Analysis of Reward Strategy and Transaction Selection in Bitcoin Block Generation

Elisabeth Senmarti Robla

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2015

Reading Committee:

Professor Radha Poovendran, Chair

Professor Linda Bushnell

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

Analysis of Reward Strategy and Transaction Selection in Bitcoin Block Generation

Elisabeth Senmarti Robla

Chair of the Supervisory Committee:
Professor Radha Poovendran
Electrical Engineering

Bitcoin was introduced back in 2009 and since then, much investment and research have focused on it. Key topics such as system vulnerabilities or the economic implications of leveraging an electronic currency have been widely examined. Other investigations have centered in analyzing specific parts of the system. Specifically, some work has focused on one of the key entities of the system, namely the miners, their activity and profitability. We extend this line of work to include transaction fees chosen by clients by presenting a complete analysis of the transaction fees and the implications for both the users of the system and the miners. In order to do so, we define specific models for clients, miners and the underlying peer-to-peer network based on observations made after analyzing historical data. Given this information, we examine the problem of choosing fees to pay for issuing a transaction and the selection of transactions added to a block by miners. We conclude that current strategies should be refined to address the expected growth in use in order to protect the long term sustainability of the system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I wish to express sincere appreciation to my advisor, Prof. Radha Poovendran, for all his help and support during the Master's program and the realization of the thesis. Along with that, I want to help the other committee member, Prof. Linda Bushnell, for her valuable input and comments. I also want to highlight the priceless support from the members of the Network Security Lab, Kali Mandal, Hossein Hosseini, Xuhang Ying, Phillip Lee, Andrew Clark, Jack Yang, Laila Abudahi, Zhipeng Liu and Sean Rice and every person in Seattle that I have met during the last 2 years. I also want to thank my family for their support and help during all these years.

DEDICATION

To my family

Chapter 1

INTRODUCTION

Modern currencies such as the dollar and the euro are controlled by governments and economic factors which determine their value and their operation. As a result, new distributed coin systems [33, 50, 53, 54] have recently appeared and are becoming increasingly attractive due to their anonymous and private nature and decentralized design. The most widely used currency, Bitcoin [56], was introduced back in 2009. Bitcoin's popularity, however, has been recently threatened by uncertainties such as the volatile conversion to physical currencies [25] or attacks against the system [2, 6, 28, 35, 43, 44, 45]. In order to maintain a continued adoption of Bitcoin, previous work has focused on the analysis of vulnerabilities of the system [5, 47, 52, 59], the interaction between involved users [30, 48, 51] and the economics and regulations [4, 27, 42, 46, 55]. However, little attention has been paid to the revenue scheme, which determines how new coins are created and the voluntary donation of small fees by the clients.

New coins are introduced into the system by miners, who generate blocks that include transactions created by clients. The process of minting coins is done by solving a proof-of-work [14] challenge that in average takes around 10 minutes. The first miner who successfully solves the problem receives a new coin (called fixed reward) plus the sum of all the fees associated with transactions in the block. Collecting the total revenue incentivizes miners to solve the challenge as fast as possible and obtain it before any other miner does. However, the value of the fixed reward halves every 4 years to control the supply of bitcoins. Currently, the fixed reward is 25 BTC and in 2140, it will be 0. On the other hand, fees are typically of the order of 10^{-4}, thus it is clear that any rational miner would disregard the fees in these circumstances. In contrast, miners are considering other factors such as total delay

or network consistency when selecting transactions to include in a block. In the long run, transaction fees, not fixed rewards, will incentivize the miners to create blocks.

The process of mining is currently profitable since the cost incurred from electricity and dedicated hardware is compensated by the expected revenue. However, it is clear that fees are not covering the expenses, but the fixed reward is. Another important point is that miners add less transactions to a block in order to improve their chances of getting the revenue. In the near future, two situations are expected to occur: decrease of the fixed reward and increase of the number of generated transactions due to the growing popularity of Bitcoin. Consequently, fees will play a key role in the decision of which transactions miners want to add in a block. Then, the overarching question becomes how the system will react to this new scenario. On one hand, clients will compete against each other to have their transaction included in a block; on the other hand, miners will still try to mine blocks as fast as possible but the main driving incentive will be the value of the fees. Only fees large enough to compensate the cost will incentivize miners to add transactions to a block.

In this thesis, we analyze two essential functions of the Bitcoin system: choosing transaction fees by clients and selecting transactions to put in a block by miners. In both cases, each participant intends to increase their revenue or, at least, to minimize the cost of using Bitcoin.

Our contributions are summarized as follows. Firstly, we analyze 6 months of transactions, from August 2014 to February 2015 and find correlations among parameters such as fees, market price or size of the blocks. Secondly, based on the observations, we model the behavior of participants (miners and clients) as well as we define a specific model for the underlying peer-to-peer network. The network model is important due to the influence of the computational delay [32]. With these models, we determine the optimum fee that a client needs to choose which is both fair to him and incentivizes the miner to add the transaction to the block. Thirdly, we present a model that describes the way that miners select transactions. We define a model for the current approach to the mining process, provide some improvements and discuss them. Finally, we conclude that aside from maximizing their rev-

enue, miners are also considering the sustainability of the system. Moreover, the current behavior of miners and clients is not suitable for handling larger volumes of transactions.

This thesis is organized as follows. In Chapter 2, we provide essential background on Bitcoin and present the main parameters and building blocks. Chapter 3 reviews previous work with a focus on mining pools and transaction fees. Chapter 4 presents observations on the actual system. In Chapter 5, we analyze the way transaction fees are chosen and in Chapter 6, we investigate the way miners select transactions to add in a block. Finally, our conclusions are presented in Chapter 7.

Chapter 2

BACKGROUND

Bitcoin is a peer-to-peer decentralized system that was first introduced in 2008 with the publication of a paper [56] by Satoshi Nakamoto. In January 2009, both the first implementation of the software [37] and the first block (also known as Genesis Block) were released and transactions started to occur. This electronic cash system is an anonymous payment scheme that allows the exchange of money between parties that are only identified by an alphanumerical address. For convention, it is commonly assumed that Bitcoin refers to the electronic cash system while bitcoin or BTC relate to the cryptocurrency. Satoshi is another way of expressing an amount and is equivalent to 10^{-8} BTC, the lowest possible amount.

2.1 How Bitcoin Works

There are three main sources of information about Bitcoin: the original paper written by Satoshi Nakamoto [56], the Bitcoin Wiki [7] and the software implementation *bitcoind* of the protocol [37]. However, there is no central authority so if a bug or a flaw is discovered, only if the Economic Majority [12] supports the proposed solution, an update is issued. Some changes have been already applied to the initial definition of the system [10].

One of the main properties of a currency is the ability to be easily transferred from one party to another. For that, Bitcoin users generate transactions that describe this exchange and release this information to the network. After being somehow verified, the transaction is stored publicly in a distributed ledger (also known as *blockchain*) that contains the whole Bitcoin history. Figure 2.1 shows the structure of both the transactions (left) and the blocks (right).

The main components of the system are the transactions and the blocks. Transactions

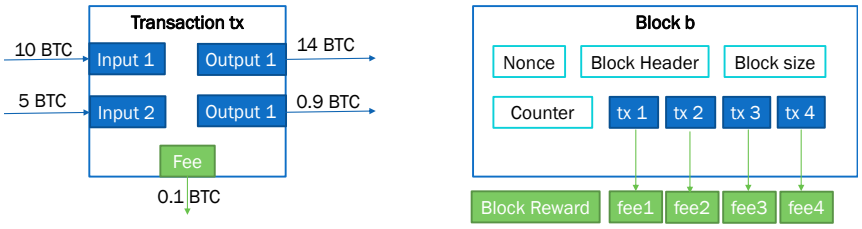

Figure 2.1: On the left, there is an example of the basic structure of a transaction, consisting some inputs, some outputs and a fee, which is equivalent to the difference between the inputs and the outputs. On the right, there is an example of a block, with four transactions. In this case, the total reward for the miner is the sum of all the fees and the fixed reward, which, at the time of writing, is equal to 25 BTC.

[17] are described by some inputs, some outputs and some verification steps. Inputs are defined by the hash of the transactions where they were last spent (a reference that include an id and an index that indicates which output is of the transaction) and a signature script (which includes the signature and the public key). Outputs are described by the amount of money sent and another script that describes, at least, the owner of the money (scripts can be highly complicated if desired and can require, for example, two private keys to spend the output). The difference between the total input and the total output constitutes the total fees of the transaction that will be collected by the miner of the block that includes this transaction in the ledger. Optionally, more complicated parameters can be defined, such as contracts [11]. On the other hand, blocks [9] consist of a header, which includes fields such as the previous block hash, and some of the previously defined transactions.

The system relies on public-key primitives to generate signed transactions that can be later easily verified and non-repudiated. The public-key scheme used in the Bitcoin protocol is ECDSA, more precisely *secp256k1*, with security equivalent to 2^{128} [15]. By using this algorithm, the signature issued with the private key can be efficiently verified with only the private key.

Bitcoin addresses are the identifiers of each input or output of any transactions. In other words, they are the designated name for some amount of money that belongs to a specific

party. They consist of a Base-58 encoding of the concatenation of the version (0x00), the hash of the key (RIPEMD-160 over the result of computing SHA-256 of the ECDSA public key) and its *checksum* (first 4 bytes of the computation of the hash function SHA-256 over the concatenation of the version and the result of the function SHA-256 over the hash of the key). Apart from issuing and signing transaction, Bitcoin addresses are heavily related to possession because the only way to claim ownership over some money is by possessing the private key. Since one of the main characteristics of Bitcoins is anonymity (the ledger is public, but the identity of the parties involved in the transactions isn't), once somebody loses the key, he also loses the associated money and there is no backup process to recover from it. This must not be confused with dormant bitcoins, which are money that people own but simply have not been spent in a long time. A related quantifier, namely Bitcoin Days Destroyed, is a value computed as the amount of Bitcoins of a transaction times the number of days since they were last spent.

The amount of bitcoins in circulation is limited to 21 million, a quantity that is expected to be achieved between 2110 and 2140. As explained in [5] while governments can increase the amount of coins according to the economic growth, Bitcoins can only be appreciated or depreciated. Consequently, a deflationary spiral could be catastrophic for the viability of the system. Another problem is the volatility of the value of the currency, which hardens the estimation of the evolution of the scheme. Since January 2009, Bitcoins reached its maximum value in November 2013 when its price was around $2150, but, at the time of writing, its value is around $235.

2.2 Architecture of the Bitcoin System

The Bitcoin system consists of a peer-to-peer network that interconnects the Bitcoin community, namely clients, nodes and miners, as shown in Figure 2.2.

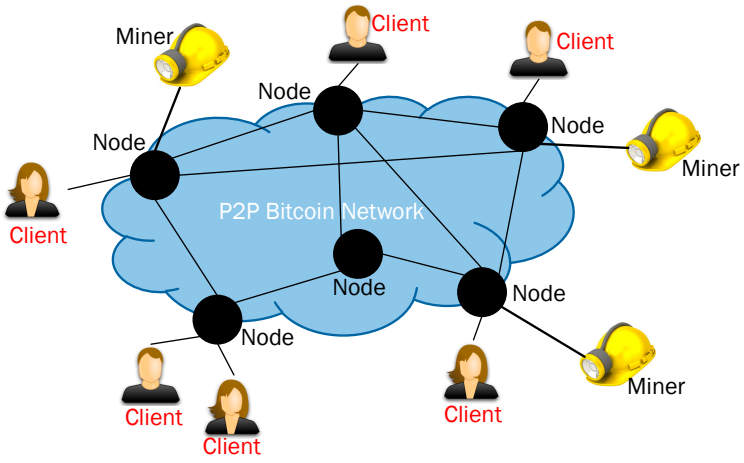

Figure 2.2: The Bitcoin network consists of the P2P interconnection of clients, miners and nodes over the Internet.

2.2.1 Clients

Clients (also known as lightweight clients) are the users of the e-cash system. They send and receive money from other clients that are also part of the network by creating online transactions that are recorded in the Bitcoin ledger. Clients are the only ones who possess both the public and the private key associated to the money that they own.

2.2.2 Nodes

Nodes are computer devices that sustain the Bitcoin network. On one hand, they store the latest up-to-date copy of the ledger and the transactions issued by the clients that haven't been included in a block, namely the unconfirmed transactions (Figure 2.3) . On the other hand, nodes create a mesh core network by being interconnected both to other nodes and to clients and miners. Their mission is to relay all the information (transactions and blocks) so data reaches all the interested parties in the network. Typically, each node can manage up to 8 outgoing connections and 117 incoming connections, even though in practice, those values can be modified.

Nodes can be setup by any interested user. At the time of writing, the amount of active

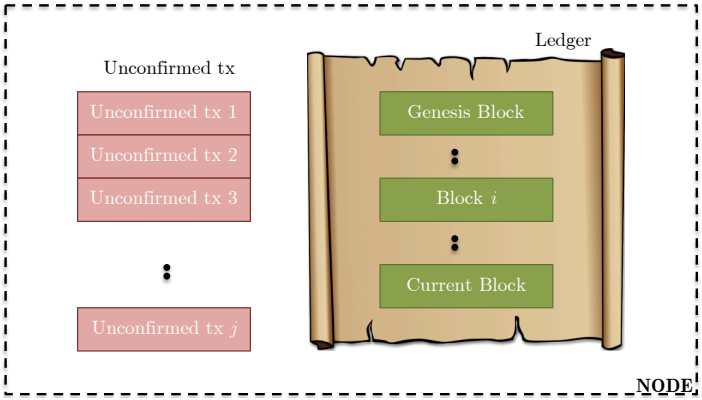

Figure 2.3: Every node contains an up-to-date copy of the ledger and the unconfirmed transactions issued by clients that want to be included in a block.

nodes fluctuates between 4000 and 8000 and the United States is the area with more deployed nodes, followed by Europe [24].

2.2.3 Miners

Miners are actors whose goal is to create a block that contains the transactions generated by the clients. This method is known as the mining process and once it is achieved, the block is relayed to the network and if it reaches consensus (it is the first one to be correctly created), the miner receives a reward for his work.

