Faculty of Science and Technology
Department of Computer Science

Trading Network Performance for Cash in the Bitcoin Blockchain

subtitle

—

Enrico Tedeschi

Master Thesis in Computer Science

Abstract

Nowadays blockchain systems are emerging and they are spreading each day more. Cryptocurrencies are the biggest example of a physical implementation of this protocol, having in 2012 more than 50 thousands transactions per day and reaching now in 2017 more than 350 thousands of transactions approved every day. In this thesis we evaluate the most famous blockchain system, the *Bitcoin blockchain*. Public blockchains have emerged as a plausible messaging substrate for applications that require highly reliable communication. However, sending messages over existing blockchains can be cumbersome and costly as miners require payment to establish consensus on the sequence of messages. The blockchain protocol requires an always growing size of the informations stored in it so its *scalability* is the biggest problem. For that reason we collected data to be analyzed and stored in our own dataframe, saving up to $x10$ space for the analysis.

This thesis will consider the network performance of the Bitcoin public ledger when used as a massaging substrate. From 2009 to 2017 a lot of analysis has been done on Bitcoin blockchain and meanwhile its block size limit changed multiple times, from 256 bytes up to 1 Mb, the Bitcoin price raised from ~ 0.7 \$ to more than 4.000 \$ and different papers were published discussing whether changing or not the block size limit or talking about the fees a miner could get from clients. We read and considered previous analysis on Bitcoin blockchain, we then present our own dataset, which contains a significant portion of the Bitcoin blockchain, updated at 09-2017, discuss our results and compare them with other evaluations from past years, then we also discuss how the fee paid to miners evolves during time and how much a client could pay for a faster approval time, plus we take into consideration transaction visibility, blockchain growth and fees paid to miners. From this we propose and evaluate, using machine learning techniques, three different cost prediction models for predicting bandwidth per Bitcoin cost of upcoming transaction. The models can be used by application to throttle network traffic to optimize message delivery. We also discuss and consider, according to the data obtained, whether the block size limit should be increased for an higher *throughput* or not.

People are using Bitcoin because it has a lower fee rate and no central authority, we aim to find any possible relation between the fee paid from a transaction to a miner and the approval time of this transaction, plus we also noticed that the bigger is the blockchain size the more the system become centralized, since only few members, or nodes, of the Peer to Peer network can support and use the full blockchain.

Bitcoin blockchain has been analyzed with a blockchain analytics system, developed using Bitcoin's API and data were collected both by using the API and parsing `blockchain.info` HTML pages. A total of # transactions has been evaluated, more transactions than ever were considered before and useful informations about the Bitcoin blockchain emerged. This thesis gives also a measurement about accuracy of data provided from `blockchain.info`.

Contents

List of Figures

My list of definitions

1 Introduction

In 1964, Paul Baran [24] represented a very clear topology describing the differences between a centralized, decentralized and distributed network (Figure 1.1). Since then, the attention in developing systems moved from a centralized scheme to a distributed one, leaving most of the computation to every single user in the network rather than a central coordinator. Such a change might be easy for systems that do not require much of security, where authentication or authorization is minimal. However, the more a system needs to be secure, the more the decentralization process might be tricky as it becomes very important to rely on some trusted central coordinator. Systems that more than others need to be secure are the one related to e-commerce, banking and trades, all systems that have to deal with money.

In 1983, a research paper by David Chaum introduced the idea of digital cash [28]. In 1990 he founded *DigiCash*, an electronic cash company that closed because of bankrupt in 1998. After that, other systems such as *e-gold* (1996) and *PayPal* (1998) emerged. However, these systems allowed digital money transfer while they were still relying on a central authority. In 2008 Satoshi Nakatomo has presented Bitcoin [46], the first decentralized digital currency. Until 2008 e-commerce used to rely exclusively on financial institutions serving as trusted third parties. Those are involved in the electronic payments process and they have to guarantee consistency of the transactions and security of data.

Decentralized digital currencies are not dependent on any trusted third parties and they are built over a Peer to Peer (P2P) network where every component

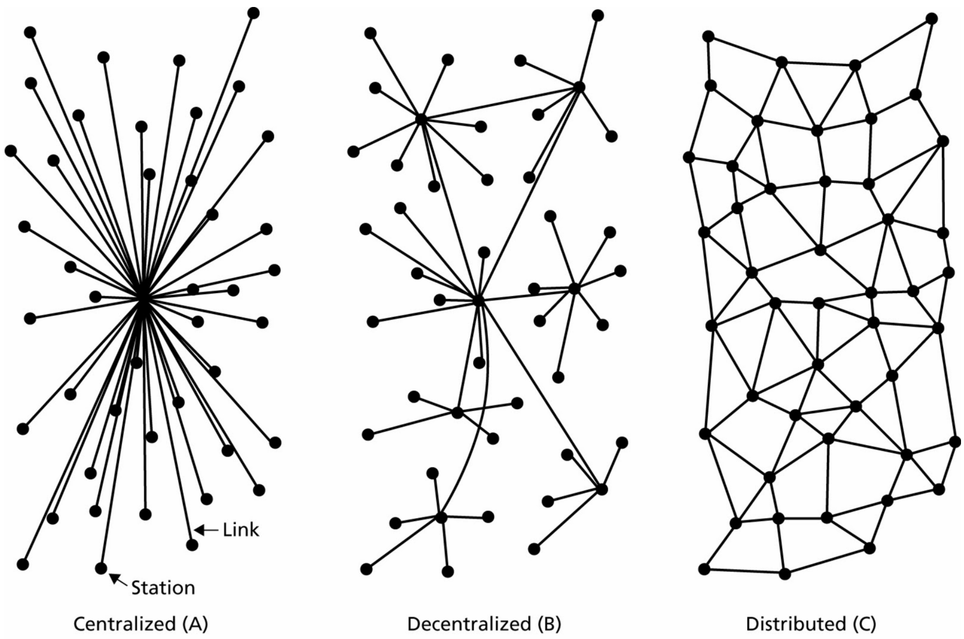

Figure 1.1: Differences between network topologies. Source: On Distributed Communication Networks, Paul Baran, 1964.

