FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Lehrstuhl für Informatik 7

Rechnernetze und Kommunikationssysteme

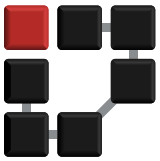
Jack Henschel

Quantitative Aspects of the Blockchain: Proof Of Work, its Energy Demand and Alternative Consensus Mechanisms

Seminararbeit im Fach Informations- und Kommunikationstechnik

8. Februar 2018

Please cite as:
Jack Henschel, “Quantitative Aspects of the Blockchain: Proof Of Work, its Energy Demand and Alternative Consensus Mechanisms,” Seminar Thesis (Seminararbeit), University of Erlangen, Dept. of Computer Science, February 2018.



Friedrich-Alexander-Universität Erlangen-Nürnberg
Department Informatik
Rechnernetze und Kommunikationssysteme

Martensstr. 3 · 91058 Erlangen · Germany

https://www7.cs.fau.de/

Quantitative Aspects of the Blockchain: Proof Of Work, its Energy Demand and Alternative Consensus Mechanisms

Seminararbeit im Fach Informations- und Kommunikationstechnik

vorgelegt von

Jack Henschel

geb. am 07. August 1997
in Neubrandenburg

angefertigt am

Lehrstuhl für Informatik 7
Rechnernetze und Kommunikationssysteme

Department Informatik
Friedrich-Alexander-Universität Erlangen-Nürnberg

Betreuer: **Prof. Dr-Ing. Reinhard German**
M.Sc. Jonas Schlund

Abgabe der Arbeit: **8. Februar 2018**

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.
Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.
I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

(Jack Henschel)
Erlangen, 8. Februar 2018

Abstract

Bitcoin is the most popular and well-known cryptocurrency to date. It is built upon a technology called blockchain and utilizes a distributed consensus mechanism. This Proof Of Work consensus algorithm is very energy intensive and reports of Bitcoin's growing energy usage are all over the news.

In this paper we review the underlying blockchain technology, quantitative figures of the Bitcoin and Ethereum network and how they compare to each other. We also investigate the concepts of some alternative consensus mechanisms.

Finally, we conclude that while the energy usage of Proof Of Work is very large and ever increasing, it opens up possibilities for new and previously unheard of applications. Furthermore, the elegance of Proof Of Work lies within its simplicity. In the future, however, other consensus algorithms might prevail, since Proof Of Work is only the first major iteration for the blockchain technology.

Contents

Chapter 1

Introduction

To understand how and why Proof Of Work is a valuable tool, we need to look at what the Bitcoin network is at its core.

> Bitcoin is a *consensus network* that enables a new payment system and a completely digital money. It is the first decentralized peer-to-peer payment network that is powered by its users with *no central authority or middlemen*.
>
> Much of the trust in Bitcoin comes from the fact that it *requires no trust at all*. Bitcoin is fully open-source and decentralized. [...] No organization or individual can control Bitcoin, and the *network remains secure even if not all of its users can be trusted*. [1]

The challenge here is finding a consensus between participants who do not know or trust each other. The underlying problem was first described by Leslie Lamport, Robert Shostak and Marshall Pease from Stanford Research Institute International, Menlo Park, California, in their paper "The Byzantine Generals Problem" [2].

A group of generals sieges an enemy city. The generals need to agree on whether to attack the city or retreat. Some generals may prefer to attack, others may want to retreat. The *distributed* generals need to coordinate. It is important that every general *agrees* on the decision and follows through, otherwise they lose.

The situation is further complicated by the presence of traitorous generals, who may not only cast a vote for a suboptimal strategy, but also do so selectively. For instance, if nine generals are voting, four of whom support attacking while four others are in favor of retreat, the ninth general may send a vote of retreat to those generals in favor of retreat, and a vote of attack to the rest. Those who received a retreat vote from the ninth general will retreat, while the rest will attack. This would be fatal for the attackers.

Furthermore, the generals cannot communicate directly, since they are distributed around the city, but need to use messengers. These may fail to deliver votes or may forge false votes. The so called *Byzantine fault tolerance* can be achieved if the loyal generals have a majority agreement on their strategy [2].

This hypothetical scenario can be mapped onto computer science by identifiying the generals with individual computers and messengers with digital communication systems such as the internet.