2.2.4 P2P Network

The Bitcoin network is a peer-to-peer network that runs over the Internet and deploys the Bitcoin protocol. Connections are unencrypted and work over a TCP channel and both IPv4 and IPv6 are supported. Since the network runs over the Internet, it is also subject to, for example, congestion problems or delays.

2.3 How Mining Works

Information about Bitcoin transactions is permanently stored in blocks. These blocks [9] are put together in a time-sequential ledger called block chain that is publicly available. Each block contains a magic number (4 bytes), the block size (4 bytes), the block header (6 bytes), a transaction counter (from 1 to 9 bytes) and a non-empty list of transactions. At the time of writing, the maximum size of the block is 1 MB. The block header contains the version of the block, the 256-bit hash of the previous block header, the 256-bit hash of the Merkle root of the transactions in the block, the current timestamp in seconds, the current target and the nonce.

The process of creating a new block is called *mining* and it is a resource-intensive computation also known as proof-of-work. This process of mining implies the creation (mint) of new Bitcoins, that are generated as a result of this block generation process in the form of a fixed reward for the miner. Although we will be using the notion of *proof-of-work* along the document, this concept is wrongly adopted because the Bitcoin mining puzzle can be solved really quickly with some low probability and only follows its description in expectation. In the Bitcoin system, this proof-of-work [14] consists of computing the hash SHA-256 of the block's header with different nonces until the result is less than a specific target. This target [16] is 256 bits long and is computed in such a way that the average amount of time it takes to solve this puzzle is around 10 minutes. The target is updated every 2016 blocks (around 2 weeks) for reasons of stability and low latency and the lower the target, the more the difficulty. It is directly related to the total mining power of the network (the more miners, the bigger the difficulty) and even if it increases, the mining process rate remains constant.

Mining [13] is an incentive-based activity because there is a reward associated to it. The first miner to get his block accepted by the network receives a fixed amount plus the sum of the fees of every transaction of that block. The fixed amount started to be 50 BTC and it halves every 210,000 blocks. Therefore, when the value of 210,000 blocks was reached for first time (in November 28th, 2012), the reward halved and at the time of writing, it is 25

BTC. The next halve (when block 420,000 is reached) is expected to happen around 2017 and the 34th halve is projected to happen in 2140. At that point, this fixed revenue will become 0 and that will be the end of the appearance of new coins. Then, the only reward will be the fees of the transactions. The projection of every halve can be consulted in [22].

In the current implementation of the Bitcoin Core software, there is a fixed way of choosing the transactions that will be included in the block [18]. However, all this settings can be modified and they are not guaranteed in custom implementations. There are two points that are used in the decision process: the fee and the priority of the transaction. The priority can be computed as [18]:

$$priority = \frac{input\,value \cdot input\,age}{size\,in\,bytes}. \tag{2.1}$$

Then, 50,000 bytes are used for the highest-priority transactions, regardless of transaction fee, following a highest-fee-per-kilobyte transactions first. Then, 700,000 bytes are filled up by transactions with a minimum fee of 0.00001 BTC/kb, also following the highest-fee-per-kilobyte rule. The remaining transactions will be candidates in the next block.

At this point, the amount of transactions every 10 minutes is lower than the maximum capacity of the blocks. However, since adding transactions to a block does not increase the difficulty of the problem (hashes are done over the header and it has constant size) miners can be tempted to add 0-fee transactions. This way doesn't incentivize transactions but makes the user see them as a volunteer donation. If that is changed, a proposed solution is to make nodes prefer blocks that contain at least a specific number of known transactions.

2.3.1 Proof-of-Work

In Bitcoins, the creation of a block consists of solving a proof-of-work challenge that takes, in average, 10 minutes to be solved. This proof-of-work is based on *Hashcash* [3] and is summarized in Algorithm 1 [31]. The process, shown in Figure 2.4, starts with the computation of SHA-256 over the 1024-bit message obtained by concatenating the version[1], the hash of

[1]Version 2 at the time of writing.

the previous block, the Merkle Root hash, the current timestamp[2], the value of the target, the nonce, the length and the padding data. Internally, it is done in two phases, one for the first 512 bits and another for the last 512 bits following the Davies - Meyer construction [58]. Secondly, the output of the first phase is concatenated with the last 256 bits of the length and padding field and the resulting bitwise string is hashed using SHA-256. Finally, the block is considered valid only if the obtained value is less than the current target.

In this process, only two parameters are variable, namely the nonce and the Merkle Root hash. The timestamp can only take, in average, 600 different values (one for each second during the 10 minutes average) so we can assume it to be constant too since modifying this field does not significantly increase the input space. Analyzing the algorithm, we can clearly see that in order to speed up the process, we can set the value of the Merkle Root hash and modify only the nonce field. This way, instead of computing a hash function three times, we compute the first hash once and for the 2^{32} possible nonces, we only compute the second and the third hash of Algorithm 1 [31]. In other words, the miner takes a specific number of transactions y, computes the Merkle Root hash and try the 2^{32} possible nonces. If none gives a valid block, the miner adds x transactions, he obtains a new Merkle Root hash and tries to solve the block again by trying the 2^{32} possible nonces. This procedure only stops when a valid block is found (here, we are not considering the case where another miner finds a valid block first).

2.3.2 Difficulty of the Mining Profit and Desired Target

Difficulty is defined as a measure of how hard it is to find a hash below a given target. Since it is desired to maintain constant the block generation rate of 1 block every 10 minutes, the difficulty is adjusted every 2016 blocks (around 2 weeks) based on the time it took to produce the previous 2016 blocks.

Another representation of the difficulty is the target. The target is an 256-bit number

[2]Value, in seconds, since Jan 1st, 1970 - 00:00 UTC.

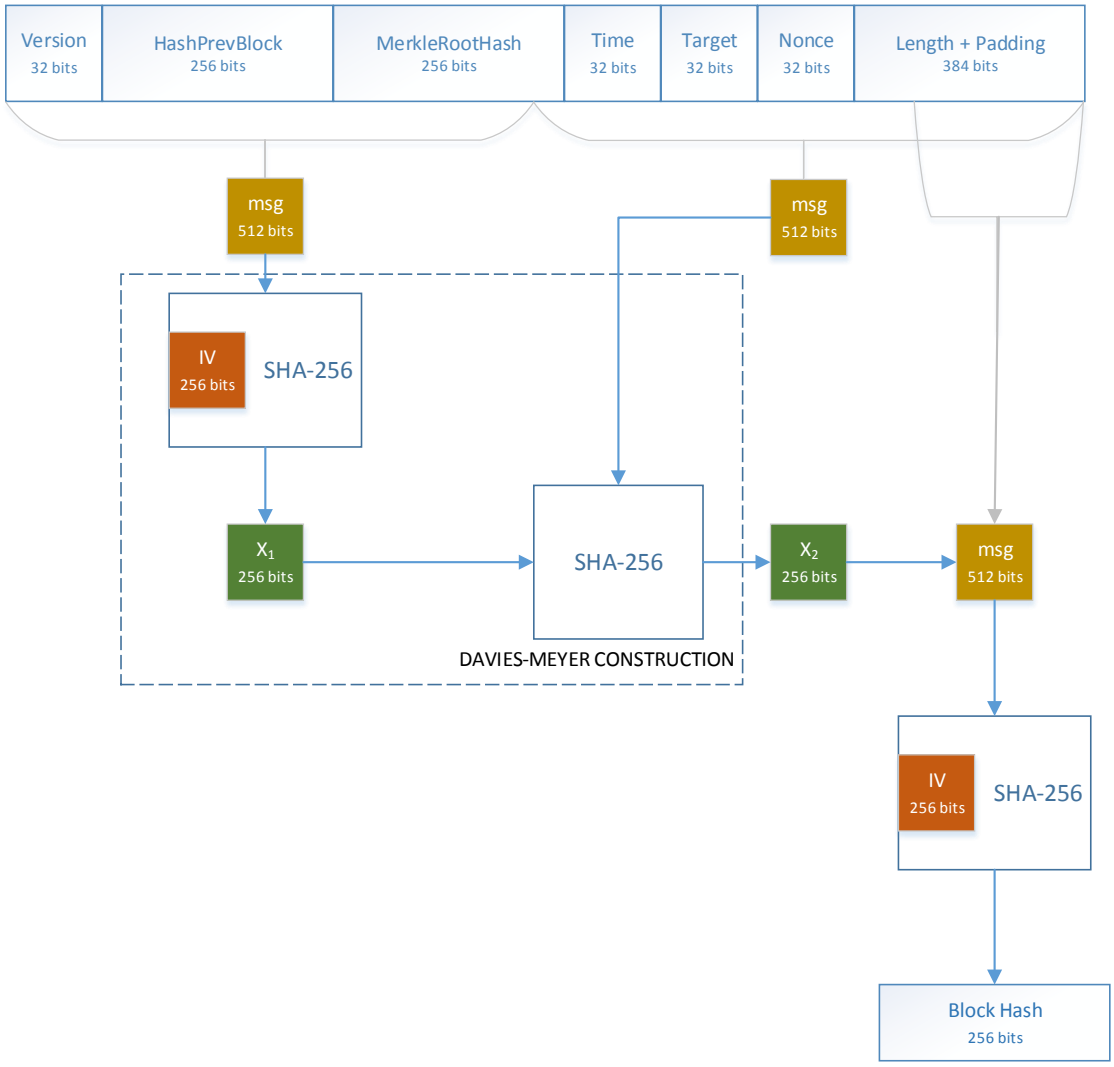

Figure 2.4: Algorithm for computing the hash of a block [31]. Only the Merkle Root hash and the nonce are variable so a result that is less than the target can only be obtained by changing those values.

Algorithm 1 Block Hashing: procedure for solving the Bitcoin proof-of-work puzzle

Input: version, hashPrevBlock, time, target, nonce, length and padding

Output: Block Header

---

function DAVIES-MEYER CONSTRUCTION [58]:

x_1 = SHA-256 $(IV, version||hashPrevBlock||MerkleRoothash)$

x_2 = SHA-256 $(x_1, MerkleRootHash||time||target||nonce||length + pad)$

return x_2

end function

---

procedure BLOCK HEADER HASH

Sort transactions in descendant order according to their fees

y = Select the first y transactions

do

MerkleRootHash = Compute Merkle Root Hash (**y**)

x_2 = DAVIES-MEYER CONSTRUCTION

Block Header Hash = SHA-256 $(IV, x_2||length + pad)$

Add a number x of transactions to **y**

while Block Header Hash > target

return Block Header

end procedure

that defines the maximum value that the hash of a block's header can take. Consequently, the lower the target, the more difficult it is to generate a block. The target is updated when the difficulty is updated and it is stored in every block's header in a compact form (4 bytes).

The value of the target for a difficulty equal to 1 is a 256-bit number with 32 leading zeros and 224 ones. Then, the difficulty is computed as:

$$difficulty = \frac{target\ (difficulty_1)^3}{current\ target}.$$

Consequently, if we express the target as a power of 2 value, we can directly obtain a more human-understandable information of the difficulty of mining:

$$target \simeq 2^{256-x}. \tag{2.2}$$

In Eq. (2.2), the exponent reflects the size of the target (256 bits) minus a value x that indicates the number of leading 0 that the hash needs to have. Furthermore, 2^{-x} is the probability of successfully finding a valid block by hashing a random data according to Algorithm 1.

With this information, a miner can compute the average time he will spent trying to find a block as:

$$time = \frac{difficulty \cdot 2^{32}}{hashrate}.$$

2.3.3 Consensus

As a distributed protocol, the Bitcoin system requires a method to ensure that all participants follow the directives of the protocol. Since there is no central authority to enforce them, Bitcoin trusts what the majority does so as long as the most of the players behave according to the rules, the system is viable and stable.

One of the problems faced by Bitcoin is that the process of transmitting information over a network is not instantly performed and there is some delay between the moment where

[3]Hash where the leading 32 bits are zero and the rest are one.

some data is relayed and the moment that the majority of the network knows about it. Consequently, it is possible that if two valid blocks are issued at similar times at different points of the network, some nodes accept one block as valid and some the other. That happens because the consensual rule is to accept the first received valid block, independently of what the other nodes of the network are doing. This situation is denominated a fork and it happens less than 1.8% of the time [32]. In this case, let's assume that some nodes added to the ledger block A and the rest added block B. Then, each miner tries to mine a new block considering only the block they received from the nodes (one of the fields in the header is the hash of the previous block). Assume that the next created block, block C, has in its header block A. With high probability, the next block will be propagated to the most of the nodes and those who considered block B as valid will remove it and update their ledger with both block A and block C.

Consensus is based on the policies that are currently accepted by the community. However, rules can be changed as long as the majority agrees on them. As an example, in August 15th 2010 there was a value overflow incident and 184 billion of new bitcoins were created [21]. In order to correct the problem, the community created the version 2 of the protocol by patching the existing software and forked the existing blockchain.

2.3.4 Profitability of Mining

The profitability of Bitcoin mining depends on a number of different factors that make it difficult to estimate a concrete value:

- The cost of the hardware: Mining can be achieved by a wide range of computer devices, from specialized hardware such as ASICS to personal laptops. Price varies according to the type and the specifications.

- The hashing power of the rig: Every hardware and every software implementation of the mining process have a specific hashing power associated which is relative to the total mining power of the system.

- The value of the current *target* hash, which determines the actual hardness to mine a block.
- The power consumption of the hardware and the cost of electricity.
- The current value of BTC.

All these variables change with time, which make it quite difficult to estimate both the current and the future profitability of mining.

2.3.5 Mining Pools

Originally, miners used to be individuals using optimal software implementation on their multi-core personal computers. However, as people realized how profitable Bitcoin could be, this evolved first to GPUs implementations and later to dedicated hardware such as FPGA and ASICS. On the other hand, miners stopped being just one computer-savvy individual and became a large amount of users organized in pools.