has the same privileges. These systems allow money exchange without a central authority, which means lower fees, no geographical separation and global trust among users. After Bitcoin, more decentralized digital currencies emerged, in 2011 *Litecoin*, originally based on the bitcoin protocol, then in 2013 Gavin Wood has presented *Ethereum* [55] and in 2014 *Monero* currency was released.

The order of transaction is essential in any cryptocurrency systems. However, establishing correct order can be problematic in decentralized cryptocurrency systems as they allow arbitrary nodes to join, including nodes that might be malicious. If arbitrary or Byzantine faults are allowed, the system might be left in an inconsistent or invalid state [40]. The ability to mask Byzantine faults has been implemented in various systems such as Byzantium [35], HRDB [53] and MITRA [42]. These protocol guarantees consistency of transactions having f faulty nodes, with a total of N nodes where $N = 2f + 1$ or $N = 3f + 1$, and a protocol like *Fireflies* [39] provides secure and scalable membership management and communication substrate in overlay network with Byzantine members. To guarantee an order of transaction all these cryptocurrencies rely on the *blockchain* protocol.

1.1 Blockchains

The need to tolerate malicious members was the reason for introducing the *blockchain* into cryptocurrency systems. The fundamental principle behind the blockchain is that consensus on transaction ordering is based on contributed computational power rather than number of participants. The blockchain works by appending transactions in blocks. Every block is generated after a relevant computation (*proof-of-work*), and each new block is appended to the public ledger of data, the blockchain, having in that way an ever growing chain of data containing every transaction ever happened.

Besides its use in cryptocurrency, this blockchain technology opens up to several usages in different sectors such as trading, file storage or identity management. Indeed it is already used by NASDAQ in its private socket market. If used in a P2P file sharing network, the blockchain removes the need of a centralized data base and heavy storage areas. Moreover it allows users to create tamper-proof digital identities for themselves. Blockchain technology opens up to usages in several important sectors such as trading, file storage, and identity management.

Blockchains essentially implements a distributed consensus protocols that enable a set of untrusted processes to agree on the content of an append only data structures. These ledgers are divided into blocks and linked together in sequence by hashes. They facilitate transactions between consenting individuals who would otherwise have no means to trust each other and deal with geographical separation and interfacing difficulties This technology promises a highly resilient and communication substrate where messages are kept potentially for a long time.

Nonetheless, decentralized digital currencies also have some side effects. The most relevant is *scalability*, due to the steady growth of the blockchain. It should be also considered that decentralized cryptocurrencies operate in open (or permissionless) networks in which the ledger of data could be manipulated from arbitrary adversaries and according also to the paper from University of Singapore [43] security of smart contracts has not received much attention yet. And since the only part not protected from cryptography is the *order of transactions* [11], an attacker would try to convince the network that a transaction occurred earlier than another one to gain money. The security bugs in smart contracts are classified as *Transaction-Ordering Dependence*, *Timestamp Dependence*, *Mishandled Exceptions* and *Reentrancy Vulnerability* [43]. In this thesis we refrain from explaining Bitcoin and its terminology in detail and refer the reader to already existing high-level [52, 27] or technical [46, 11]. description.

1.2 Problem Statement

While doing research, studying and reading papers related to blockchains, it turned out that the most urgent concerns are related to its scalability and performance, but also to the fee a client has to pay to get better latency. In 2015, Möser and Böhme write [45]:

> *Bitcoin may not be as cheap for consumers as it appears.* [. . .] *Bitcoin users are encouraged to pay fees to miners, up to* 10 *cents (United States Dollar (USD)), per trnsaction, irrispective of the amount paid.*

Rizun writes in 2015 [47]:

> *The block size limit was set at one megabyte, corresponding roughly to three transactions per second.* [. . .] *The transaction rate is over three hundred times larger than when the block size limit was introduced, and rising the limit is now being seriously considered.*

Then Croman writes in 2016 [29]:

> *The current trend of increasing the block sizes on Bitcoin protends a potential problem where the system will reach its maximum capacity to clear transactions, probably by* 2017.

It is obvious then that these problems need to be taken into consideration. In this thesis we discuss the scalability of the blockchain, how it affects the throughput and we present performance observations of the Bitcoin blockchain, analyzed with a blcockhain analytic system developed for this purpose. We provide detailed insights and analysis on how Bitcoin's characteristics, such as fee, block size and reward to different miners involved have changed over time, and provide an updated model describing how the Bitcoin blockchain will grow. We analyzed the correlation between the fee paid from a transaction and its *latency*, or the time it takes to be visible in the whole network. Three different models are proposed to describe how applications best can spend money to improve network charactertics, this affects average bandwidth available to an application.

1.2.1 Scalability

Scalability and network performances are urgent concern in existing Blockchain-based cryptocurrencies [29]. According to Ethereum white paper [11], if Bitcoin would have the same amount of transactions of a VISA circuit, its blockchain would grow about 1 MB every 3 seconds, $\sim$ 28GB per day, instead of the actual growth of ~ 0.12GB per day. In this thesis we discuss how much scalability affects centralization in Bitcoin network and how much it will impact in the next couple of years the blockchain growth.