Chapter 2

Proof Of Work

The problem described in the introduction (Chapter 1) is solved by the Proof Of Work technique first described by Satoshi Nakamoto in his paper "Bitcoin: A Peer-to-Peer Electronic Cash System" in 2008 [3].

Bitcoin tackles this problem by having each "general" work on a mathematical problem that is known to take a certain average amount of time. When a general finds a solution he passes his solution onto the other generals who will verify and then incorporate the answer to the previous problem into a new problem.

The consensus is intrinsically linked to the mathematical problem, hence the generals always trust the longest chain-of-answers available, as it would be impractical, almost impossible, for an adversary to counterfeit the long chain of answers.

Nakamoto described the basic algorithm nicely in his paper:

> The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.
>
> [The block's nonce is incremented] until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it. [3]

2.1 Block Header Structure

Without going into too much detail on the internals of the Bitcoin implementation, this serves as a brief overview.

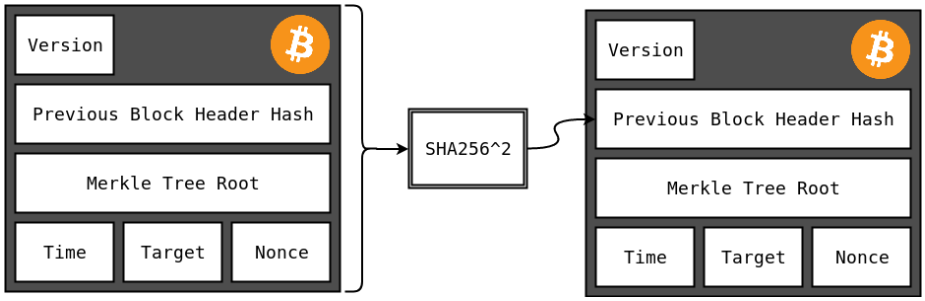


Figure 2.1 – Bitcoin Blockchain structure

Each block in the blockchain is made up of the block header (refer to Figure 2.1 and Listing 2.1) and the block data, in which the transactions are stored.

```
1  /** Nodes collect new transactions into a block, hash them into a hash
2  * tree, and scan through nonce values to make the block's hash satisfy
3  * proof-of-work requirements.  When they solve the proof-of-work, they
4  * broadcast the block to everyone and the block is added to the
5  * blockchain. The first transaction in the block is a special one
6  * that creates a new coin owned by the creator of the block. */
7  class CBlockHeader {
8    public:
9    int32_t nVersion;
10   uint256 hashPrevBlock;
11   uint256 hashMerkleRoot;
12   uint32_t nTime;
13   uint32_t nBits;
14   uint32_t nNonce;
15 }
```

Listing 2.1 – `src/primitives/block.h`, The Bitcoin Core Developers [4]

The block header contains the version of the blockchain for which this block was created (`nVersion`) as a four byte field. The next field is 32 bytes long and contains the Secure Hash Algorithm 2 256 Bits (SHA-256) hash of the previous block's header (`hashPrevBlock`) – not the hash of the entire block. This establishes the chain of answers we discussed previously. The next field is again a SHA-256 hash (`hashMerkleRoot`), but over the block data (i.e. all transactions of this block), which creates the connection between the block header and the block's content. Then there is another four byte field storing the timestamp of the block creation (in Unix time). Its value should be considered more as a hint and less as a precise date. The fifth field (`nBits`) describes the target difficulty of finding a block at the time the block was created. Lastly, the `nNonce` field is used as a nonce to set arbitrary values and therefore allow finding a block with the required difficulty.

When a miner sets the difficulty too low, hence increasing his chances of finding a block, other nodes simply reject the new block, as the difficulty of the new block

does not meet their difficulty threshold. By this mechanism "the network" adjusts the difficulty automatically.

The basic procedure the miners follow is described by the code in Listing 2.2.

```
var hash uint32 = 1
var myBlockHeader = newBlockHeader() // creates and fills block header
while (hash > myBlockHeader.nBits) {
  hash = sha256(sha256(myBlockHeader))
  myBlockHeader.nNonce++
  // Note: might need to update hashPrevBlock,
  // hashMerkleRoot, nTime and/or nBits
}
print("Heureka!")
// publish blocks to other nodes
```

Listing 2.2 – Mining Procedure

First, a new block header is allocated and filled with the relevant data (previous block hash, timestamp, etc.). Then the SHA-256 squared hash of the block header is calculated. If this hash fulfills the required difficulty, we consider ourselves very lucky and publish the block to other nodes. These verify the solution, i.e. calculate the SHA-256 squared hash of the block header, and after successful verification publish the block to other nodes.