At the time of writing, there are two kind of pools: public pools and private pools. Public pools are well-known entities such as F2Pool and Ghash.io that consist of a supervisor that manages some number of individual users, probably hundreds and even thousands. Parameters such as the reward process, whether transaction fees are kept by the pool or distributed among miners or the percentage earned by the pools differ [19]. However, in general pools require their miners to compute a simpler proof-of-work (bigger target) so the supervisor can approximate the contribution of each miner and pay accordingly.

On the other hand, private pools are unknown entities of unknown size that are also mining blocks. In this case, only information about the IP address used to relay the block is disclosed.

Chapter 3

RELATED WORK

Current directions of Bitcoin research can be classified into three categories: 1) investigation and improvements on the system [5, 32, 47, 52, 60], 2) security studies and attacks [2, 6, 28, 35, 43, 44, 45], and 3) economic analysis and regulations [4, 27, 42, 46, 55].

In Bitcoin, miners are entities that consist of either an individual user or a pool of users organized in a specific manner. Pools have been widely examined, from the team formation and reward distribution point of view [48] to the best strategy for subversive miners [30]. Analysis of the payoff schemes in public pools was carried out in [51] and the percentage of public pools was concluded to be more than 70%. This brings up the conclusion that mining is no longer a distributed activity but an economic trade incentivized by rewards that can pose a serious threat to the survival of the system. Moreover, it is assumed that the top 1% owns at least 75% of the Bitcoins in circulation [40] [57] [23] so not only mining is concentrated, wealth is also skewed. On the other hand, attacks have not only focused on the overall system but also inside or among mining pools [34]. DDoS [41] can improve the chances for a miner to be the first one to mine a block, the pool-hopping attack [61] can increase the revenues when participating in multiple public pools and the block withholding attack [35] could potentially damage the function of the system by secretly creating a longer chain that at some point would replace the current one.

As presented in [56], the need to publish all the transactions negatively affects the privacy of the system. Consequently, the only solution is to keep public keys anonymous so there is no conclusive identification of the owner of the money. However, anonymity as the possibility of not being able to link bitcoins to a specific physical person has been recently proven wrong when applied to Bitcoins. In [6], Bitcoin identities are deanonymised by correlating

the IP address of the sender to his public key. In this case, no attention is paid to who the receiver is and this linkeability is ephemeral, since once the user changes his location (or shuts down the device), the found IP address is no longer valid. Other related work has focused on mixers as a way to help anonymize the current protocol [5]. In this case, an external service randomly exchanges bitcoins between users in order to obfuscate and bring anonymity. The main problem of this scheme is the way it is implemented (there is no protection against theft) so in [26], they presented signed warranties as a way to protect the user from misbehavior. However, the existing correlation schemes provide temporal mapping (when the user disconnects, the link between IP address and public key becomes useless); no distinction is being made between different clients behind the same IP address, and the receiver of the electronic cash still remains unknown.

Bitcoin is a decentralized system where the validation of the transactions is done by consensus. Since it is a distributed system, each participant in the network has a replica of the ledger and the whole truth is what the majority accepts as true. Each node works independently so they all verify the received information and hold their own truth in such a way that the protocol is stable [47]. However, Bitcoin works over a peer-to-peer network whose transmission delay is affected by the topology and the physical connections. In [32], both the synchronization mechanism and how transactions and blocks are disseminated were investigated. The propagation of the data was assumed to be similar to the randomized rumor spreading and they concluded that for blocks bigger than 20kB there is an additional 80 ms delay for each extra kB in size until the majority knows about the block. Another important result is that the probability of a fork was measured and computed to happen less than 1.8% of the times.

A simple but realistic mining model appeared in [47]. It assumes that each player will act in his own best interest and that he will only try to mine a block if the expected reward is bigger than the exptected cost (the expected number of guesses to solve the problem is less than the reward divided by the cost times the number of guesses per second that the miner is entitled to compute). Since mining resources are not acquired and paid in bitcoins,

cost fluctuates as the exchange rate varies over time. However, the conclusion that miners are incentivized to include in the block any transaction with a non-zero fee does not consider the internals of the peer-to-peer network over which Bitcoin operates. A second model [39] tried to add the effect of the network delay into the model, but the conclusion was that at the current time, the best strategy for miners is to put the minimum number of transactions (which is 1) which contradicts the observed data.

To the best of our knowledge, the strategy of the miners has been analyzed from an attacker point of view and not from the clients point of view and the inclusion of transaction fees in it does not appear in the existing literature.

Chapter 4

ANALYSIS OF BITCOIN DATA

Bitcoin has been up and running since January 2009 and the kind of transactions created and the reason for using Bitcoin has evolved. Consequently, parameters such as fees and size have been modified along the time. In this section, we will consider a 6 month snapshot, with data collected from August 18th, 2014 until February 18th, 2015. Since the number of users is increasing and the reason for using Bitcoin is changing (bigger and more complicated transactions), analyzing a bigger time frame window would lead to invalid data. Thus, the previously defined data is assumed to reflect the latest trends for blocks and transaction information.

This data has been obtained by downloading the whole public ledger and programming a *Java* parser that correctly stores all the information about blocks and transactions in a *mysql* database. Currently, the ledger has an approximate size of 20 GB and is stored in binary form in order to minimize the total size.

4.1 Blocks

Since the length of the block header is small (< 0.1Kb) and constant, the size of a block mainly depends on the number and the size of each included transaction. The left-hand side of Figure 4.1 shows the distribution of the size, expressed in bytes, with mean 316 KB and variance $6.4 \cdot 10^{10}$. On the other hand, the number of transactions per block is decided by the winner of the proof-of-work challenge. As it is shown in the right-hand side of Figure 4.1, it follows an exponential distribution whose mean is 547 and its variance is $223,240$.

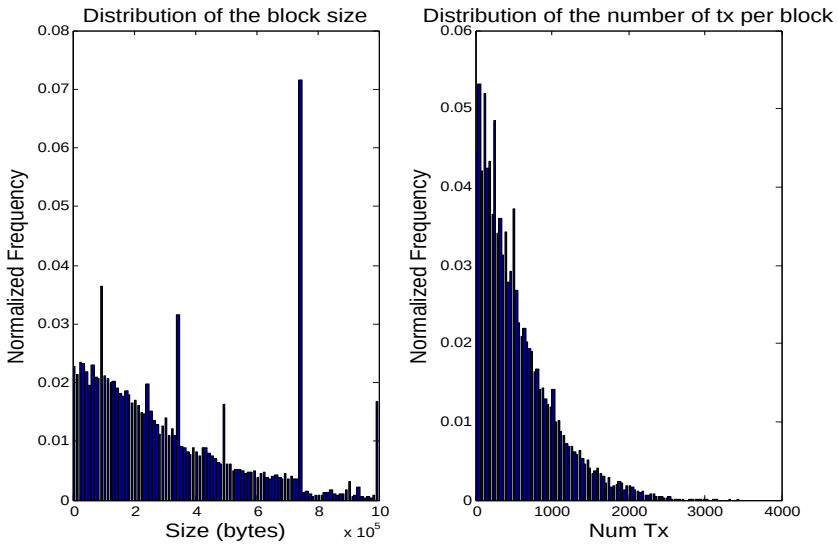

Figure 4.1: On the left, there is the distribution of the block size from August 2014 to February 2015, with mean 316 KB and variance $6.4 \cdot 10^{10}$. On the right, there's the distribution of the number of transactions per block from August 2014 to February 2015, with mean 547 and variance $223,240$.

4.2 Transactions

The size of a transaction depends mainly on the number of inputs and the length of the scripts that are part of the transaction. Its distribution is shown in the left-hand side of Figure 4.2 and it has a mean of 603.5 bytes and a variance of $3,819,900$. Another interesting parameter is the donation that a user gives away in order to reward miners, namely the transaction fee. It can be any positive value less or equal to the total input of the transaction. Its distribution is shown in Figure 4.2, with mean $1.64 \cdot 10^{-4}$ BTC and variance $9.6 \cdot 10^{-5}$.

At this point, assuming that the average size of a transaction is 603.5 bytes, the maximum block size is 1 MB and the total size of the rest of the fields is less than 0.1 KB, the maximum average number of transactions per block X is:

$$X = \frac{Max\ Block\ Size\ -\ Block\ Fields}{Average\ size\ of\ a\ transaction} \simeq 1660. \tag{4.1}$$

An unconfirmed transaction is a transaction that has not been added to a block yet. It

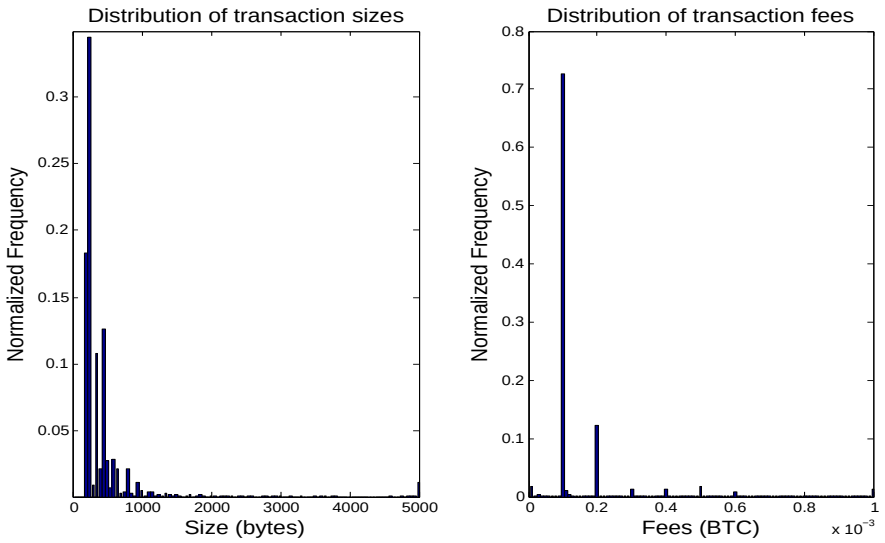

Figure 4.2: On the left, there is the distribution of the transaction size from August 2014 to February 2015, with mean 603.5 bytes and variance $3,819,900$. On the right, there is the distribution of the transaction fees from August 2014 to February 2015, with mean $1.64 \cdot 10^{-4}$ BTC and variance $9.6 \cdot 10^{-5}$.

has been observed that it highly fluctuates and is of the order of 10^3 [20].

4.3 Pools

The hash power of a miner is defined as the relative computational power with respect to the whole Bitcoin community. In order to approximate it, we observed the amount of blocks mined by every entity or pool over 6 months and obtained every participant's share. Consequently, this measurement includes not only the invested hardware but also the implementation and the strategy of each miner (Figure 4.3).

Until this point, we analyzed the distribution of parameters such as fees and sizes in a six month period. In the remaining of the section, we will investigate the evolution of these parameters and the correlation among them.

First of all, in Figure 4.4 we can see the progression of both the total fees per block and the price, in USD, of 1 BTC. There is no correlation among these two parameters, which

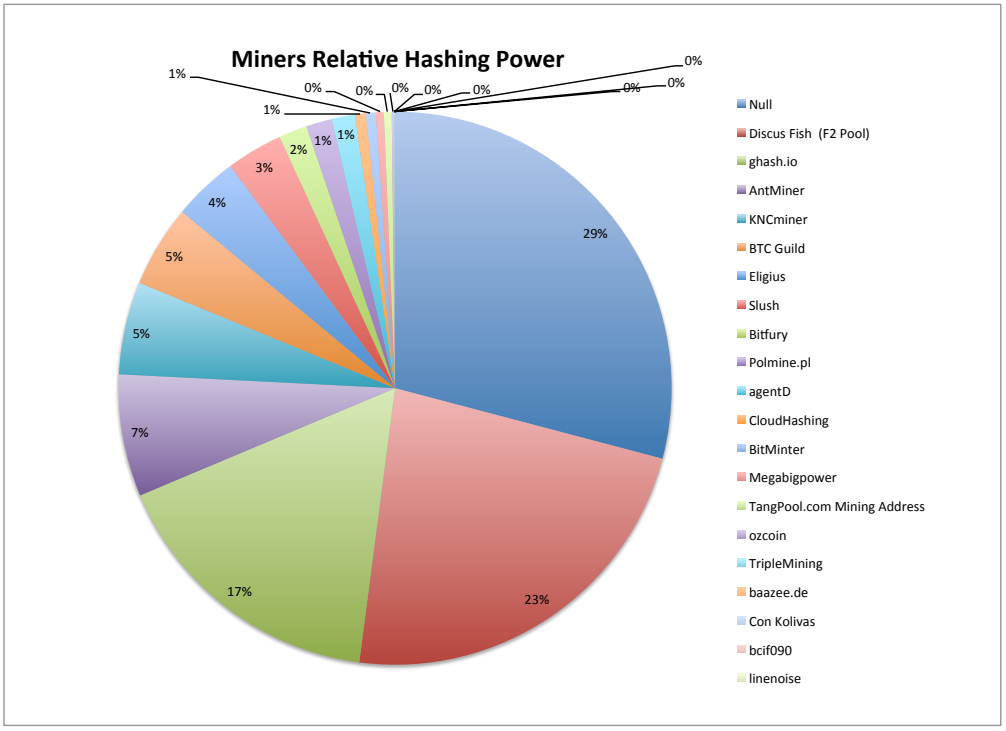

Figure 4.3: Distribution of the relative hash power of the biggest mining pools. This data is equivalent to every miner's number of mined blocks between August 2014 and February 2015.