1.2.2 Performance

Centralized schemes, like VISA are immediate, while having a throughput of 2000 transactions/sec up to 56 thousands transactions/sec [29]. It is true that Bitcoin has lower fees than centralized currency schemes, but these properties come at a performance and scalability cost. In the paper from Croman [29], they claim that Bitcoin achieve a throughput of 7 transactions/sec. In this thesis we also want to update at 2017 this statment and see how much a block size change might influence the whole network performance.

1.3 Method / Context

In this thesis we analyze a considerable part of the blockchain. In the paper from 2015 written by Möser and Böhme [45], they analyze tips and tolls in Bitcoin blockchain, they collected data until 2014 and they analyze more than 9 million of transactions. At that time there were a total of 100 thousands transactions per day, while today we count about 350 thousands txs/day, so the retrieving part turned out to be more time consuming than expected. Despite that, we aim to collect even a larger portion of the blockchain, storing data smartly in a *data frame*, which allows us to spare up to $10x$ of the space the blockchain actually requires. Then we analyze data and with *machine learning* techniques we define models, discuss about the results and how much they can be reliable in a future-wise implementation. In our data frame we store more than # transactions, with an analysis in between #date and #date. We used for the information retrieval Application Programming Interface (API) from `blockchain.info` combined with a HyperText Markup Language (HTML) parsing on every "block-page" of the same website.

Our assumption is that we can get sufficient information about the blockchain growth, the block creation time, the time for a transaction to be visible in the public ledger of data and which miners are the more trendy and which usually requires more fee by retrieving and analyzing only a portion of the blockchain, but having in that way a finer granularity than the one represented in the Bitcoin website. In that way we hope to gain more informations out of it. Moreover, sampling data from a single node in the blockchain gives statistics representative of the whole system.

For the analysis, data are retrieved block per block and part of the blockchain is saved in a text file. This finer granularity allows us to have a lot of informations that may be hidden in the statistical analysis provided from Bitcoin. It is also possible to use the informations retrieved to make future predictions about how much the Bitcoin blockchain will grow, using polynomial interpolation on the data. According on how many blocks ago are fetched, it is possible to have an accurate prediction on the blockchain growth for the next few years.

We are going to compare more recent data, retrieved real time, with the Bitcoin one and see the differences of the blockchain growth. Moreover, In the Bitcoin website for blockchain analysis, `blockchain.info` [5], the finer granularity shows data for the last 7 days while we are collecting and monitoring data at every block creation ($\sim$ 8-10 min). In that way is easier for us to check if there are any abnormalities in the ledger of public data.

1.4 Outline

2

Related Works - SotA

As mentioned in Chapter 1, scalability and analysis on the blockchain has been taken into consideration by many researchers in the past years. This chapter summarizes the most relevant papers or works that talks about Bitcoin, blockchain and decentralized cryptocurrencies, it gives a short view of what has already been done and the reults obtained. In our previous paper [51], we enhance the importance of paying for having a certain bandwidth in the Bitcoin network. A paper from Peter R. Rizun [47], explains how a rational Bitcoin miner should select transactions from his node mempool, when creating a new block, in order to maximize his profit. We discuss that and apply new data from more recent sources to this idea of "selecting the right transaction" in a way that a miner could select the right transactions to earn more money out of the whole mining process. Scalability has taken into consideration in the Position Paper of Kyle Croman [29], they analyze how fundamental bottlenecks in Bitcoin limit the ability of its current peer-to-peer overlay network to support substantially higher throughputs and lower latencies. We are going to test the throughput as well, comparing it with the one showed in this paper. Regarding fees and tolls paid in the Bitcoin blockchain we refer to the study done in 2014 from Möser and Böhme [45]. They analyze the entire blockchain and make assumptions about that these "fees" are supposed to substitute miners' minting rewards in the long run. This paper contributes empirical evidence from a historical analysis of agents' revealed behavior concerning their payment of transaction fees. Furthermore, to fully understand how is possible to make money out of the blockchain and mining, in necessary to have a view of how *VISA* [12] makes money as well.

2.1 Rizun - A Transaction Fee Market Exists Without a Block Size Limit [47]

2.1.1 Problems

A pressing concern exists over the ramifications of changing (or not) a Bitcoin protocol rule called *block size limit*. This rule sets an upper bound on the network's transactional capacity, or *throughput*. The limit was set at 1 Mb, corresponding roughly to three transactions per second. When this limit was set, it was over eight hundred times greater than what was required. However in 2015, blocks were filled near capacity and users experienced delays. In 2015 the transaction rate was over three hundred times larger than when the block size limit was introduced. One of the concerns is whether, in the absence of a limit or if the limit is far above the transactional demand, a healthy transaction fee market would develop which charges users the full cost to post transactions. The object of this paper is to consider whether or not such a fee market is likely to emerge if miners, rather than the protocol, limit the block size.

2.1.2 Methods

This paper shows how a Bitcoin miner should select transactions from his node's mempool when creating a new block in order to maximize his profit in the absence of a block size limit. *Block space supply curve* and *mempool demand curve* are explained, and the paper shows how the supply and demand curves from classical economics are related to the derivatives of these two curves. In the paper Rizun claims that the block-size limit determines the transaction throughput. In this paper he derives the *miner's profit equation* and then he introduces two novel concepts called the *mempool demand curve* and the *block space supply curve*.