Bitcoin uses SHA-256 function squared, that is applying the hash function twice, due to the birthday attacks on the smaller but related Secure Hash Algorithm 1 (SHA-1). SHA-1's resistance to birthday attacks has been known to be vulnerable and a collaboration between Google Research and CWI Amsterdam has shown it is possible two create two distinct PDF files which result in the same SHA-1 hash [5].

2.2 Incentives

There is a difference between *mining* and *minting* a coin. Minting is the creation of coins – real or virtual – by doing a negligible amount of work, for instance what federal banks are doing. For cryptocurrencies this refers to the creation of blocks without having to do find a nonce matching the difficulty requirement. Mining on the other hand requires doing hard work, in the case of Bitcoin solving hash puzzles with a very low chance of success.

As Jemima Kelly put it for Reuters [6]:

> The process has come to be known as "mining" because it is slow and intensive, reaping a gradual reward in the same way that minerals such as gold are mined from the ground.

The miners are not doing this hard work out of altruism, but because there is a business case for it. Solving one of the hash puzzles is rewarded with a so

called block reward and additionally with all transaction fees included in the block. The block reward is set by the network and is currently at 12.5 BTC. This value is halved every 210,000 blocks (this equates to about four years at a block time of ten minutes) and the next block reward halving is expected to happen in mid 2020 [7]. The transaction fees are set by each node initiating a transaction and can be set arbitrarily high or low, though a low transaction fee is unlikely to provide the miner with enough incentive to include this particular transaction in the next block, as the volume of transactions per block is limited.

Examining the expenses of Bitcoin mining on the other hand, we first take a look at the upfront cost of buying mining hardware.

2.3 Mining hardware

When Bitcoin first started in 2009 it was easily possible to mine blocks with regular Central Processing Units (CPUs) and Graphics Processing Units (GPUs). However, as the power of the entire network increased, their hash rate was too low (i.e. not producing enough hashes per second), and they were replaced with Field-Programmable Gate Arrays (FPGAs). Nowadays it is only profitable to mine Bitcoin with specialized hardware, so called Application-specific Integrated Circuits (ASICs).

Table 2.1 – Hashing Power of Bitcoin Mining Hardware, Source: [8]

Hardware	Hashing Power
CPU	25 MH/s
GPU	500 MH/s
FPGA	10,000 MH/s
ASIC	14,000,000 MH/s

One of the most powerful and efficient appliances (as of December 2017) is the Bitmain Antminer S9, shown in Figure 2.2. This is not a single chip, but rather a complete unit only requiring an additional power supply unit.

Table 2.2 – Bitmain Antminer S9 Specifications, Source: Bitmain [9]

Bitmain Antminer S9 Specifications	
Hash Rate	13.5 TH/s
Power Consumption	1300 W
Power Efficiency	0.098 J/GH
Lithography Process	16 nm
Price	1500 EUR

Figure 2.2 – Bitmain Antminer S9, Source: Bitmain [9]

Due to the modern lithography process of 16nm it is able to achieve a very high power efficiency of 0.098 joules per gigahash. Nevertheless, the entire unit consumes 1.3 kilowatts of power (Table 2.2).

Chapter 3

Energy Demand of Proof Of Work

Proof Of Work is a very energy intensive application. Based on assumptions of the power consumption of mining hardware shown in Chapter 2.3 and the total network hash rate the energy demand of the entire mining network can be estimated.

3.1 Bitcoin's energy demand

First, we use the Antminer S9 shown in Chapter 2.3 to calculate a lower bound for Bitcoin's energy demand.

$$H_R = 14,788,692,144 \frac{GH}{s} \tag{3.1}$$

The hash rate was obtained on 14th of December 2017 from [10]. As we already discussed, the block header contains the difficulty for the current block and because the difficulty is directly linked to the hash rate of the network, it is therefore possible to derive the current hash rate of the entire network [11].