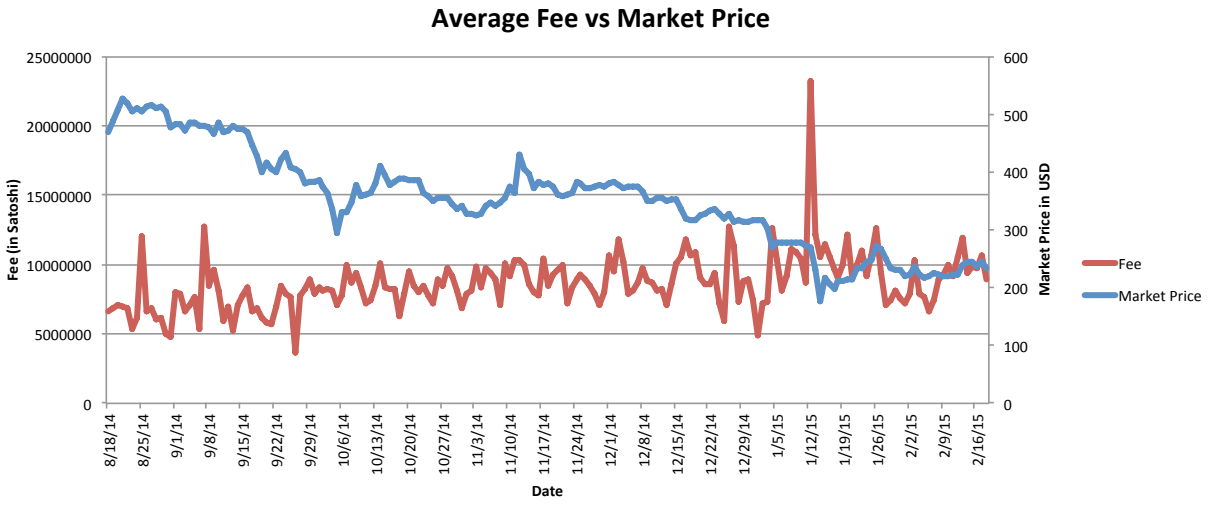

Figure 4.4: Evolution of the average block fee (in Satoshi) versus the market price of 1 BTC in USD, from August 2014 to February 2015.

lead to the conclusion that clients are not considering the real cost of mining (which is a function of the price of the electricity) when rewarding them.

On the other hand, Figure 4.5 shows the gradual change of both fees and the block size. It is interesting to note that fees are not affected by the relationship between physical currencies such as the dollar (USD) and the bitcoin (BTC) and that they have remained more or less constant along the time. Consequently, as shown in Figure 4.6, while the price of 1 KB of a block remained stable, as the exchange to USD has decreased, the total fees received by the miners has decreased.

In Appendix A, Table A.2 and Table A.1 reflect the total number of mined blocks (and therefore, the relative hash power rate of each pool), the average block size, the average block fee (expressed in Satoshi), the average time it took to mine a block, the average number of transactions per block and the average input per block (also expressed in Satoshi) belonging to each pool. Data was obtained by analyzing data from August 2014 to February 2015. Values related to time are obtained from the blocks' timestamp and since entities in the Bitcoin network might not be fully synchronized, this data can be slightly inconsistent.

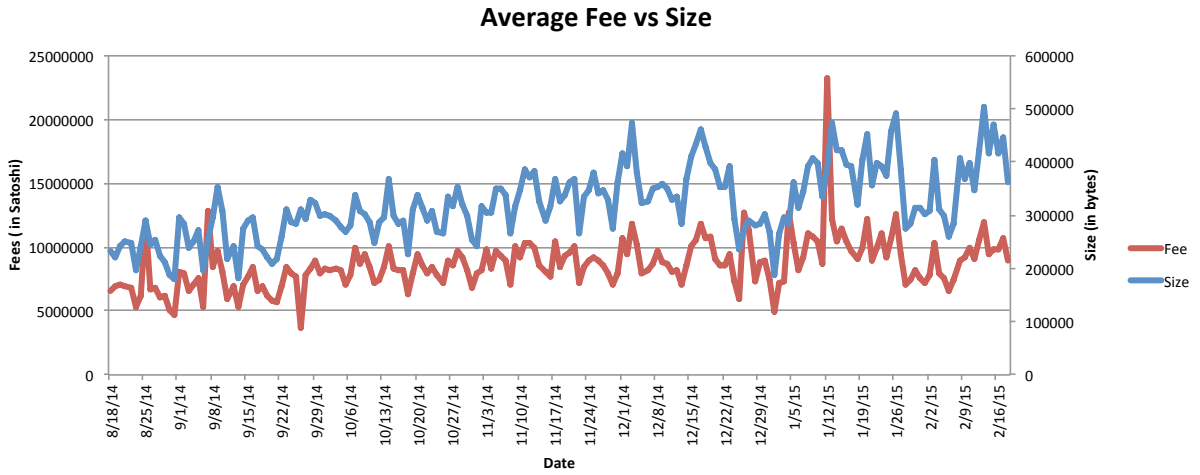

Figure 4.5: Evolution of the average block fees (in Satoshi) versus the size of the block per day (in bytes), from August 2014 to February 2015.

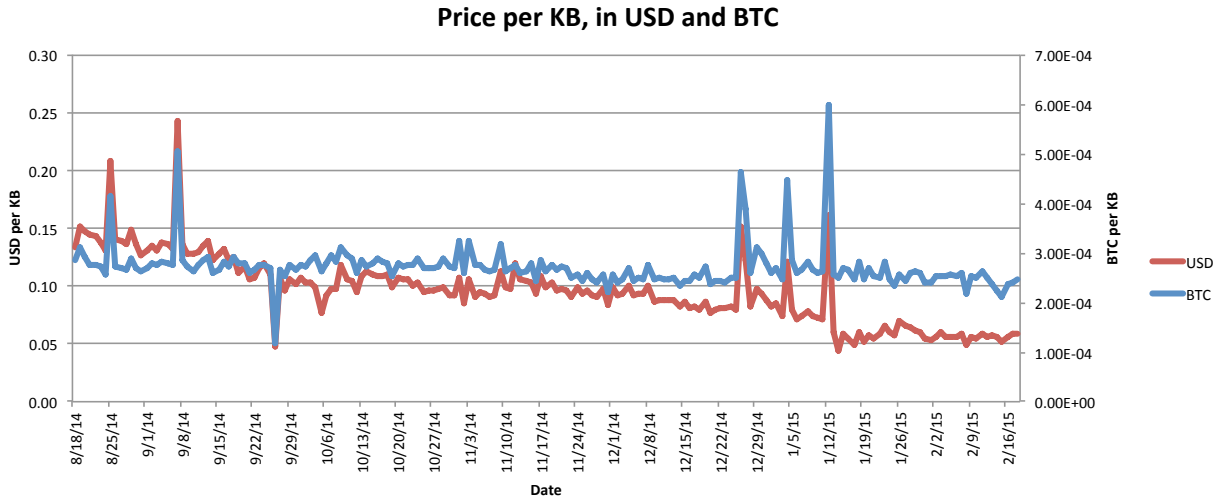

Figure 4.6: Evolution of the price, both in USD and BTC, per 1KB of size, from August 2014 to February 2015.

Conclusions are summarized as follows:

- Although Ghash.io reached a 55% of market share in June, 2014 [36], this value has decreased to around 17% so the fear for the 51% attack has been neutralized. Even though the company made clear that they had no intention to damage the Bitcoin system, people were still preoccupied about one party having so much control and started to join other pools.

- There is a big percentage of hashing power that belongs to unknown parties. In the tables, we put them together under the name "Unknown" because information about them is limited and only the address used when the block was relayed is publicly available. Some of them might be the same over time, with different IP addresses, because the investment on hardware is high enough to think that users want, at least, to amortize it. That increases the difficulty of successfully identifying them and tracking their evolution.

- In general, data from the last 8 pools (Megabigpower, TangPool.com, ozcoin, TripleMining, bcif090, linenoise, baazee.de, Con Kolivas) is not conclusive because information is scarce and they mined less than 0.01% of the blocks in the 6 month time frame analyzed.

- In general, there is a correlation between the average size and the average values of fees, total inputs and number of transactions so the bigger the size, the higher the rest of the parameters. This leads to the conclusion that miners are including not more than a few zero-fee transactions.

- As an exception, Poolmine.pl data shows slightly uncommon patterns. On one hand, the size, the number of transactions and the fee per block are unusually low, but at the same time, the average time necessary to mine a block is much bigger than the rest. In this case, we can conclude that this pool tries mine a low number of transactions, even

as time goes by, and they get the highest fee/KB, which during that time was $4.2 \cdot 10^{-4}$ BTC/KB. On the contrary, the rest of the pools were obtaining between $2.6 \cdot 10^{-4}$ and $2.9 \cdot 10^{-4}$ BTC per KB.

Chapter 5

DETERMINATION OF TRANSACTION FEES

The current implementation of the Bitcoin protocol does not enforce the use of transactions fees, which are optional and chosen independently by the clients. On the other hand, since the fixed reward is significantly big compared to fees, miners are not incentivizing clients to choose them wisely either, thus fees are typically forgotten in the analysis of the system. However, this situation will change. The fixed reward per block is halved every four years so after certain time, assigning fees will play a key role in order to compensate the miner's mining cost.

In this chapter, we will investigate the problem of determining the fee associated with a transaction. For that purpose, we will first define a specific model for clients, miners and the underlying communication system, namely the peer-to-peer network, considering the observations from Chapter 4. Then, we will use these models to analyze the role of the transactions fees and the way to choose them.

5.1 Notation

The notation that will be used along this chapter is the following:

- N is the total number of clients. Since a client cannot spent again the money until the transaction has been confirmed in the blockchain, N is also the number of transactions that are waiting to be included in a block. We will use i to refer to the i-th client or i-th transaction.

- M is the total number of miners. A miner is defined as an individual user or a pool of users who work together. We will use j to refer to the j-th miner.

- X refers to the maximum number of transaction that a block can fit in. On the other hand, x is a vector of all the miners' decisions where x_j denotes the amount of transactions that miner j decides to put in the block.

- α denotes the vector of hash power rate for all miners, where α_j is the value associated with miner j and $\Sigma_{j \in M} \alpha_j = 1$.

5.2 Proposed Models

In this section, we will define the model for the Bitcoin peer-to-peer network, miners and clients. Each of them has its own incentives and requirements which determine their actions and strategies.

5.2.1 Bitcoin Network Model

The Bitcoin network (Figure 2.2) is modeled as an undirected random graph G = (V, E) where each node has a fraction $0 \leq \alpha_v \leq 1$ of the total computational power of the network s.t. $\Sigma_v \ \alpha_v = 1$. The size of the network is of the order of $\sim 10^4$. Only those nodes whose $\alpha_v \neq 0$ are miners competing for being the first one to mine the next block. The rest are either clients (if they generate transactions) or nodes (if they relay blocks and transactions and store a copy of the ledger).

The generation of blocks follows a Poisson Process with mean $\lambda = \frac{1}{600 \ segons}$ since blocks are created at a rate of 1 every 10 minutes. Therefore, the interarrival time follows an exponential distribution with mean $\lambda = \frac{1}{600 \ segons}$.

After one block has been mined, it is propagated over the network until all nodes are aware of it. The transmission of this information is affected by the delay associated with each physical link and the verification time needed to validate each transaction in the block.

5.2.2 Miners' Model

Miners are nodes whose $\alpha_j \neq 0$. Let us consider a set of miners $\mathcal{M} = \{1, ..., M\}$ in the Bitcoin network where $M \geq 2$. Each miner j has a relative hash power α_j with respect to the total hash power of the network ($\Sigma_j\ \alpha_j = 1$) and miners compete against each other in order to be the first one to solve the proof-of-work. The miners' goal is to maximize their revenue, which comes from the fixed reward of successfully mining a block R plus the sum of all the fees f of all the transactions included in that block. However, the process of solving this puzzle is not free so there is an associated cost that is a function of the time t it takes to mine the block and the miner's hash power rate α_j. Consequently, the expected revenue u_j of each miner j is defined as:

$$u_j = (R + \sum_{k=1}^{x_j} f_k)\ P_j(\alpha_j) - cost(t, \alpha_j)$$

where $P_j(\alpha_j)$ is the probability that $miner_j$ mines the block and fees f are sorted in descendant order.

The process of successfully mining a block can be divided into two phases: the mining phase and the relay phase. In the first phase, the probability that miner j mines the block is directly proportional to his hash power rate α_j. The probability of success of the mining process is not only a function of the hardware, but also of the software implementation and the strategy. Therefore, α_j is defined as the mining capacity of a miner so owning α_j is equivalent to claiming that the miner j is successfully creating blocks with probability α_j.

On the other hand, in the relay phase the time it takes to transmit the block is a function of the size of the block [32] for any block bigger than 20kB. Therefore, the bigger the block the more time is needed to relay the data to the majority of nodes of the Bitcoin network so there is a small penalization to the probability obtained in the second phase. This procedure is shown in Figure 5.1. Even though miner 1 and miner 3 found a valid block at similar times, since the block candidate 1 is smaller, it has been propagated to the majority of nodes thus accepted by most of them before block candidate 3. As explained in Section 2.3.3,

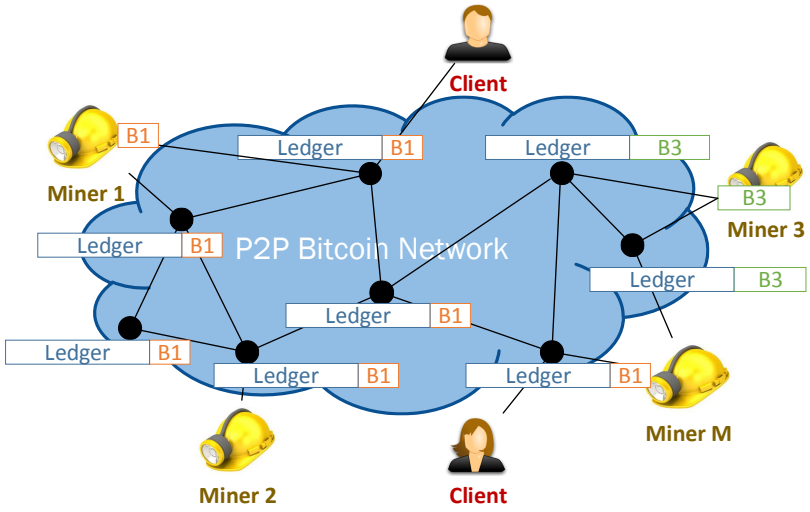

Figure 5.1: The figure shows the propagation of two valid blocks created at similar times whose size differs. The candidate block 1 is smaller so the time it took to propagate it to the majority of the nodes was smaller and therefore, it was accepted by consensus even thought the candidate block 3 was created more or less at the same time.

consensus consists of what the majority knows. This delay problem only applies to blocks because transactions too small to be affected (size $<$ 20Kb).