Miner's Profit Equation

Every time a block is mined, the miner expects to generate a revenue $\langle V \rangle$ at hashing cost $\langle C \rangle$ to earn profit per block

$$\langle \Pi \rangle = \langle V \rangle - \langle C \rangle. \tag{2.1}$$

Miner's profit equation in 2.1 shows the gain of a miner $\langle \Pi \rangle$, where the hashing cost is represented as follows:

$$\langle C \rangle = \eta h T. \tag{2.2}$$

So the hashing cost $\langle C \rangle$ is directly dependent from the miner's individual hash rate, h, the cost per hash, η, and the creation time, T. Moreover, is important to consider the expectation value of a miner's revenue per block, this value is represented with $\langle V \rangle$ and is equal to the amount he would earn if he won the block multiplied by his probability of winning. So the expected revenue would be: $\langle V \rangle = (R + M)h/H$, where the amount he would earn is the sum of the block reward, R, and the transaction fees, M. His probability of winning, assuming all blocks propagating instantly, is equal to the ratio of his hash rate, h, to the total hash rate of the Bitcoin network, H. The problem with this equation is that it does not reflect the miner's diminished chances of winning if he chooses to publish a block that propagates slowly to the other miners. If a miner finds first a valid block, but his solution is received after most miners are working on another, then his block will likely be discarded. This effect is called *orphaning*. The equation, considering the orphaning factor, $\mathbb{P}_{orphan}$, is the following:

$$\langle V \rangle = (R + M)\,\frac{h}{H}\left(1 - \mathbb{P}_{orphan}\right). \tag{2.3}$$

Where P_{orphan} increases with the amount of time a block takes to propagate to other miners. Indeed, if τ is the block propagation time, the probability of orphaning is defined as:

$$\mathbb{P}_{orphan} = 1 - e^{-\frac{\tau}{T}}. \tag{2.4}$$

In conclusion the *miner's profit equation* is defined as:

$$\langle \Pi \rangle = (R + M)\frac{h}{H}e^{-\frac{\tau}{T}} - \eta h T \tag{2.5}$$

A *rational miner* selects which transactions to include in his block in a manner that maximizes the expectation value of his profit. This selection is explained with the *mempool demand curve* and the *block space supply curve*.

The Mempool Demand Curve

The set of transactions that still need to be approved and included in a block is called *mempool*. The mempool set is denoted with $\mathcal{N}$ and the number of transactions contained within it as n. According to the size limit, a block can select a $b \leq n$ transactions from $\mathcal{N}$ to create a new block $\mathcal{B} \subset \mathcal{N}$. A block first includes transactions with an higher *fee density*, ρ. This last, is a ratio between the *transaction fee*, t_f and the *transaction size*, t_q. To construct the mempool demand curve, is necessary first sorting the mempool from greatest fee density to least and then associating an index $\{i : 1, 2, \ldots, n-1, n\}$ with each transaction in the resulting list. The mempool demand curve will be then a graphical representation of the sum of the fees offered by each transaction in this sorted list:

$$M_{demand}(b) \equiv \sum_{i=1}^{b} fee_i, \tag{2.6}$$

and the sum of each transaction's size in bytes:

$$Q(b) \equiv \sum_{i=1}^{b} size_i. \tag{2.7}$$

The mempool demand curve represents then the maximum fee, $M_{demand}(b)$ a miner can claim by producing a given quantity $Q(b)$ of blockspace.

The Block Space Supply Curve

The size of the block a miner elects to produce controls the fees he attempts to claim, $M(Q)$, and the propagation time he chooses to risk, $\tau(Q)$. The block space supply curve represents the fees a miner requires to cover the additional cost of supplying block space Q. This cost grows exponentially with the propagation time. The equation which represents this curve is the following:

$$M_{supply}(Q) = R\left(e^{\frac{\Delta\tau(Q)}{T}} - 1\right), \tag{2.8}$$

where $\Delta\tau(Q) \equiv \tau(Q) - \tau(0)$. The propagation time ,$\tau$, is just an esteem from the propagation delay versus the block size.

Maximizing the Miner's Profit

To maximize his profit, the miner construct a mempool demand curve and a space supply curve. The block size Q^* where the miner's surplus, $M_{demand} - M_{supply}$, is largest represents the point of maximum profit. Considering this point Q^* of maximum profit, Rizun considers three market conditions for Bitcoin transaction fees: *healthy*, *unhealthy* and *non-existent*. In a healthy fee market, the miner's surplus is maximized at a finite quantity of block space, and thus a miner is incentivized to produce a finite block. In an unhealthy market, the miner's surplus continually increases with block space, and therefore a rational miner should produce an arbitrary large block. In a non-existent market, including *any* transactions results in a deficit to the miner, and so the miner is better off producing an empty block. A rational miner will produce a big block if his mempool is full of high fee density transactions, and will produce an empty block if no transactions pay a fee sufficient to offset the orphaning risk.

2.1.3 Results

In conclusion, they show that a transaction fee market should emerge without a block size limit if miners include transactions in a manner that maximizes the expectation value of their profit. A critical step in establishing this result was their calculation of the miner's cost to supply additional block space by accounting for orphaning risk.