$$P = \eta \cdot H_R = 0.1 \frac{\mathrm{J}}{\mathrm{GH}} \cdot 14.79 \frac{\mathrm{EH}}{\mathrm{s}} = 1{,}478{,}869{,}214 \frac{\mathrm{J}}{\mathrm{s}} \approx 1.4\,\mathrm{GW} \tag{3.2}$$

Multiplying the efficiency of the hardware η with network hash rate H_R results in an electrical power of 1.4 gigawatts. Since we assumed very high efficiency in this calculation, the results need to be interpreted as an absolute lower bound. As Karl J. O'Dwyer and David Malone noted in their paper "Bitcoin Mining and its Energy Footprint" in 2014:

> The actual network will be a mix of hardware of types at different levels of efficiency, so we expect that the actual efficiency will be between the two. This suggests that the total power used for Bitcoin mining is around 0.1-10GW. [12]

However, we also need to keep in mind that the efficiency of mining hardware only continues to increase and most Bitcoin miners are running very up to date and efficient hardware, in order to reduce their electricity cost.

Multiplying these power figures with an amount of time, we get hourly, daily and annual values for Bitcoin's energy usage.

Table 3.1 – Bitcoin energy usage

	Efficiency	Hourly (E_H)	Daily (E_D)	Annually (E_A)
Lower bound:	0.1 J/GH	1.4 GWh	33.6 GWh	12.3 TWh
Digiconomist:	0.283 J/GH		90 GWh	33 TWh

Table 3.1 includes estimates from Digiconomist [13], who used a different efficiency value for the mining hardware. According to their numbers, Bitcoin mining is 0.15% of the world's electricity consumption [13].

3.2 Ethereum's energy demand

Ethereum is another very popular blockchain. Unlike Bitcoin, its Proof Of Work algorithm is *ASIC-resistant*, thus mainly GPUs are utilized for mining, as they can be reconfigured. This leads to a very diverse, decentralized and heterogenous landscape of miners all around the world, because anyone with a GPU can start mining Ethereum without the need of buying additional hardware etc. but it also leads to a higher overhead and generally less efficient mining.

The NVIDIA GTX 1070 is currently the most efficient GPU for Ethereum mining [14], with a conversion rate of 5 megahashes per second (the Antminer S9 we mentioned for Bitcoin mining is at 13.5 gigahashes per second, see Section 2.3).

Just like we did for Bitcoin, the hash rate of the entire Ethereum network can be derived from the block difficulty. Ethereum is currently at 125 terahashes per second (obtained on 14th of December 2017 from [15]).

The lower bound for the electrical power can then again be calculated by multiplying the efficiency with the hash rate, resulting in a value of 625 megawatts (3.3), about half the power of the Bitcoin network (see Equation 3.2).

$$P = \eta \cdot H_R = 625\,\text{MW} \tag{3.3}$$

Table 3.2 – Ethereum energy usage

	Efficiency	Hourly (E_H)	Daily (E_D)	Annually (E_A)
Lower bound:	5 J/MH	625 MWh	15 GWh	5.5 TWh
Digiconomist [16]:	10.83 J/MH		30 GWh	11 TWh

3.3 Comparison: Bitcoin and Ethereum network

Having calculated these numbers, we can do a quantitative comparison between Bitcoin and Ethereum. Numbers in Table 3.3 not previously mentioned in this paper are from [17, 18].

Table 3.3 – Energy usage comparison of Bitcoin and Ethereum

	Bitcoin	Ethereum
Hashrate	14 EH/s	125 TH/s
Price per coin	16,385 USD	728 USD
Transaction per day	400,000	900,000
Transaction volume per day	3 mio. BTC	10 mio. ETH
Transaction volume per day	50 bill. USD	8 bill. USD
Block time	10 min	15 s
Annual energy	12.3 TWh	5.5 TWh
Energy per block (E_{Block})	233 MWh	2.6 MWh
Energy per transaction (E_{TXN})	84 kWh	16 kWh

While Bitcoin's hash rate is roughly ten times higher than Ethereum's, Ethereum still manages to push twice the amount of transactions per day compared to Bitcoin. This is due to the very aggressive block generation time of 15 seconds, compared to the 10 minutes Bitcoin uses.

$$E_{Block} = \frac{E_{1H}}{Blocks_{1H}} \tag{3.4}$$

$$E_{TXN} = \frac{E_{1D}}{TXN_{1D}} \tag{3.5}$$

With these transaction statistics we can derive the amount of energy it takes to produce a single block (Equation 3.4) or even a single transaction (Equation 3.5). While it takes only one-tenth of the electrical energy to produce a block in the Ethereum network compared to Bitcoin, looking at the energy demand per transaction, Ethereum uses only one-forth of the electrical energy compared to Bitcoin (Table 3.3). Generally speaking for Proof Of Work algorithms, as the network hash rate (and therefore the block difficulty) increases, it takes more and more

energy to produce a block. This in turn then affects the energy required for each transaction, as they need to be stored inside the blocks.

