) to get growth of blockchain per day and divide by 144, the number of blocks in a day. Solving this yields that this node would be able to keep up with the blockchain until around November 2019. This table gives us the estimated fail date for different download speeds.

Speed	Description	Block size	Estimated fail date
3 Mbps	Required for SD movie	50 MB	November 2019
5 Mbps	Required for HD movie	90 MB	July 2020
25 Mbps	Required for Ultra HD movie	400 MB	February 2022

Of particular note is that if Bitcoin's exponential growth had continued, blocks would be around 4 MB as of today, April 2017.

It is also of particular interest that 3 Mbps is just on the lower end of average of customer tests in China. In 2011, the U.S. state with the lowest bandwidth was Arkansas, with an average bandwidth of 3 Mbps. In Akamai's State of the Internet for Q3 2015 [8] the average connection speed in China was measured to be 3.7 Mbps, with 33% of customers having access to 4 Mbps or greater.

3.3 Mining would become centralized

Currently, Bitcoin mining is centralized through mining pools. Mining pools require excellent connectivity for their customers. Individual workers in the mine often run special hardware at home. Concern has been voiced that if the demands of running a node become too great then these individual miners may need to shut down. Ultimately, the concern is that this would lead to mining centralization.

This concern is directly related to our analysis of what is required to run a node. If a node can be run at home, then individual miners can mine from home.

3.4 Orphan rate will increase

The slow propagation of larger blocks can lead to a higher orphan rate. Mining pools with the best connection to the network would have the lowest orphan rate. In the absence of a block size limit some mining pools might decide to mine smaller blocks to lower the propagation time. Larger, more connected mining pools might be able to have a revenue advantage by mining larger blocks.

However, data center costs will be higher as well. Larger blocks encourage mining pools to upgrade, and thus strengthen the network.

With the block propagation speed improvement due to Xthin blocks measured in [2, 3, 4, 5, 6] a conservative fivefold increase in block size can be allowed without slowing block propagation. These effects are strengthened when going through a censored connection such as the Great Firewall of China.

Miners may broadcast smaller blocks to decrease the orphan chance. These smaller blocks would leave transactions unconfirmed and could lead to a backlog that will make Bitcoin show traits similar to what we are seeing today. However, these effects would not be as dramatic. This mode of a cryptocurrency leads to a better experience from an end user's perspective.

4 Economic mitigation

We consider alternate designs of a peer-to-peer electronic cash system. We suggest that such a cash system should value two qualities.

Desired Property 1 *End users should have a degree of certainty that transactions will be confirmed.*

Desired Property 2 *End users should view the pricing structure of transactions as being intuitive.*

The value of these desired properties is clear. A limit on block size has been proposed to mitigate DoS attacks 1 and 2 and also to maintain connectivity of nodes through tenuous internet connections. It is clear, though, that this limit can compromise our desired properties.

We first consider a small unit of value we call a "dot". We will discuss choosing this value later. For now it shall suffice that a dot is a small non-negligible unit of value much like a penny in the United States.

We suggest a user experience as follows. During a low-use period of the cash system, transactions are broadcast with a fee of 1 dot per kB of the transactions. If the use of the currency increases then transactions are broadcast with a fee of 2 dots per kB, then 3 dots, then 4, with the goal of all transactions being confirmed if broadcast with a fee of 5 dots per kB or more.

In order to be able to achieve desired property 2 we suggest that the hard cap condition be modified. A hard limit is a condition such as "A block is accepted if it is valid and under 2 MB." We recommend changing this to "A block is accepted if it is valid and under 2 MB or the fee density is over 4.2 dots per kB." This second condition can be verified only by inspecting the block. As written, there could be very large blocks. Current implementation has a message size of 32 MiB. We assume and recommend that blocks be limited by this size, until such time that the message limit can be raised. This second condition places user experience over that of miners.

We discuss an implementation of this applied to the digital cash system Dash. We assume that the maximum block size has been raised to 2 MB as agreed on by the masternodes. If two of the last three blocks are under 1.5 MB then the standard client broadcasts transactions of 1 dot per kB. If two of the last three blocks are over 1.5 MB but not both over 1.7 MB, then transactions are broadcast with 2 dots per kB price. If two of the last three blocks are over 1.7 MB then the standard client broadcasts a transaction of at least 3 dots per MB. If two of the last three blocks are over 1.7 MB and if two of these blocks have total fees over 5100 dots or more, then broadcast transactions with fees of at least 4 dots per kB. If two of the last three blocks have a size over 1.7 MB and two of these have total fees over 6800 dots, then broadcast transactions with fees of 5 dots per kB. If demand for use of the cash system continues to grow then fee density will be around 5 dots per kB and larger blocks will be allowed by the protocol.