2.2 Möser & Böhme - Trends, Tips, Tolls: A Longitudinal study of Bitcoin Transaction Fees [45]

2.2.1 Problems

The Bitcoin protocol supports optional direct payments from transaction partners to miners, also called *fees*. Acknowledging their role for the stability of the system, the right level of transaction fees is a hot topic of normative debate. The actual costs of the system are not extensively studied yet. Disregarding intangible factors of (in)convenience, Bitcoin may not be as cheap for consumers as it appears. The main problems/questions that this paper focuses on are:

1. Do higher transaction fees lead to faster confirmation?
2. Do impatient users offer higher fees?
3. Do mining pools enforce strictly positive fee systematically (excluding 0-fee transactions)?

2.2.2 Methods

They enhance the definition of transaction fee, which is encoded as difference between the sum of all inputs and the sum of all outputs of a transaction. Then to study trends of Bitcoin transaction fee conventions over the past couple of years, they combine data from different sources. They load the blockchain by parsing the block files of the Bitcoin Core client [2] and extract information on the size of the block and transactions. Additional data is fetched from `blockchain.info`, such as information about miners. Furthermore, data on bitcoin exchange rate is taken from `coindesk.com`, which provides an average Bitcoin price in USD. The time range selected for the analysis is in between January 2011 and August 2014. To answer question (1), they compared time when a transaction is first seen on the network and the timestamp of the block that includes the transaction, calculating in that way transaction latency, t_l. They analyze a representative subset of 9000 transactions randomly chosen from all eligible transaction between June 2012 and May 2013, then to answer question (2) they compute for each transaction the holding time, which is the period until the output was spent again, and compared their fees to see if they are higher. To answer question (3) is necessary get informations about major mining pools. They used data from `blockchain.info` to retrieve useful informations about miners and major miners were analyzed such as *AntPool*, *50BTC*, *BitMinter*, *Slush*, *ASICMiner* and more.

2.2.3 Results

Trends

Overall, they claim that Bitcoin transaction fees are lower than 0.1% of the transmitted value, which is significant below the fees charged by conventiaonal payment systems. It appeared to them that hard size limit do not (yet) significantly drive the level of transaction fees. In our thesis we want to test if this is still true. Regarding trends for the fees paid per transaction over time, the first notable change from 0 and 0.01 Ƀ fee occurs after June 2011, transactions with fee of 0.0005 Ƀ appear and account for about 20-30% of all transaction. In the second quarter of 2012 the transactions paying 0.0005 Ƀ raised to 60-70% of all transactions. In the fourth quarter of 2012, 30-40% of all transactions were paying a fee of 0.001 Ƀ. In May 2013, the nominal value of 0.001 Ƀ makes space for a tenth: 0.0001 Ƀ. This fee level stays on and gains a share of more than 70% towards 2015. In order to reason about these changes, they mapped important events in the Bitcoin ecosystem. Generally, there seem to be two main reasons for shift in trends: changes to the Bitcoin reference implementation and actions by large intermediaries in the ecosystem. The emergence of 0.0005 Ƀ fees in June 2011 can be mapped to the release of version 0.3.23 of the Bitcoin Core client, which reduced the default transaction fee from 0.01 Ƀ to 0.0005 Ƀ. The raise of these last transaction fees in the second quarter of 2012 is probably due to the launch of the gambling website *SatoshiDice* [16]. On May 2013, version 0.8.2 of Bitcoin Core was released.

Tips

There is a small share of transactions that did not offer fee to miners, most of them offered default fee amount but some of them were even willing to pay a higher fee. A plausible reason is that paying more in fee leads to a faster confirmation. After the analysis turned out that half of all zero-fee transactions had to wait more than 20 minuets for their first confirmation. In contrast to that, paying a 0.0005 Ƀ fee lead to an inclusion into a block in half of the time. 10% of all zero-fee transactions took almost 4 hours to confirm, in contrast to 40 minutes for transactions paying a 0.0005 Ƀ fee. The difference between paying 0.0005 Ƀ or 0.001 Ƀ fee is not as pronounced, but the difference in medians are still statistically and economically significant.

Tolls

Analysis on pool behavior regarding a possible systematic exclusion of zero-fee transactions has been done. Shares have shifted between pools quite extensively. In 2013, BTC Guild had a market share of up to 40%, in 2014 both GHash.IO and Discus Fish ousted this pool. Also, the share of other pools has risen in 2014. Previous incumbents like Slush or 50BTC have lost popularity. Possible reasons include economic and technical factors, like pool fees, service availability, or robustness against attacks. Given the dominance of a few mining pools, they evaluated whether some pools systematically enforce fees. The results show that two pools, Discus Fish and Eligius, have a considerably higher share of blocks without any zero-fee transaction, with 30.6% for Eligius and 62.5% for Discus Fish, in contrast to an average of 14.4%. Over than that though, there is no clear evidence for enforcement of strictly positive transactions fees.

2.3 Croman - On Scaling Decentralized Blockchains [29]

2.3.1 Problems

The increasing popularity of blockchain-based cryptocurrencies has made scalability a primary and urgent concern. The main question that this paper focuses on is the following:

> *Can decentralized blockchains be scaled up to match the performance of a mainstream payment processor? What does it take to get there?*

At the time of writing, the Bitcoin blockchain took 10 min or longer to confirm transactions, achieving 7 transactions/sec maximum throughput. Visa credit card confirms a transaction within seconds and processes 2000 transactions/sec ona verage with peaks of 56, 000 transactions/sec. This paper aims to place exploration of blockchain scalability on a scientific footing. Bitcoin community has put forth various proposals to modify the key systems parameters of block size and block interval. In this paper they show that such scaling by reparametrization can achieve only limited benefits. This because because Bitcoin generates a lot of network traffic, due to its decentralization. There are a lot of peers in the network and they all have to interact. To ensure that most of the nodes in the overlay network have sufficient throughput they set two guidelines:

- **Throughput limit.** The block size should not exceed 4 MB given 10 minutes average block interval. Corresponding at maximum 27 transactions/sec.
- **Latency limit.** The block interval should not be smaller than 12 seconds.