Delay penalization caused by size can be modeled as an exponential function with exponent $(1-\alpha_j)\lambda$, where λ is the inverse of the average time it takes to mine a block, namely 600 seconds and α_j is the hash rate power of miner j. This monotonically decreasing function assumes that each miner competes against the rest of the community (namely, $1-\alpha_j$) and that each miner is directly connected to α_j proportion of the network (the bigger the resources, the more connections to the network). Thus, the probability that miner j successfully mines a block is:

$$P_j = \alpha_j \cdot e^{-(1-\alpha_j)\lambda\, x_j}.$$

We consider that the cost of the mining rig has been already amortized so the the value of the cost is directly related to the time it takes to mine a block (in average, 10 minutes), the price of electricity in kWh and the hardware consumption h_j that each miner j utilizes

(which is a function of the hash power rate α_j).

$$cost_j = h_j(\alpha_j) \cdot T \cdot electricity.$$

Theorem 1. *The miners best strategy consists of maximizing its reward function, namely* $u_j(x,t)$.

Proof. First of all, we approximate the summation of fees to the product of a constant c_j times the total number of transactions that the miner includes in the block x_j.

$$u_j(x,t) = (R + c_j x_j)\ P(\alpha_j) - cost(t, \alpha_j).$$

Then, we find the extremums by equaling to 0 the first derivative:

$$\begin{aligned} \frac{\partial u_j}{\partial x_j} &= -(1-\alpha_j)\lambda\ (R + c_j x_j)\ \alpha_j\ e^{-(1-\alpha_j)\lambda\, x_j} + c_j\ \alpha_j\ e^{-(1-\alpha_j)\lambda\, x_j} \\ \frac{\partial u_j}{\partial x_j} &= 0 \Rightarrow x_j = \frac{1}{(1-\alpha_j)\lambda} - \frac{R}{c_j}. \end{aligned} \tag{5.1}$$

Finally, we show that this extremum is a maximum by computing the second partial derivative and verifying that at the point of interest, the value is negative.

$$\frac{\partial^2 u_j}{\partial x_j^2} = (1-\alpha_j)^2\lambda^2\ (R + c_j x_j)\ \alpha_j\ e^{-(1-\alpha_j)\lambda\, x_j} - 2c_j\ \alpha_j\lambda(1-\alpha_j)\ e^{-(1-\alpha_j)\lambda\, x_j}.$$

Then, we substitute x_j for the value found in (5.1):

$$\frac{\partial^2 u_j\left(\frac{1}{(1-\alpha_j)\lambda} - \frac{R}{c_j}\right)}{\partial x_j^2} \overset{?}{<} 0.$$

Since c_j is always a positive value, we verify that for any value of α_j and λ, the inequality holds and the computed x_j is a maximum:

$$\begin{aligned} 0 &\overset{?}{>} \left[(1-\alpha_j)^2\lambda^2\ \left(R + c_j\left(\frac{1}{(1-\alpha_j)\lambda} - \frac{R}{c_j}\right)\right)\ \alpha_j - 2c_j\ \alpha_j\lambda(1-\alpha_j)\right]\ e^{-(1-\alpha_j)\lambda\,\left(\frac{1}{(1-\alpha_j)\lambda} - \frac{R}{c_j}\right)} \\ 0 &\overset{?}{>} (1-\alpha_j)\lambda\ (R + (\frac{c_j}{(1-\alpha_j)\lambda} - R)) - 2c_j \\ 0 &> -c_j. \end{aligned}$$

□

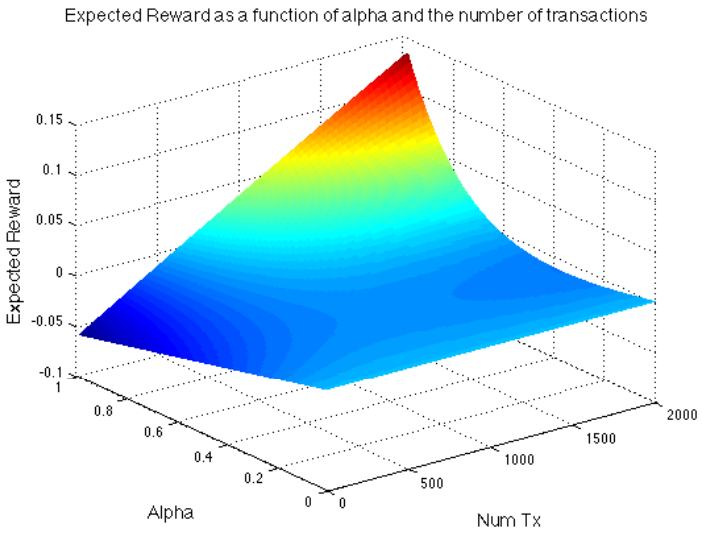

Figure 5.2: Miner's expected revenue as a function of α_j and the number of transactions. As α_j increases, the miner has more computational power so the number of transactions he can add increases too.

Given this model, we can plot the expected revenue as a function of miner's rate α_j and the number of transactions x_j added in a block as shown in Figure 5.2. As α_j increases, the number of transactions that the miner should add increases too because the gain in mining power compensates the delay posed by the transmission of a bigger block.

5.2.3 Clients' Model

Let us consider a set of clients $\mathcal{N} = \{1, ..., N\}$ on the Bitcoin network where $N \geq X$. That implies that the number of transactions that miners will include in a block will be less than the total number of unconfirmed transactions available, which agrees with the observed data [20]. Each client i creates transactions characterized by the total input value, the size and the fee f_i (which are public), plus a limited budget b_i that he is willing to pay according to his desire p_i (or urgency) to be added in the next block (which are private). Moreover, w_i represents the actual number of blocks created before transaction i is successfully included in a block.

Clients want to minimize the fee they should put in their transaction. Therefore, they want to minimize the cost function $t_i(f_i)$:

$$t_i(f_i) = \frac{b_i}{f_i} + w_i \cdot p_i. \tag{5.2}$$

Function $t_i(f_i)$ considers the two parameters that a client is interested in: the time until his transaction is confirmed (without loss of generality, here we assume that transactions are confirmed after the block they are in is accepted) and the fee they paid. Both parameters need to be minimized but there is a trade-off between what the client pays and the time until he receives the confirmation. On the other hand, the budget is the maximum fee that the client wants to pay (it may depend on the input money involved, the clients' purchasing power and so on).

The fee should consider the size of the transaction, because the amount of transactions that can be included in a block depends only on the size (1MB in total, at the time of writing). From this point on, when referring to a fee, we will consider the value per $\simeq 0.6$ KB (average transaction size observed).

Transactions fees are defined as the donation made by the sender or senders of a transaction to the miners of the Bitcoin network. At the time of writing, there is no minimum or standardized value associated with this fee but the current version of the Bitcoin Core software (0.9.3) implements a default fee based on some properties. The last modification to the present transaction fee behavior was introduced in version 0.9.0 [8]. From this release, the required fee to relay transactions across the network and for miners to consider the transaction in their blocks is 0.01 mBTC per kilobyte. However, this rule can be ignored if the transaction is less than 1 Kb long, all the ouputs are 0.01 BTC or larger (in order to discourage spam) and the value of the priority is over 57,600,000 [18]. On the other hand, those are guidelines implemented in the official Bitcoin Core software. That may not refrain other users from designing its own criteria and following these rules does not mean that the transaction will be included in a block by the miners either.

5.3 Community Point of View Analysis

The Bitcoin design relies on incentivizing the mining process by rewarding the miner who successfully creates a block. At the time of writing, there is a fixed reward of 25 BTC per block so the fees added by the users are quite negligible (around $1.64 \cdot 10^{-4}$ BTC per transaction in average). However, since the fixed reward is halved every 200,000 blocks, at some point both values will be of similar order and fees will play a key role when incentivizing the process of mining. On the other hand, as pointed out by Gavin Andersen [1], miners only include transactions that at least cover the cost of processing them. However, based on the observations we can claim that miners also include zero-fee transactions so there is some charity going on that can threaten the viability of the Bitcoin system. Those two facts bring up the following question: Can we determine the best fee a client should put in its transaction that is both fair and competitive against other clients, while still incentivizing the mining process?

In this section, we will analyze transaction fees from the whole Bitcoin community point of view in order to find an equilibrium between what clients should pay and what miners should receive in exchange so the system remains stable. As such, the goal is to find some policies that protect the viability of the system.

5.3.1 Problem formulation

Transaction fees are money donated by clients to miners in order to reward the process of validating transactions and creating blocks. Consequently, since both parties try to maximize the revenue, the Bitcoin community acts as a mediator in this situation.

From the point of view of the clients, generating a transaction and relaying it across the network does not guarantee that the transaction will be accepted by a miner and included in his block; by default, miners fill their blocks with 50 kilobytes of high-priority transactions, and then with 700 kilobytes of the highest-fee-per-kilobyte transactions so their decision is conditioned to what the rest of the clients are doing.

On the other hand, from the point of view of the miners, the main goal is to have profits. However, the exchange from Bitcoins to physical currencies (e.g. dollars) is quite volatile and parameters such as transaction fees don't consider this fluctuation. Since the cost of mining is not paid in bitcoins (electricity bill, hardware rig) setting a fixed default fee as from version 0.9.0 of the reference implementation, might not be a good idea.

The Bitcoin community consists of 3 participants (miners, nodes and clients) with different requirements and motivations. In order to answer the question of how users should determine fees and when miners should accept them, we make the following assumptions:

- The Bitcoin system was developed as an anonymously distributed alternative to traditional coins so at this point, the join goal of the Bitcoin community is to popularize it and extend its use.

- Miners seek the maximum economical benefit when creating new blocks. They behave honestly and don't engage in attacks to other pools such as DoS; all of them follow the same strategy and add the same transactions to the block they try to mine, that is the k transactions with the highest fees (consequently, the probability of wining is only a function of their hardware); and they don't collude among them when choosing transactions.

- Nodes are non-profit devices that act independently of clients and miners and follow the consensual rules defined by the Bitcoin community.

- Clients try to minimize the fee they pay but at the same time, they are interested in following the defined rules so their transaction is validated and included in a block.

5.3.2 Proposed Method

Let's design a competitive strategy that any miner should follow when choosing transactions in order to maximize his overall profit.

We consider a single-round, sealed-bid auction $\mathcal{A}$ for a limited number of units as in [38].

1. N Bitcoin clients who created a transaction between the generation of the previous block and the current time.

2. A vector f of the submitted fees. The i-th component f_i is the fee defined by client i.

3. An output vector $x = (x_1, x_2..., x_N)$ whose value is 1 for each transaction included in the block and 0 for the rejected ones.

4. A limited maximum number of supplies X equivalent to the maximum number of transactions allowed in a block.

5. The profit of the network is
$$\mathcal{A}(f) = R + \sum_i f_i.$$

A naïve approach is to define a rule such that only the X transactions with the highest fees are selected. However, then the bidders' best strategy may not be the truth, so the overall profit could be less than expected. In [38], deterministic symmetric auctions were proved to be non-competitive. Consequently, we propose the use of a randomized solution such as the following adaptations of Random Sampling Optimal Price Auction (RSOP) and Random Sampling Profit Extraction Auction (RSPE) [38].

Bitcoin Random Sampling Optimal Price Auction (Bitcoin RSOP)

RSOP is a randomized auction that maximizes the auctioneer profit by incentivizing bidders to bid the real amount that they are willing to pay. This algorithm uses the concept of *optimal single sale price*[38].

Definition 1. *Given a vector b of bids, the optimal single sale price is defined as:*

$$opt(b) = argmax_{z_i} iz_i$$

where z_i is the i-th largest bid in b.

Then, from RSOP [38], we define a variation called Bitcoin RSOP that we summarize in Algorithm 2.

Algorithm 2 Bitcoin RSOP

Input: Fees f

Output: Accepted transactions

1: Sort all the fees in descendant order
2: Partition fees f uniformly at random into 2 sets, f' and f''
3: Compute the optimal single sale prices, p' and p'', for each set: $p' = opt(f')$ and $p'' = opt(f'')$
4: Accept transactions in f' that are equal or bigger to p''
5: Accept transactions in f'' that are equal or bigger to p'

This algorithm sorts all the candidate transaction fees in descendant order, partitions them at random into 2 sets and computes the optimal price for each set. Then, based on these values, the miner adds in his block all transactions from the first set that have fees greater that the optimal price of the second set and all transactions from the second set that have fees greater that the optimal price of the first set. If the size of the block surpasses the allowed size, transactions with the lowest fees are discarded until the size matches the required dimension.

p' and p'' define the minimum fee that miners should accept. This policy can work with the existing design as long as it is only applied to the 700Kb that don't consider the priority of the transaction. This way, it solves both the problem of charitable miners (it is fair that clients pay when generating transactions and miners should not accept those ones with low fees) and the problem of defining a minimum fee that is equitable for everyone.

Bitcoin Random Sampling Profit Extraction Auction (Bitcoin RSPE)

Similar to RSOP, there is another type of randomized auction that is guaranteed to achieve a profit very close to the optimal one: RSPE. It uses the concept of *Optimal Single Price*

Omniscient Auction [38], defined as follows.

Definition 2. *Given a vector b of bids and let z_i be the largest bid in b, then the optimal single sale price $\mathcal{F}$ determines the value k such that kz_k is maximized. All bidders with $b_i \geq z_k$ win at price z_k while the rest lose and the profit is:*

$$\mathcal{F} = max_{1 \leq i \leq N} i z_i.$$

Now, from RSPE [38], we define a variation called Bitcoin RSPE that we summarize in Algorithm 3.

Algorithm 3 Bitcoin RSPE

Input: Fees f

Output: Accepted Transactions

1: Sort all the fees in descendant order
2: Partition fees b uniformly at random into 2 sets, b' and b''
3: Compute the optimal single profit for b' and b'': $F' = F(b')$ and $F'' = F(b'')$
4: **for each** set **do**
5: Find the largest k such that the highest k fees are at least F/k
6: Charge the k transactions with the higher k fees with R/k and reject all others
7: **end for**

This algorithm sorts all the candidate transaction fees in descendant order, partitions them at random into 2 sets and computes the optimal single profit for each set. Then, it finds the largest number of transactions whose fees (which are sorted in descendant order) are at least the *optimal single profit* divided by this number, for both sets. Again, if the size of the block surpasses the allowed size, transactions with the lowest fees are discarded until the size matches the required dimension.