This model is inspired by a pressure cooker. A pressure cooker is designed to keep pressure in the container, but there is a release valve. When the pressure becomes dangerous, the release valve can safely release that pressure.

When we say that the standard client broadcasts a transaction with a given fee density we have in mind a certain user experience. There should be a pop-up explaining what the fee is, and that it is a fee that should achieve a quick confirmation. The user would be asked to confirm the broadcast. If the user declines then they are allowed to set their fee, but receive a warning that their transaction may not be confirmed.

When Dash rolls out a third layer, then the masternodes should allow the third layer to query the current dot level for the light client to use to calculate the fee. If five out of seven agree then that should be accepted as the fee level. If there is more variance, then the maximum of up to 5 dots should be taken.

5 Results of economic mitigation

Simply put, economic mitigation achieves desired properties 1 and 2. Generally, a transaction with non-custom fee density should always confirm. If an attacker initiates DoS attack 1 then the client will generally adjust after 5 minutes, leaving at most 4 MB of transactions in limbo. Without the pressure release, a sustained attack could keep these transactions in limbo. With the pressure release, however, new transactions with the higher fee could pop the release, allowing for these transactions to be confirmed. With economic mitigation, DoS attack 1 is less successful. This assumes that most miners always include transactions with the highest fee density, which is in the economic interest of miners and is what is generally observed in practice.

Economic mitigation is designed to raise the price of transactions during periods of stress. If the block size ever decreases below a threshold then the fee resets to a lower value. This is a pricing structure that models business practices more faithfully. This allows for a surge-pricing structure. If Dash is mainly used in one country and use of it subsides during the evening, then end users can schedule transactions during off times to save on fees. This protocol is also designed to avoid the runaway fees that we have seen with Bitcoin.

By checking two of the last three blocks any attempt to manipulate the fee would be likely to be unsuccessful. An attacker with 40% of the hashing power would only have a 35.2% chance of capturing two out of the last three blocks. A miner with 10% of the hashing power only has a 2.8% chance of finding two out of the last three blocks.

Economic mitigation allows for a “Stress Mode” of sorts. Our proposed Stress Mode would be preferable to a mode in which user transactions fail.

Economic mitigation allows us to estimate the cost of DoS attack 1 and 2; this will allow us to evaluate the risk of these attacks. It would require 10,000 dots of fees to fill up a 2 MB block with a 5 dot per kB fee density.

Economic mitigation allows merchants and services to estimate their costs. An exchange could set a withdraw fee of 5 dots and be fairly certain that they would recover the network fees.

How should the cost of a dot be determined? There is no magic answer to this question. One way is to set a local currency target price of a dot, and a local currency target price of 1 dash. We can then use these targets to estimate the future experience of Dash users.

The reason 4.2 dots per kB was chosen for whether to accept a block was that this value will allow some wiggle room when most transactions are broadcast at 5 dots per kB. This should allow for PrivateSend transactions with lower fees. This also will encourage the clearing out of some transactions with a fee density of 4 dots per kB or lower.

6 Long Term Growth Management

If economic mitigation is adopted we could imagine a scenario in which blocks are averaging 3 MB and fees are consistently 5 dots per kB. Then one could imagine a situation in which the client is programmed to always broadcast all transactions at 5 dots per kB, and any size restriction on blocks is removed. This change will be needed for the expected growth.

It is the hope of the author that in such a case that the fee could be managed in such a way to responsibly grow the system. Currently a kB of the Bitcoin blockchain goes for about $2 in fees. It would seem that this is likely a maximum that demand will allow fees to be. The larger a fee is, the smaller demand will be for transactions. With the training wheels off, an observation of an equilibrium can be made. There may very well be a stagnation that takes place with a fee that is fixed in terms of dash. When a fee is fixed in terms of dash, then people will think of it as increasing and decreasing with the price of dash in local currency. If demand for dash and Dash transactions increases, then so will the price of transactions, which will lower demand for Dash transactions.