The community also proposed radically different scaling approaches, and introduced mechanisms such as Corallo's relay network, a centralized block propagation mechanism. One of the main contribution of this paper was to *quantify* Bitcoin's current scalability limits within its decentralized components. Their findings leaded them to the position that *fundamental protocol redesign is needed for blockchains to scale significantly while retaining their decentralization*. Plus, scalability is not a single metric and measurement and understanding of many important metrics, like *fairness* or *mining power utilization*, are lacking. Monitoring and measuring a decentralized blockchain from only a few vantage points poses significant challenges. In this paper they call for better measurements techniques, by continuously monitor the health of the decentralized system to answer key questions such as: "*To what extent can we push system paramteres without sacrificing security?*".

2.3.2 Methods

In this paper they manly focused on:

- **Maximum throughput.** At the time of writing (2016) maximum throughput was 3-7 transactions/sec. Number constrained by Q and $\mathcal{T}$.
- **Latency.** Time for a transaction to confirm, t_l. A transaction is considered confirmed when it is included in a block, roughly 10 minutes expectation.
- **Bootstrap time.** The time it takes to a new node to download and process the history necessary to validate the current system state. In 2016 that was roughly 4 days.
- **Cost per Confirmed Transaction (CPCT).** The cost in USD of resources consumed by the entire Bitcoin system to confirm a single transaction. It could be summarized in:
  1. *Mining:* Expended by miners generating the proof of work for each block.
  2. *Transaction validation:* The cost of computation necessary to validate that a transaction can spend the outputs referenced by its inputs, dominated by cryptographic verifications.

3. *Bandwidth:* The cost of network resources required to receive and transmit transactions, blocks and metadata.
4. *Storage:* The cost of storing all currently spendable transactions, which is necessary for miners and full nodes to perform transaction validation, and of storing the blockchain's historical data, which is necessary to bootstrab new node that join the network.

The cost per transaction for Bitcoin was calculated performing a back-of-the-envelope calculation by summing up the electricity consumed by the network as a whole, as well as the hardware cost of mining equipment. They projected their estimates based on the *AntMiner S5+* mining hardware [14]. They assume a 1 year effective lifetime for the hardware and that the average hashing rate of the network is 450, 000, 000 GH/s. Furthermore, they assume an average price per KWh of 0.1$. Two scenarios are possible, the first is when the Bitcoin network is operating at maximum throughput of 3-7 transactions/sec. This limit is constrained by the 1 MB block size limit and the variable transactions size. The lower bound is inferred from the average transaction size of 500 bytes, while the upper bound is based on an unusually small transactions size of 250 bytes. The second scenario is based on the average throughput of Bitcoin network, which is based on statistics collected in October 2015, and it resulted to be of 1.57 transactions/sec. They show then a Bitcoin cost breakdown assuming that the entire network contains 5400 full nodes and they evince that is a fallacy to assume that transaction costs necessarily have to be offset by transaction fees. Indeed, the costs of running full nodes may be offset by financial externalities such as selling items whose costs computational time for a node or confirming a transaction without trusting third parties. Said so, is important to enhance that miners are bereft of these two factors and they need to be compensated. According to new measurements on the block propagation time, they also defined **X% effective throughput** as follows:

$$\text{X\% effective throughput} = Q/(\text{X\% block propagation delay})$$

Considering that the propagation delay is, for 1 MB block size and 90 % of block propagation, 2.4 minutes, they calculated an X% effective throughput of 55 Kbps ≡ 26 tx/sec having a 90 % of block propagation.

Throughput limit

They observed that the block size Q, and interval $\mathcal{T}$, must satisfy:

$$\frac{Q}{\text{X\% effective throughput}} < \mathcal{T}$$

having in that way, for a 10 minutes block interval a block size that should not exceed 4 MB for $X = 90\%$ and 38 MB for $X = 50\%$. Given $\mathcal{T} = 10$ minutes the block size should not exceed 4 MB, corresponding to a throughput of at most 27 transactions/sec.

Latency limit

To improve the system's latency it could be enough to reduce the block interval. To maintain effective throughput that would also require a reduction in the block size. Propagating a block smaller than 80 KB would not make full use of the network's bandwidth, as latency would still be a significant factor in the block's propagation time. To propagate a 80 KB block to 90 % of the nodes would take roughly 12 seconds. In conclusion, to retain at least 90 % effective throughput and fully utilize the bandwidth of the network, the block interval should not be smaller than 12 seconds.

2.3.3 Results

More difficult to measure metrics could also reveal scaling limitations, example *fairness*. Their measurement results suggest that top 10 % nodes receive a 1 MB block 2.4 minutes earlier than the bottom 10 %, meaning that some miners could obtain a significant lead over others solving hash puzzles. In the end, they want to rethink the design of a scalable blockchain, organizing it around a decomposition of the Bitcoin system into a set of abstraction layers that they called *planes*. In a hierarchical of dependency from bottom to the top the layers are:

1. **Network Plane.** It propagates transaction messages. Bitcoin's network protocol do not fully utilize underlying network bandwidth, making Bitcoin's Network Plane the bottleneck in transaction processing. A solution could be to avoid denial-of-service by propagation of invalid transactions, a node must fully receive and validate a transaction before further propagations.
2. **Consensus Plane.** Functionality that mines blocks and reaches consensus on their integration in the blockchain. It receives messages from Network Plane and outputs transactions ready to be insert in the system ledger. Bitcoin's blockchain protocol has a three-way-tradeoff between, *consensus speed*, *bandwidth* and security, and the increment of two of them leads to the loss of the third. For example, if the first two are improved then there is loss in the mining power that secures the system.
3. **Storage Plane.** It functions as a global memory that stores and provides availability for authenticated data produced by the Consensus Plane. Storage Plane in Bitocin only supports `writes` operations that append data and doesn't support *delete* operations. The only supported *read*

operation downloads the entire ledger, a process that require four days. The community has proposed ideas such as Unspent Transaction Output (UTXO) data structure.