As before, this method allows the network to set up a fair minimum fee for the transactions that will be included in a block. Since fees change over time and depend on the clients, each block will have a different minimum fee.

In both cases, both methods incentivize clients to tell the truth and bet the maximum value that they are willing to pay, while at the same time, miners maximize their profit. Following these approaches, all users end up paying the same, despite the amount of money involved in the transaction.

5.4 Clients Point of View Analysis

In the previous scenario, we considered a single strategy for all the miners dictated by the Bitcoin community. However, miners are independent entities and engage in a competition where only one of them can win. In this section, we will analyze how clients can determine the value of the fee according to the model defined in Section 5.2.3 and considering that miners follow the model explained in Section 5.2.2.

5.4.1 Problem formulation

Miners and clients compete all against each other motivated by the economical reward. Again, this incentive determines their decisions which, in general, differ. On one hand, miner A may decide to put a different number of transactions than miner B while transaction C has a bigger fee than transaction D, even though the size and the inputs are similar. Diversity is, in general, good, but if we don't have a Bitcoin community that defines some rules, which strategy should clients select?

We make the following assumptions:

- Each miner i is applying his own best strategy, which is a function of his hash power rate α_i, to obtain the number of transactions he should add in the block.

- A miner decides which transactions wants to add in a block based on the values of the fee, the size and the priority (Eq. (2.1)) of the transaction. Consequently, the client's transaction will appear in the next block (and thus, will be confirmed in the blockchain) if the winner of the proof-of-work challenge decides to put it in his block.

- The distribution of fees and the distribution of the number of transactions per block are known.

- Mining is an independent event over time so finding a block at one point does not give any advantage in future mining competitions.

N clients are competing for a limited number of units X. Those units are the number of available spots in a block, whose size is limited. The clients' scenario is defined as follows:

1. N Bitcoin clients who created a transaction between the mining of the previous block and the current time t.

2. A vector $p = (p_1, p_2..., p_N)$ which represents the client's willingness of being added in the next block (i.e. urgency). This value p is translated into the probability that his transaction will be in the next block.

3. A vector $b = (b_1, b_2..., b_N)$ of budget limits (i.e. 1% of the total inputs) that reflects the maximum amount that the client is willing to pay. The i-th component b_i is the fee defined by client i.

4. A limited number of supplies X equivalent to the number of transactions that fits in the block.

5. A fixed revenue R per block that is determined by the protocol.

Given a budget b and a desired probability p of being added in the next block, a client wants to compute the fee he should add to the transaction. For that, the client must find the minimum fee that satisfies the required probability p:

Definition 1. *Let* $0 < p \leq 1$ *and* $b > 0$. *The client's fee can be computed as:*

$$min\ fee,\ where\ fee \leq b$$

$$s.t.\ \{p \leq P(client_i's\ transaction\ is\ in\ the\ next\ block)\}.$$

If this condition cannot be satisfied, the fee will be equal to the client's budget.

5.4.2 Proposed Method

The considered scenario consists of N Bitcoin clients competing to be one of the up to X number of transactions that a block can fit in. The decision of which transactions to add is made individually by each miner and it is a function of his own hashing power rate α and the transaction fees. There are M miners trying to mine a block but only one will win so only the transactions he decided to add will be in the next block. Miners goal is to maximize their utility function so their strategy consists of maximizing their expected reward.

Given a fee f_i and the number of participants N, the probability that a client's transaction is in the next block (from now on, we will refer to it as the i-th transaction) can be mathematically expressed as:

$$P_i(f_i) = \sum_{j \in M} Q_j(\alpha_j, f_i)\ R_j(\alpha_j).$$

- $P_i(f_i)$: Probability that $client_i$'s transaction is in the block.
- $Q_j(\alpha_j, f_i)$: Probability that $client_i$'s transaction is in $miner_j$'s block.
- $R_j(\alpha_j)$: Probability that $miner_j$ mines the block.

As explained before, the probability that a miner mines a block is equal to his hash rate power α_i and it can be derived straightforward from the exponential distribution. On the other hand, the computation of the probability that $transaction_i$ is in the next block is more tricky. First of all, it is a function of number of transactions picked up by the miners x, the fee f_i of the i-th client and the total number of participants N. The solution of this problem is summarized in Algorithm 4. Let's define F as a random variable that characterizes the fee distribution and X_j as a random variable described by the distribution of the number of transactions included in a block by miner j. Initially, we set the suggested fee f equal to the budget b_i. Then, we need to find the expected position of $client_i$ among every transaction fee, when they are sorted in descendant order. After that, we need to compute the probability that the number of transactions picked up by the j-th miner is

bigger or equal to the expected position. Finally, the last step consists of comparing the obtained probability with the desired probability p_i. If the value is greater or equal to it, we decrease the suggested fee f and run the algorithm again. If not, we set the fee to the previous suggested fee.

Algorithm 4 Probability that $transaction_i$ is in $miner_j$'s block

Input: b_i, p_i, N, distribution F, distribution X_j, α_j

Output: fee

Initialization: $f = b_i$, $fee = b_i$
while $f \geq 0$ **do**
    $E[position] = P(F \geq f) \times N$
    $q = \sum_{j=1}^{M} P(X_j \geq E[position]) \cdot P(\alpha_j)$
    if $q \geq p_i$ **then**
        break;
    else
        $fee = f$
        decrease f
    end if
end while

5.5 Simulations

In this section, simulations following the previous method are provided. First of all, we will consider the scenario where each user decides his own fee based on a desired probability q. In this case, R is set to a value of similar order to the average fee ($\sim 1.9 \cdot 10^{-4}$), the number of clients is known and the amount of transactions that each miner will include is fixed. Moreover, we consider both the case where the distribution of fees is exponential and Gaussian (as observed in Chapter 4). Results are shown in Figure 5.3 with a plot that reflects the relationship between the desired probability q versus the value that the client

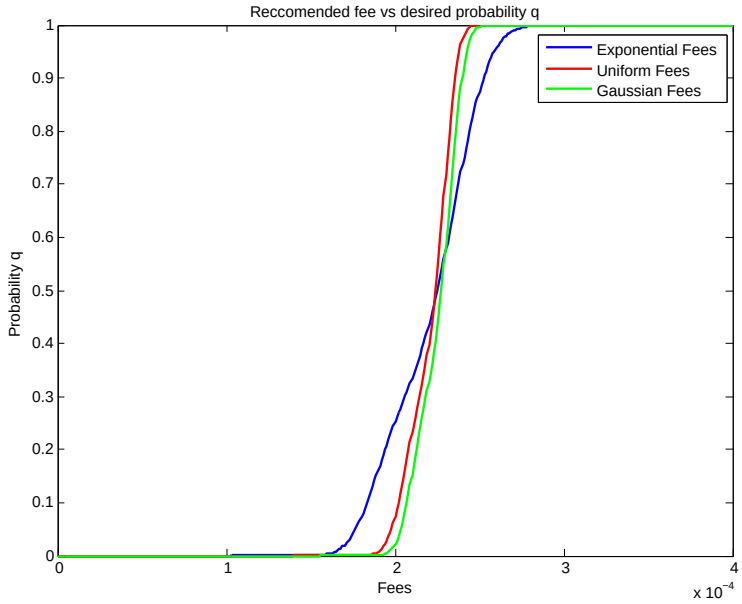

Figure 5.3: Recommended fee as a function of the desired probability q to be included in the next block. As this q increases, the fee needs to compensate it so it increases too.

should add as a fee, given that the rest of the clients are choosing the fees according to the observed distribution of fees (Figure 4.2).

The main restriction of the previous results is that we are considering that each client chooses q according to the distribution of fees. Consequently, if all clients start to apply this strategy, the results may not be valid anymore. That is why we also considered the case where the distribution of the fees is random. This way, if the desired probability follows a random distribution, we can see that the simulation results still apply.

5.6 Discussion

In this section, we considered the clients' selection of fees and we analyzed both the case where the network enforces a fair value and also when miners compete against each other in a selfish way. For both cases, at the time of writing there is a fixed revenue of 25 BTC that makes fees insignificant. However, if clients continue disregarding fees and the community

keeps accepting transactions with low fees, it poses a threat to the viability and the survival of the currency.

The main problem of the first approach is that fees should be modified after been issued. This is not contemplated in the current implementation but it could be add in the future. The main advantage of it is that clients are incentivized to tell the truth and miners will maximize their profit. On the other hand, this solution does not work in a colluding environment.

In the second approach, we considered the case where the relay phase plays an important role in the success of mining. Thus, miners want to choose the optimal amount of transactions to put in a block in such a way that their probability of success is not highly penalized by the size of the block. However, there are some ways to overcome this problem. The simplest one consists of creating a full node that is connected to the majority of the network and to the miner. That way, the transmission delay becomes negligible even though this solution goes against the decentralized principle that the creator of Bitcoin was looking for [56].

Another way of analyzing the Bitcoin model is from the game theoretical perspective. However, a more precise model [39] was proved to be non-concave so an equilibrium could not be found. In that case, the system could be analyzed in terms of probabilities of success but the best strategy will be to add as least transactions as possible (1 transaction). This statement does not match the observations presented in Chapter 4, probably because miners are also interested in the survival of the system.

Chapter 6

SELECTION OF TRANSACTIONS

In Chapter 5, we showed that clients try to minimize the fee they give away to miners. However, since the bigger the block, the lower the probability of successfully mining a block, if the donation doesn't compensate this loss, miners might be tempted to create blocks with the minimum amount of transactions. In this section, we will present the miners' individual mining model considering the internal process of generating a block. Then, we will propose a modification to improve their profitability and we will discuss why miners aren't implementing it.

6.1 Preliminaries

First of all, let's introduce some mathematical background on Discrete Time Markov Chains (DTMC).

Let's consider a discrete-time stochastic process $\{X_n\}$ over a countable state space $\mathcal{S}$. Then, X_n is called a Discrete Markov Chain (DTMC) if for all $n \in \{0, 1, 2, \cdots\}$ and $j, i, i_0, \ldots, i_{n-1} \in \mathcal{S}$:

$$P(X_{n+1} = j | X_n = i, X_{n-1}) = i_{n-1}, \ldots, X_0 = i_0) = P(X_{n+1} = j | X_n = i).$$

So state X_{n+1} depends only on X_n and it is independent of all the past states $X_{n-1}, \ldots, X_0$.

We refer to P_{ij} as the transition probability from state i to state j and the transition probability matrix P as:

$$P = \begin{bmatrix} P_{00} & P_{01} & \ldots \\ P_{10} & P_{11} & \ldots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

A stationary distribution $\boldsymbol{\pi}$ exists if the following two conditions are met:

$$\boldsymbol{\pi} P = \boldsymbol{\pi}. \tag{6.1}$$

$$\sum_i \pi_i = 1. \tag{6.2}$$

If $\boldsymbol{\pi}$ exists, the state distribution converges to a steady state stationary distribution regardless of the initial distribution.

State $j \in \mathcal{S}$ is said to be absorbing if $P_{jj} = 1$.

6.2 Observations

In this section, we will highlight specific observations for developing the mining model obtained by analyzing the Bitcoin ledger:

- The number of transactions included in a block increases as the time to solve the puzzle increases.

- The number of valid and available unconfirmed transactions is always bigger than the maximum number of transactions that can be (and are) fitted in a block.

- Transactions in the blocks are not sorted in descendant order according to the fees.

- The average size of a block (around 0.3 MB) is way below the maximum block size (1MB). Hence, miners are not selecting as many transactions as possible.

- There is also a small amount of empty blocks which don't contain any transaction and whose generation time is close to 0.

- Individual miners are organized in pools in order to increase the overall hashing power. That is because they prefer having a regular income rather than playing the lottery.

In terms of pools, mining pools are made up of an unknown number of miners managed by the pool supervisor. In some cases, the supervisor can be in charge of verifying transactions and computing the Merkle Tree hash so the miners only need to focus on trying different nonces. Thus, the miners job becomes even more specialized and the speed is also increased. Miners inform the pool manager of every block they create whose hash is below a threshold (which is bigger than the current difficulty). Hence, the manager can compute the contribution of each miner when it is time to distribute the benefits of the pool.

We refer the reader to Chapter 4 for specific results about fees and sizes, among other parameters, which were obtained from data collected from August 2014 to February 2015.

6.3 Proposed model

In the Bitcoin environment, we define miners as entities whose goal is to produce valid blocks. Apart from supporting the system, miners are also motivated by the monetary revenue they obtain for successfully mining blocks. They also want Bitcoin to continue existing so their source of money never ends. In this section, we will take into account these motivations plus the parameters that characterize the mining process (i.e. hash power, transactions per block, fees, cost).

Let's start defining a Discrete Time Markov Chain (DTMC) as shown in Figure 6.1 for each miner. Without loss of generality we assume that time is measured in units equal to the time it takes to compute the block hash for 2^{32} possible nonces with a fixed Merkle Root hash. That is equal to $\beta \cdot 2^{32}$ hash computations, where $1 < \beta \leq 2$ depending on how optimized the implementation of the proof-of-work is [31]. A step is defined to be the movement from one state and the following one so between state 1 and state 3, there are two steps or time steps. Assume that at k-th state n_k transactions are added to the Merkle Root computation (therefore, they are added to the block) and with probability p, this miner won't find the solution. On the other hand, with probability $(1-p)$ he will solve the proof-of-work challenge and will end the process (again, we are not considering what happens if another miner finds the solution before).