From an end user’s perspective, if the price of 5 dots per kB were consistent in terms of local currency, then end users could make decisions easier. If in the future this price ever goes down then users would welcome this change. Perhaps the fee could be used to manage growth of the network. If the network has unused capacity, lower the fee. One might suggest that many small fee decreases would be preferable to one severe fee reduction. With modest fee reductions the growth of the network could be measured and any need for raising the fee could be avoided.

Using the fee to manage growth of the network is analogous to the Federal Reserve adjusting interest rates. It is one parameter that affects everything else. Higher interest rates are believed to slow economic growth, and lower interest rates to encourage economic growth. The same can be said for fees. In a future with digital cash, perhaps monetary policy should be set by the fee instead of interest rates. In fact, the fee does change the effective interest rate of masternode holders, so maybe this is not such a new idea.

Perhaps a good starting value for a dot is 10^{-5} dash. This would result in a cost of $50 per MB if dash were trading at $1000 and the 5 dot stress mode were triggered. When the price of dash is trading at much lower than $1000 the client could broadcast a larger fee, but 10^{-5} dash per dot would allow the price to grow a bit over an order of magnitude without end users paying over $0.05 per kB.

7 Other Security Concerns and Thoughts

It is clear that masternodes are resistant to a Sybil attack. Currently, the high threshold required to stake for a masternode serves as an inhibitor for a Sybil attack. The introduction of distributed masternodes introduces new game-theoretic challenges required to preserve this same level of Sybil attack resistance. Economic incentives must be explored. Also related, the consequence of one masternode operator overseeing several distributed masternodes needs to be fully understood before deployment.

Dash intends to allow masternodes to vote on variables modified during a spork. Perhaps if masternodes voted on the key responsible for turning spork variables on and off, future difficulties such as coordinating masternodes during a spork could be avoided. If one entity has the key to turn variables on and off, then the ability to coordinate sporks would not be lost. Allowing masternodes to vote on this key will still have the desired effect of decentralizing spork decisions.

Current implementation of ECDSA requires the choice of two integers. Then a signature is computed with those numbers. Further, these numbers must be different, or the signature will reveal the private key. In algebraic geometry we think of this as a morphisim from the signature space to the affine plane. It would be interesting to explore the possibility that this morphism will factor through a fibration. If successful, the encoding of the algorithm could be such that only one projective point is needed to be chosen in a projective line. We would expect that an encoding of the resulting signature could be such that it would require fewer bytes. If the morphism factors, the ECDSA could be more efficient. The algorithm would be the same, just encoded differently. Therefore, no loss of security would occur. This would also eliminate one transaction malleability attack vector. If it factors, this could be done in a beautiful way, and not in an ad-hoc way. Since a projective point could be encoded as just one number if we avoid the case of an insecure signature, then the encoding could be such that this type of insecure signature would *not be possible.* This issue did arise on Android devices due to a poor random number generator.

8 Thanks

The author would like to thank Gavin Andresen, Will Anderson, and Josh Harvey for helpful discussion. He would also like to thank Will Anderson and Lisa McGunnigle for helpful review of the manuscript.

References

[1] Nakamoto, S. “Bitcoin: a Peer-to-Peer Electronic Cash System”

[2] Rizun, P. et al. “Towards Massive On-Chain Scaling: Presenting Our Block Propagation Results With Xthin” Medium. N.p., 30 May 2016. Web. 20 Mar. 2017.

[3] Rizun, P. et al. “Towards Massive On-Chain Scaling: Block Propagation Results With Xthin.” Medium. N.p., 31 May 2016. Web. 20 Mar. 2017.

[4] Rizun, P. et al. “Towards Massive On-Chain Scaling: Block Propagation Results With Xthin.” Medium. N.p., 04 June 2016. Web. 20 Mar. 2017.

[5] Rizun, P. et al. “Towards Massive On-Chain Scaling: Block Propagation Results With Xthin.” Medium. N.p., 04 June 2016. Web. 20 Mar. 2017.

[6] Rizun, P. et al. “Towards Massive On-Chain Scaling: Block Propagation Results With Xthin.” Medium. N.p., 06 June 2016. Web. 20 Mar. 2017.

[7] Bitcoin Core. Release notes for 0.11.0

[8] Akamai Q3 2015 State of the Internet

About the author:

Darren Tapp has been an actor in the Bitcoin space since 2011. He earned a Ph.D. in mathematics from Purdue University in 2007. He has been a technical writer and on-air personality for Neocash Radio since April 2013. He teaches algebra to high-school-age students as a hobby. He also provides business intelligence to a medium-sized company.