4. **View Plane.** A view is a data structure derived from the full ledger whose state is obtained by applying all transactions. For Bitcoin miners, it is unnecessary to operate on the full ledger that stores the entire transaction history. Miners and nodes in Bitcoin locally compute and operate on a view of the ledger called UTXO set, which specifies the current balance of all entities in the system. Bitcoin requires all consensus nodes to verify all transactions, and based on the result of the computation, every node needs to update its view, e.g. UTXO sets, locally and generating then the same conclusion of all other nodes in the system, representing in that way an honest set of consensus nodes, keeping an high availability.
5. **Side Plane.** Allows off-the-main-chain consensus.

/3

Technical Background

4

Blockchain Analytics System

5

Blockchain Observations

6

Conclusions

6.1 Discussion

6.2 Future Implementation

6.3 Comments

show bibliography [46], [55], [23], [33], [43], [31], [11], [17], [44], [48], [37], [39], [19], [35], [53], [42], [1], [7], [26], [8], [50], [24], [49], [5], [9], [10], [18], [21], [20], [41], [22], [13], [29], [34], [36], [14], [3], [4], [47] [6], [51], [32], [38], [12], [25], [45], [15], [54].

References

[1] Bitcoin api's, api-v1-client-python. https://github.com/blockchain/api-v1-client-python.

[2] Bitcoin client application. https://bitcoin.org/en/bitcoin-core/.

[3] Bitcoin mining hashing rate. https://blockchain.info/charts/hash-rate.

[4] Bitcoin nodes. https://bitnodes.21.co/.

[5] Bitocoin's blockchain website. https://blockchain.info.

[6] Construct a linear, no-fork, best version of the bitcoin blockchain. https://github.com/bitcoin/bitcoin/tree/master/contrib/linearize.

[7] Ethereum api's, pyethereum. https://github.com/ethereum/pyethereum.

[8] Ethereum wiki/patricia tree. https://github.com/ethereum/wiki/wiki/Patricia-Tree.

[9] Ethereum's blockchain analysis website. https://etherscan.io.

[10] Ethereum's blockchain website. https://etherchain.org/.

[11] Ethereum's white paper. https://github.com/ethereum/wiki/wiki/White-Paper.

[12] How visa make money. https://revenuesandprofits.com/how-visa-makes-money-understanding-visa-business-model/.

[13] Matplotlib for data plotting. https://matplotlib.org.

[14] Mining hardware comparison. https://en.bitcoin.it/wiki/Mining_

hardware_comparison.

[15] pandas: Python Data Analysis Library. http://pandas.pydata.org/, 2012.

[16] Satoshi dice - gambling with bitcoin. https://satoshidice.com/rules, 2012.

[17] Ethereum foundation - the solidity contract-oriented programming language. Technical report, https://solidity.readthedocs.io/en/develop/, 2014.

[18] Bitcoin mining process. http://bitcoinminer.com/, 2015.

[19] Bitcoin website – mining. https://www.bitcoinmining.com, 2016.

[20] Ethereum project – website. https://www.ethereum.org, 2016.

[21] tradeblock.com – analysis on the blockchain. https://tradeblock.com, 2016.

[22] Alfred V. Aho and Jeffrey D. Ullman. *Foundations of Computer Science*. Computer Science Press, Inc., New York, NY, USA, 1992.

[23] Adam Back. Hashcash - a denial of service counter-measure. Technical report, 2002.

[24] Paul Baran. *On Distributed Communications: Introduction to Distributed Communication Networks*. The Rand Corporation, 1964.

[25] Massimo Bartoletti, Andrea Bracciali, Stefano Lande, and Livio Pompianu. A general framework for bitcoin analytics. *CoRR*, abs/1707.01021, 2017.

[26] Georg Becker. *Merkle Signature Schemes, Merkle Trees and Their Cryptanalysis*. 2008.

[27] Rainer Böhme, Nicolas Christin, Benjamin Edelman, and Tyler Moore. Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives*, 29(2):213–38, May 2015.

[28] David Chaum. Blind signatures for untraceable payments. In D. Chaum, R.L. Rivest, and A.T. Sherman, editors, *Advances in Cryptology Proceedings of Crypto 82*, pages 199–203, 1983.

[29] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels,

Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. *On Scaling Decentralized Blockchains*, pages 106–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[30] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1–10, Sept 2013.

[31] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi. *Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab*, pages 79–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[32] Jacob Donnelly. Bitcoin network still backlogged with tens of thousands of unconfirmed transactions, causing delays. *Bitcoin Magazine*, July 2015.

[33] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '92, pages 139–147, London, UK, UK, 1993. Springer-Verlag.

[34] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert van Renesse. Bitcoin-ng: A scalable blockchain protocol. *CoRR*, abs/1510.02037, 2015.