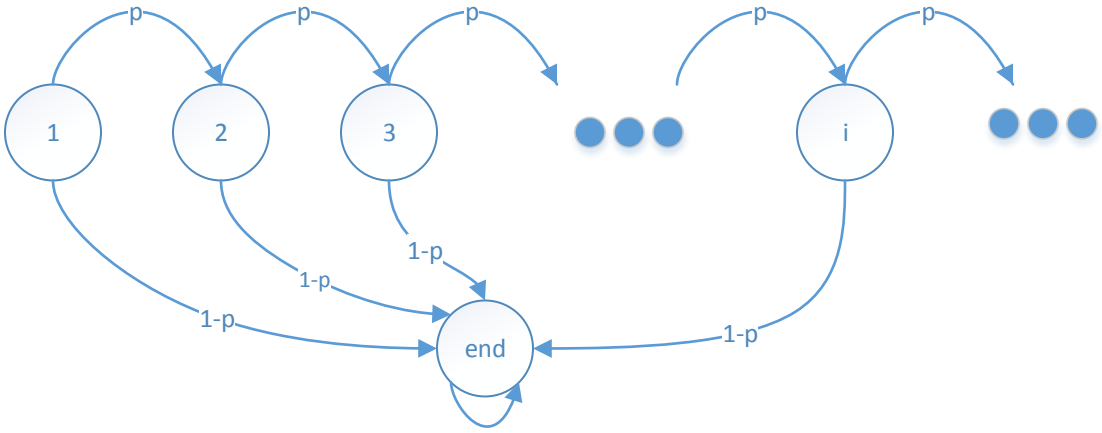

Figure 6.1: Discrete Time Markov Chain (DTMC) that describes the process of solving the proof-of-work puzzle. Time is measured in units equivalent to the computation of the block hash for 2^{32} different nonces, given a fix number and order of transactions (fixed value of the Merkle Root hash). Given any state k, the challenge is solved with probability $1 - p$. Otherwise, with probability p the miner changes the value of the Merkle Root hash and tries again to solve the puzzle with the 2^{32} possible nonces.

From this point, we can compute the probability Q_k of being in the k-th state by equaling the inputs of a given state to its outputs:

$$\begin{aligned} Q_2 &= Q_1 \cdot p \\ Q_3 &= Q_2 \cdot p = Q_1 \cdot p^2 \\ Q_4 &= Q_3 \cdot p = Q_1 \cdot p^3 \\ &\dots \end{aligned}.$$

Therefore, the probability Q_k for $\forall k > 0$ can be generalized to:

$$Q_k = Q_1 \cdot p^{k-1}.$$

With the previous information, we can also obtain the transition probability matrix:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ 1-p & 0 & p & 0 & 0 & \dots \\ 1-p & 0 & 0 & p & 0 & \dots \\ 1-p & 0 & 0 & 0 & \ddots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Now, we can compute the stationary distribution π by solving the system of equations obtained from conditions (6.1) and (6.2):

$$[\pi_0\pi_1\dots] \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \dots \\ 1-p & -1 & p & 0 & 0 & \dots \\ 1-p & 0 & -1 & p & 0 & \dots \\ 1-p & 0 & 0 & -1 & \ddots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = [0\ 0\ \dots\ 1] \tag{6.3}$$

$$\boldsymbol{\pi} = [\pi_0\pi_1\dots] = [1\ 0\ 0\ \dots].$$

Therefore, for $0 \leq p < 1$ this model converges to a stationary distribution regardless of the initial distribution which means that each miner will eventually solve the proof-of-work puzzle and reach the end state. If p was equal to 1, then the Markov Chain would keep transitioning to the next state forever.

Let's define w as the state where the miner solves the proof-of-work puzzle (the state before arriving to the end state). Then, we can define T_k for a given state k as the probability that $k \leq w$ or, in other words, as the probability that state k is reached during the mining process. Knowing that state 1 will be reached with probability one, the probability of reaching state k is defined as:

$$T_k = T_1 \cdot p^{k-1} = p^{k-1}.$$

Now, we prove that the end state will be also certainly reached under these circumstances

(we are not considering externalities like the case where another miner solves the block):

$$P(\text{find a valid block}) = P(T_{end}) = \sum_{k=1}^{\infty} T_k(1-p) = T_1(1-p)\sum_{k=1}^{\infty} p^{k-1} = T_1 = 1.$$

Now, we can define the expected reward $E_j[Revenue]$ of the mining process for each individual miner or pool j. As in Chapter 5, for the analysis we approximate the summation of the highest fees with a constant c_j times the total number x_j of included transactions. The reward consists of two terms, namely the expected benefit and the expected cost. The former is equal to the sum of all the revenues (the fixed block reward R plus all the transaction fees f) conditioned to the probability α_j that the j-th miner wins. On the other, the latter describes the cost of mining which is a function of the cost of mining per second and the total time t that the miner invested in trying to solve the proof-of-work puzzle.

$$\begin{aligned} E_j[Revenue] &= benefit - cost \\ &= (R + \Sigma_i f_i)\ \alpha_j - cost \\ &= (R + cx_j)\ \alpha_j - cost. \end{aligned} \tag{6.4}$$

In the remaining of the section, we will define the parameters involved in Eq. (6.4).

Number of transactions

The number of transactions added in the Merkle Root hash after checking all the possible nonces is observed to be monotonically increasing with mean μ. That is consistent with the best strategy of adding the least number of transactions in a block so the relaying delay does not critically endanger the miner's probability of mining a block. Consequently, the miner's strategy for computing the Merkle Root is to first sort all transactions based on the fee/KB and then, start trying with one transaction, then some more and so on. We define this increasing function $g(k)$ at the k-th state as:

$$n_k = g(k) = X(1 - e^{-\gamma k}). \tag{6.5}$$

The expected number of transactions can be computed as the number of transactions associated with each state times the probability that the valid block is found during that

state:

$$E[\text{number of transactions}] = \sum_{k=1}^{\infty} P(K=k) \cdot n_k = \sum_{k=1}^{\infty} (1-p)^{k-1} p \cdot n_k. \tag{6.6}$$

Cost of mining

The cost of mining is a function of the time it takes to mine a block and the resources that a miner has. Then, given that miner j spent t seconds mining, the total hardware consumption is h and the price of the electricity is e, the cost per block can be computed as:

$$\text{cost per block} = t \cdot h \cdot e. \tag{6.7}$$

For every step of the Markov Chain, miner j is paying:

$$\text{cost per step} = (2^{32}\ \beta\ + 1)\ Hash \cdot h \frac{kWh}{GHash} \cdot e \frac{\$}{kWh}.$$

The expected mining cost can be computed as the cost per step times the probability that the valid block is found in that step:

$$E[\text{mining cost}] = \sum_{k=1}^{\infty} P(K=k) \cdot k \cdot cost = \sum_{k=1}^{\infty} (1-p)^{k-1} p \cdot k \cdot cost.$$

Since the average mining time is 10 minutes, the expected mining cost can also be computed as $600s \cdot cost/s$.

Adding up, the expected reward first defined in Eq. (6.4) is:

$$\begin{aligned} E_j[Revenue] &= (R + cx_j)\ \alpha_j - cost \\ &= \Big(R + c \sum_{k=1}^{\infty} (1-p)^{k-1} p \cdot n_k\Big) \alpha_j - \sum_{k=1}^{\infty} (1-p)^{k-1} p \cdot k \cdot cost. \end{aligned} \tag{6.8}$$

6.4 Analysis

After presenting the mining model, in this section we will use real values to analyze the impact of the model in the actual Bitcoin scenario.

Until this point, the number of hash computations in each time step has been noted to be $\beta \cdot 2^{32} + 1$. From now on, we will approximate this value to 2^{33} considering an average

implementation of $\beta = 2$. This must not be confused with the number of possible nonces, namely 2^{32}, for a given Merkle Root hash. Now, we can compute the probability of not solving the proof-of-work, $0 \leq p < 1$, by looking at the real difficulty of the mining process. As pointed out in Section 2.3.2, this value is biweekly updated so it matches the desired difficulty of the system. As of March 11th, 2015:

$$target^1 \simeq 2^{256-67}. \tag{6.9}$$

Equation (6.9) states that the hash of the block's header needs to have at least 67 leading zeros and that the probability that a computed block hash is less than the target, given a random data input, is approximately 2^{-67}. Therefore, the probability that one nonce does not give a valid block is $(1 - 2^{-67})$. Based on that, we can compute the probability that the miner is not successful in one step of the DTMC, namely none of the 2^{32} trials (2^{32} nonces) give a valid answer:

$$\begin{aligned} P(no\ success\ in\ 2^{32}\ trials) &= (1 - 2^{-67})^{2^{32}} \\ &= \left(\left(1 - \frac{1}{2^{-67}}\right)^{2^{-67}}\right)^{2^{32}} \\ &\simeq e^{-2^{-35}} \\ &\simeq 1 - 2^{-35}. \end{aligned}$$

After that, we can define the value of the number of transactions included in a block. The function $g(x)$ in Eq. (6.5) converges to the maximum number of transactions allowed in a block and the value of γ is adjusted so the number of transactions after 2^{67} guesses ($\frac{2^{67}}{2^{32}} = 2^{35}$ time steps in the Markov Chain) is equal to the mean $\mu = 547$ number of transactions per block observed in the ledger[2]. This value $\gamma = 1.16 \cdot 10^{-11}$ is computed as $g(2^{34}) = 547$ for a maximum number of blocks X equal to 1660 (Eq. (4.1)). We claim that the way miners achieve $g(x)$ is by using the nonce field. Miners are allowed to modify the corresponding 96 bytes available in the coinbase transaction and add any information there. For example,

[1]As of March 11th, 2015, target = 6843611944711461888388750121126858921758200033619581337 6

[2]Data from August 2014 to February 2015.

Pool	Block height	Message
AntPool	352551	'aMined by AntPool sc0��� U1ӐŔ "
BTC Guild	352550	&a#ÿMined by BTC Guild,��mm��NF$G���A"K>�<���w$T8e&e$K
Slush	352556	,a/P2SH/��1U@X24df74c/slush/

Table 6.1: Examples of the information contained in the coinbase transaction.

in the Genesis Block (the first mined block) it says The Times 03/Jan/2009 Chancellor on brink of second bailout for banks. Pools usually add information about them like examples in Table 6.1.

As presented before, the miner's probability of success for a specific number of time steps follows a geometric distribution whose mean is equal to $\frac{1}{p} = 2^{35}$:

$$P(K = k) = (1 - p)^{k-1}p.$$

Consequently, in average a miner will have to compute 2^{67} trials ($\frac{2^{67}}{2^{32}} = 2^{35}$ time steps of the Markov Chain) before solving the proof-of-work puzzle.

In terms of cost, for example let's consider a product such as *CoinCraft A1 ASIC* with up to 40 GHash/s and power usage of 1 W/GHash in Turbo mode [29]. As of March 2015, the price for 1kWh in Seattle was around $0.12. [49]. Thus, for every unit of time (previously defined as $1 + \beta \cdot 2^{32}$) cost would be:

$$\text{cost per step} = 2^{33} Hashes \cdot \frac{1W}{GHash/s} \cdot \frac{\$0.12}{1kWh} = \$ \cdot \frac{8}{3} \cdot 10^{-7}.$$

The current values of the parameters are summarized in Table 6.2.

Right now, miners are selecting the least amount of transactions that they can. However, they could improve their chances of winning if the function of the amount of transactions $g(x)$ increased slower. We propose that instead of adding transactions to the Merkle Root hash directly, miners sort the order of the chosen transactions and recompute the Merkle Root hash accordingly. This way, $g(x)$ grows slower as in shown in Figure 6.2.

Parameter	Description	Value
p	probability of success in the k-th time step	$1 - 2^{-35}$
1-p	probability of failure in the k-th time step	2^{-35}
a	number of candidate blocks created in each time step	2^{32}
b	number of hashes performed per block candidate	$\beta \simeq 2$
c	number of hashes performed in a time step	$\beta \cdot 2^{32} + 1 \simeq 2^{33}$
d	average number of time steps needed to solve the proof-of-work puzzle	2^{35}

Table 6.2: Summary of the actual values of the model

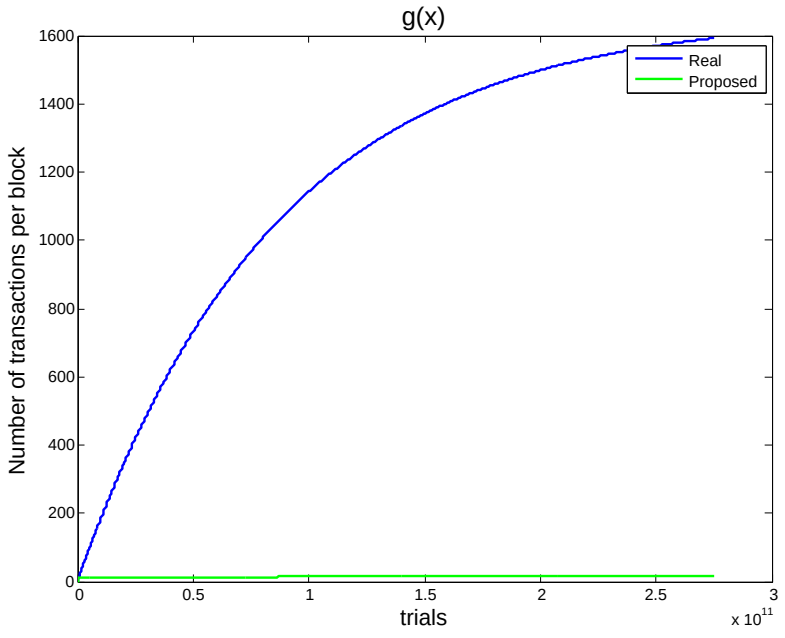

Figure 6.2: Number of transactions per block as a function of the number of trials, namely the candidate blocks computed before that did not achieve the expected target.

For this proposal, miners can be the ones who compute the Merkle Root hash or contribute in a pool where the supervisor computes it for them and they only have to query him.

6.5 Discussion

In this chapter, we considered the internals of the mining process in order to define a model that reflects the strategy of miners when selecting transactions. Therefore, the inclusion of the procedure for finding the block's header hash and the restriction that miners may not want to add lots of transactions because big blocks are slower to relay leads us to a trade-off: miners want the Bitcoin system to survive but at the same time are incentivized by the reward they are getting at the current time. Since the fixed reward is halved around once every 4 years, this behavior is expected to change as time goes by and fees will play a more important role in the future.

Until this point, we have utilized real data to compute averages that reflect the mean time and mean cost of mining, as a function of the hardware rig and software implementation. However, the system is expected to expand and while now it is only creating slightly less than 3 transactions per second, it could, at some point, significantly increase this value. Visa is serving around 2000 transactions per second, which looks like a more realistic number if we consider the world's population. If that was the case, there would be two main options for Bitcoin: increase the size per block or increase the block generation rate.

For the first case, increasing the size of the block would allow the miners to add more transactions to it. However, since the bigger the block, the more time it takes to be relayed to the entire network, this change should be accompanied by an policy that enforces a minimum amount of transactions per block. Rules are not popular in the Bitcoin community, but the increase in size is also seen as something unavoidable for the survival of the electronic coin.

On the other hand, the second option would be to increase the block generation rate. This method could jeopardize the effectiveness of the proof-of-work, specially if the increment is proportional to the arrival rate. For example, in order to maintain an equivalent equilibrium, for 2000 transactions per second the system should create a new block more or less every

second, which is not compatible with the actual design of the system.

To conclude, miners are minimizing the number of transactions that they include in a block but they are not optimizing this process. The rate they add new transactions to the Merkle Root Tree once they have exhausted all the possible nonces could be heavily minimized by sorting the already included transactions (that way, the Merkle Root hash varies). Observations back this conclusion because the selection of the amount of transactions per block does not take advantage of this more profitable solution. However, this is not a recommendable behavior as it could endanger the survival of the money.

Chapter 7

CONCLUSIONS

In this thesis, we analyzed both the determination of fees and the impact of fees over the miners' strategy for selecting transactions to include in a block. Firstly, we observed that miners do not consider clients' remunerations because these fee values are considerably low compared to the fixed reward. Thus, miners are only incentivized to mine blocks as quickly as possible to collect the fixed associated prize. Then, based on the proposed models, which include the penalization of relaying big blocks to the network, we conclude that the current values of the fees cannot compensate the risk derived of the increase in the block size. Finally, miners are not optimizing the computation of the proof-of-work, but instead they are considering a 'social conscience' that compels them to add an average of near 500 transactions per block. This is far from the maximum number of transactions per block, but contributes to maintain the queue of unconfirmed transactions constant-length. While this behavior is not a problem with the current transactions arrival rate, the system is not prepared to handle transaction volumes similar to Visa, who manages around 2000 transactions per second.

However, some questions remain widely open. It is unclear how the community is going to solve the upcoming challenge of decreased fixed revenue and the resulting limitation of the coins in circulation. Moreover, we will investigate the problem of colluding miners operating in different pools. Another line of research will be the analysis of the benefit of running a malicious node connected only to the targeted pool which relays only information about it. Bitcoin is a rising topic in our society and much compelling research work is expected to appear in the following years.

BIBLIOGRAPHY

[1] Gavin Andresen. Back-of-the-envelope calculations for marginal cost of transactions. https://gist.github.com/gavinandresen/5044482. Accessed: 2015-4-6.

[2] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in bitcoin. In *Financial Cryptography and Data Security*, pages 34–51. Springer, 2013.

[3] Adam Back. Hashcash-a denial of service counter-measure. http://www.hashcash.org/papers/hashcash.pdf, 2002.

[4] Aleksandra Marta Bal et al. *Taxation of virtual currency*. PhD thesis, Institute of Tax Law and Economics, Faculty of Law, Leiden University, 2014.

[5] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. Bitter to betterhow to make bitcoin a better currency. In *Financial cryptography and data security*, pages 399–414. Springer, 2012.

[6] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of clients in bitcoin P2P network. *CoRR*, abs/1405.7418, 2014.

[7] Bitcoin. https://en.bitcoin.it/wiki/Main_Page. Accessed: 2015-2-16.

[8] Bitcoin core version 0.9.0 released. https://bitcoin.org/en/release/v0.9.0. Accessed: 2015-2-9.

[9] Block. https://en.bitcoin.it/wiki/Block. Accessed: 2015-2-9.

[10] Common vulnerabilities and exposures. https://en.bitcoin.it/wiki/Incidents. Accessed: 2015-2-16.

[11] Contracts. https://en.bitcoin.it/wiki/Contracts. Accessed: 2015-2-17.

[12] Economic majority. https://en.bitcoin.it/wiki/Economic_majority. Accessed: 2015-2-16.

[13] Mining. https://en.bitcoin.it/wiki/Mining. Accessed: 2015-2-9.

[14] Proof of work. https://en.bitcoin.it/wiki/Proof_of_work. Accessed: 2015-2-9.

[15] Sec 2: Recommended elliptic curve domain parameters. http://perso.univ-rennes1.fr/sylvain.duquesne/master/standards/sec2_final.pdf. Accessed: 2014-10-22.

[16] Target. https://en.bitcoin.it/wiki/Target. Accessed: 2015-2-9.

[17] Transaction. https://en.bitcoin.it/wiki/Transaction. Accessed: 2015-2-9.

[18] Transaction fees. https://en.bitcoin.it/wiki/Transaction_fees. Accessed: 2015-2-9.

[19] Comparison of mining pools. https://en.bitcoin.it/wiki/Comparison_of_mining_pools. Accessed: 2015-4-13.

[20] Unconfirmed transactions. https://blockchain.info/unconfirmed-transactions. Accessed: 2015-4-7.

[21] Value overflow incident. https://en.bitcoin.it/wiki/CVE-2010-5139. Accessed: 2015-4-1.

[22] Bitcoin core version 0.9.0 released. https://en.bitcoin.it/wiki/Controlled_supply. Accessed: 2015-2-9.

[23] Top 100 richest bitcoin addresses. http://bitcoinrichlist.com/top100. Accessed: 2015-4-7.

[24] Global bitcoin nodes distribution. https://getaddr.bitnodes.io/. Accessed: 2015-4-13.

[25] Blockchain. Market price. https://blockchain.info/charts/market-price. Accessed: 2015-5-5.

[26] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten. Mixcoin. anonymity for bitcoin with accountable mixes. 2014.

[27] Jerry Brito and Andrea Castillo. *Bitcoin: A primer for policymakers*. Mercatus Center at George Mason University, 2013.

[28] Coinbase. A message from the coinbase security team. https://community.coinbase.com/t/a-message-from-the-coinbase-security-team/476. Accessed: 2015-4-28.

[29] Coincraft a1 28nm asic - bitmine. http://bitmine.ch/coincraft-28nm-asic/. Accessed: 2015-2-16.

[30] Nicolas T Courtois and Lear Bahack. On subversive miner strategies and block withholding attack in bitcoin digital currency. *arXiv preprint arXiv:1402.1718*, 2014.

[31] Nicolas T Courtois, Marek Grajek, and Rahul Naik. The unreasonable fundamental incertitudes behind bitcoin mining. *arXiv preprint arXiv:1310.7935*, 2013.

[32] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE, 2013.

[33] Dogecoin. Dogecoin. http://dogecoin.com/. Accessed: 2015-5-5.

[34] Ittay Eyal. The miner's dilemma. *arXiv preprint arXiv:1411.7099*, 2014.

[35] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.

[36] Ittay Eyal and Emin Gn Sirer. How to disincentivize large bitcoin mining pools. http://hackingdistributed.com/2014/06/18/how-to-disincentivize-large-bitcoin-mining-pools/. Accessed: 2015-4-16.

[37] GitHub. bitcoind. https://github.com/bitcoin/bitcoin/. Accessed: 2015-4-29.

[38] Andrew V Goldberg, Jason D Hartline, Anna R Karlin, Michael Saks, and Andrew Wright. Competitive auctions. *Games and Economic Behavior*, 55(2):242–269, 2006.

[39] Nicolas Houy. The bitcoin mining game. *Available at SSRN 2407834*, 2014.

[40] Business Insider. 927 people own half of all bitcoins. http://www.businessinsider.com/927-people-own-half-of-the-bitcoins-2013-12. Accessed: 2015-4-7.

[41] Benjamin Johnson, Aron Laszka, Jens Grossklags, Marie Vasek, and Tyler Moore. Game-theoretic analysis of ddos attacks against bitcoin mining pools. In *Financial Cryptography and Data Security*, pages 72–86. Springer, 2014.

[42] Nikolei Kaplanov. Nerdy money: Bitcoin, the private digital currency, and the case against its regulation. *Loy. Consumer L. Rev.*, 25:111, 2012.

[43] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. *IACR Cryptology ePrint Archive*, 2012:248, 2012.

[44] Ghassan O Karame, Elli Androulaki, and Srdjan Capkun. Double-spending fast payments in bitcoin. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 906–917. ACM, 2012.

[45] Philip Koshy, Diana Koshy, and Patrick McDaniel. An analysis of anonymity in bitcoin using P2P network traffic. 2014.

[46] Ladislav Kristoufek. What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. *arXiv preprint arXiv:1406.0268*, 2014.

[47] Joshua A Kroll, Ian C Davey, and Edward W Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013, 2013.

[48] Yoad Lewenberg, Yoram Bachrach, Yonatan Sompolinsky, Aviv Zohar, and Jeffrey S Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. 2015.

[49] Seattle City Light. Summary rates table. `http://www.seattle.gov/light/rates/summary.asp`. Accessed: 2015-3-13.

[50] Litecoin. Litecoin. `https://litecoin.org/`. Accessed: 2015-5-5.

[51] Loi Luu, Ratul Saha, Inian Parameshwaran, Prateek Saxena, and Aquinas Hobor. On power splitting games in distributed computation: The case of bitcoin pooled mining. 2015.

[52] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140. ACM, 2013.

[53] Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 397–411. IEEE, 2013.

[54] Andrew Miller, Ari Juels, Elaine Shi, Bryan Parno, and Jonathan Katz. Permacoin: Repurposing bitcoin work for data preservation. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 475–490. IEEE, 2014.

[55] Tyler Moore and Nicolas Christin. Beware the middleman: Empirical analysis of bitcoin-exchange risk. In *Financial Cryptography and Data Security*, pages 25–33. Springer, 2013.

[56] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[57] Cryptocoins News. Who owns all the bitcoins an infographic of wealth distribution. `https://www.cryptocoinsnews.com/owns-bitcoins-infographic-wealth-distribution/`. Accessed: 2015-4-7.

[58] Bart Preneel. Davies–meyer hash function. In *Encyclopedia of Cryptography and Security*, pages 136–136. Springer, 2005.

[59] Fergal Reid and Martin Harrigan. *An analysis of anonymity in the Bitcoin system*. Springer, 2013.

[60] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In *Financial Cryptography and Data Security*, pages 6–24. Springer, 2013.

[61] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980*, 2011.

[62] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. Bitiodine: Extracting intelligence from the bitcoin network. 2014.

[63] What is the incentive to collect transactions? `https://bitcointalk.org/index.php?topic=165.msg1595#msg1595`. Accessed: 2015-2-9.

Appendix A

MINER'S DATA

The following tables summarized information about mining pools collected from August 2014 to February 2015. In total, 27964 blocks were successfully mined.

Pool Name	Hash Rate	Mined Blocks	Avg(Size)	Avg(Fee) in Satoshi
Unknown	0.29	8139	277204.8275	8002385.1232
Discus Fish (F2Pool)	0.23	6422	364355.2862	9420625.9123
Ghash.io	0.17	4640	302594.5177	8016864.2491
AntMiner	0.07	1996	359694.5286	9197922.9770
KNCminer	0.05	1517	352460.2742	9186167.4766
BTC Guild	0.05	1339	281886.7558	8106436.8394
Eligius	0.04	1071	326588.7638	8803337.8189
Slush	0.03	922	351700.0423	9181720.6052
Bitfury	0.02	466	305900.3991	8010801.3605
Polmine.pl	0.01	419	127433.4964	5356376.5251
AgentD	0.01	383	348030.3029	9062264.5953
CloudHashing	0.01	177	183571.4068	6375181.5989
BitMinter	0.01	162	313733.7160	7653673.3827
Megabigpower	< 0.01	129	347110.5194	8710405.8527
TangPool.com	< 0.01	128	456400.2266	10335908.7422
Ozcoin	< 0.01	40	322535.2750	8446035.5750
TripleMining	< 0.01	10	123588.4000	4648597.8000
Bcif090	< 0.01	1	558962.0000	14033336.0000
Linenoise	< 0.01	1	292005.0000	8420954.0000
Baazee.de	< 0.01	1	618674.0000	11800397.0000
Con Kolivas	< 0.01	1	226377.0000	7101683.0000

Table A.1: Mining pools with the total number of blocks successfully mined, the average size and the average fee per block, expressed in Satoshi. Data was collected from August 2014 to February 2015.

Pool Name	Avg(Time diff)	Avg(Num Tx)	Avg(Input) in Satoshi
Unknown	580.0765	496.0500	1370.3247
Discus Fish (F2Pool)	563.6017	593.4083	1759.7557
Ghash.io	573.8651	534.8235	1491.2030
AntMiner	532.4123	617.8848	1790.6718
KNCminer	554.6618	604.4891	1747.8240
BTC Guild	576.7020	521.2577	1392.1777
Eligius	437.9356	550.0859	1572.4071
Slush	573.0184	600.4252	1732.0315
Bitfury	592.2961	505.8798	1491.5193
Polmine.pl	981.0788	348.2768	593.9547
AgentD	598.6319	605.7807	1730.6240
CloudHashing	578.4802	403.6328	880.0734
BitMinter	544.9444	561.1111	1569.4568
Megabigpower	739.6667	606.2403	1733.3953
TangPool.com	565.1563	703.3125	2225.7656
Ozcoin	664.5	570.4	1564.0250
TripleMining	677.8	327.9	596.9
Bcif090	917	738	2216
Linenoise	501	538	1419
Baazee.de	269	742	3432
Con Kolivas	792	429	1067

Table A.2: Mining pools with their average time to successfully mine block, the average number of transactions per block and the average input value, expressed in Satoshi. Data was collected from August 2014 to February 2015.