[35] Rui Garcia, Rodrigo Rodrigues, and Nuno Preguiça. Efficient middleware for byzantine fault tolerant database replication. In *Proceedings of the Sixth Conference on Computer Systems*, EuroSys '11, pages 107–122, New York, NY, USA, 2011. ACM.

[36] Begnaud Francis Hildebrand. *Introduction to Numerical Analysis: 2Nd Edition*. Dover Publications, Inc., New York, NY, USA, 1987.

[37] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[38] Nicolas Houy. The economics of bitcoin transaction fees. Working Papers 1407, Groupe d'Analyse et de Théorie Economique (GATE), Centre national de la recherche scientifique (CNRS), Université Lyon 2, Ecole Normale Supérieure, 2014.

[39] Håvard D Johansen, Robbert van Renesse, Ymir Vigfusson, and Dag Johansen. Fireflies: A secure and scalable membership and gossip service. *ACM Transactions on Computer Systems (TOCS)*, 33(2):5:1–5:32, May 2015.

[40] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[41] Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

[42] Aldelir Fernando Luiz, Lau Cheuk Lung, and Miguel Correia. Mitra: Byzantine fault-tolerant middleware for transaction processing on replicated databases. *SIGMOD Rec.*, 43(1):32–38, May 2014.

[43] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 254–269, New York, NY, USA, 2016. ACM.

[44] Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. Demystifying incentives in the consensus computer. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 706–719, New York, NY, USA, 2015. ACM.

[45] Malte Möser and Rainer Böhme. *Trends, Tips, Tolls: A Longitudinal Study of Bitcoin Transaction Fees*, pages 19–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[46] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf, 2008.

[47] Peter R. Rizun. A transaction fee market exists without a block size limit. Technical report, 2015.

[48] Chaitya B. Shah and Drashti R. Panchal. Secured hash algorithm-1: Review paper. Technical report, Indus Institute of Technology and Engineering, Gujarat Technological University, 2014.

[49] William Stallings. *Cryptography and Network Security: Principles and Practice*. Pearson Education, 3rd edition, 2002.

[50] M. Swan. *Blockchain: Blueprint for a New Economy*. O'Reilly Media, 2015.

[51] Enrico Tedeschi. Paying for bandwidth in blockchain internet applications. Technical report, UiT Arctic University of Norway, 2016.

[52] Sarah Underwood. Blockchain beyond bitcoin. *Commun. ACM*, 59(11):15–17, October 2016.

[53] Ben Vandiver, Hari Balakrishnan, Barbara Liskov, and Samuel Madden. Tolerating Byzantine Faults in Transaction Processing Systems Using Commit Barrier Scheduling. In *ACM SOSP*, Stevenson, WA, October 2007.

[54] Michael Waskom, Olga Botvinnik, drewokane, Paul Hobson, David, Yaroslav Halchenko, Saulius Lukauskas, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Marcel Martin, Alistair Miles, Kyle Meyer, Tom Augspurger, Tal Yarkoni, Pete Bachant, Mike Williams, Constantine Evans, Clark Fitzgerald, Brian, Daniel Wehner, Gregory Hitz, Erik Ziegler, Adel Qalieh, and Antony Lee. seaborn: v0.7.1 (june 2016), June 2016.

[55] Dr. Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. Technical report, 2014.

A

Terminology

RLP: Stands for recursive length prefix. It is a serialization method for encoding arbitrary structured binary data (byte arrays).

KEC-256: Another serialization method generating a 256-bit hash.

full node: A full node in a decentralized digital currency peer-2-peer network, is a node that stores and processes the entirety of every block, storing locally the entire size of the blockchain.

light node: A light node in a decentralized digital currency peer-2-peer network, is a node that only stores the part of the blockchain it needs.

satoshi: Unit of the Bitcoin currency. 100,000,000 satoshi are 1 BTC (Bitcoin).

B

List of Symbols

t_B	transaction approved in a block B.
t_{in}	transaction input in bitcoin (Ƀ). All the money sent.
t_{ou}	transaction output (Ƀ). All the money received.
t_f	transaction fee (Ƀ). $t_{in} - t_{ou}$.
t_q	transaction size, in bytes.
t_l	commit latency of a single transaction. $B_{epoch} - t_{epoch}$.
$\mathcal{T}$	expected block interval time ($\sim 10\,\text{min}$)
$\mathbb{P}_{orphan}$	probability that given a block is orphaned.
τ	block solution propagation time, we consider a $\tau = 10$ seconds

according to Decker [30].

η cost per hash.

$\langle\Pi\rangle$ expectation value of a miner's profit per block.

$\langle V\rangle$ expectation value of a miner's revenue per block.

$\langle C\rangle$ expectation value of a miner's hashing cost per block.

R block reward, currently at 12.5 Ƀ.

h miner's individual hash rate.

H total hash rate of Bitcoin network.

Q block size or block space in bytes.

Q^* the block size that maximizes the miner's expected profit.

ρ fee density, or the price per byte for block space.

M money, bitcoin (Ƀ).

$M_{demand}(b)$ partial sum of the b transaction fees in mempool in order of descending fee density.

$M_{supply}(Q)$ miner's cost due to orphaning to produce a certain block size Q.

$\mathcal{N}$ the set of transactions in a miner's mempool.

n number of transactions in a miner's mempool.

B	single block.
B_t	transaction root that links to every transaction in a block B.
B_{epoch}	timestamp of a block B. Epoch of when the block was included in the blockchain
t_{epoch}	timestamp of a transaction t. Epoch of when t was first seen in the network

/C

Listing