3.4 Electricity demand comparison

Recalling the lower bounds for the annual energy demand of Bitcoin (12.3 TWh) and Ethereum (5.5 TWh), we can compare these numbers to other real world examples. However, comparisons like those of various newspaper against the energy usage of various countries are not referable. A comparison to other large energy users in the field of information technology yields more valuable results.

The "United States Data Center Energy Usage Report" by the U.S. Department of Energy states:

> In 2014, data centers in the U.S. consumed an estimated 70 [TWh], representing about 1.8% of total U.S. electricity consumption. [19]

Those include the large data centers run by the "Big Four": Amazon, Google, Facebook and Microsoft. Digging further, data reported by Bloomberg New Energy Finance indicates Bitcoin's power demand is comparable to that of Amazon, the second biggest corporate buyer of electric power in the U.S., alone (Table 3.4) [20].

Table 3.4 – Comparison of Bitcoin and Ethereum to Amazon and Google

	Ethereum	Bitcoin	Amazon (US)	Google (US)
Electric capacity in GW	0.6	1.4	1.219	3.186

Another interesting illustration is NSA's Utah Data Center, formally known as "Intelligence Community Comprehensive National Cybersecurity Initiative Data Center", depicted in Figure 3.1.

Figure 3.1 – NSA Utah Data Center, Source: Parker Higgins [21]

> The U.S. Army Corps of Engineers has estimated the NSA facility will require 65 megawatts of electricity to run its equipment around the clock, which would put its annual power bill at about $18 million. [22]

Therefore, one could run ten of NSA's data centers for the power usage of the Ethereum network. We will leave it up to the reader to decide what is more valuable.

While the Bitcoin and Ethereum network together use at least 18 terawatthours of electricity per year, this is only one tenth of the renewable energy produced in Germany alone in 2016, as 188 TWh were generated, according to the German Ministry for Economy and Energy [23].

Chapter 4

Alternative Consensus Mechanisms

Proof Of Work is the consensus mechanism which got, and still gets, cryptocurrencies started. However just like cryptocurrencies are just one application for blockchains, Proof Of Work is just one consensus mechanism for blockchains. As the name implies, all the algorithm has to do is achieve a consensus - on something - amongst the participants, though we discussed the difficulty of this in the introduction (Chapter 1).

4.1 Proof Of Useful Work

Instead of spending all the work on simply computing hash functions, the energy could also be used for something useful. In this section we take a look at various approaches to Proof Of Useful Work.

4.1.1 Data Furnace

One of the first ideas that comes to mind is using the generated heat by the mining process to heat homes and offices. On first sight, this idea might seem astronomic. However, researches from Microsoft argue in their paper “The Data Furnace: Heating Up with Cloud Computing” that the approach leads to a smaller carbon footprint and a reduced cost of hardware ownership. In the case of cloud computing it also significantly decreases the proximity to users, therefore lowering latency. They call the concept “Data Furnance” [24].

Though, the practical challenges of the implementation are another factor to consider, and may be even greater than those of the original problem.

4.1.2 Science

A more realistic example for Proof Of Useful Work schemes is found in the field of science.

> [The] Primecoin network searches for special prime number chains known as Cunningham chains and bi-twin chains. The distribution of these prime chains are not well-understood currently as even for its simplest case twin primes their infinite existence is not proven. [25]

Finding these prime number chains becomes exponentially harder as the chain length is increased. That way, the difficulty in the Primecoin network is set and the difficulty is adjusted after each single block, targeting one block per minute [25].

Primecoin is comparable to the Great Internet Mersenne Prime Search (GIMPS) project which is run by enthusiasts on their computers all around the world and has found most of the largest prime numbers known to date.

4.1.3 Problems

Most Proof Of Useful Work schemes suffer two general problems:

Exhaustable, meaning running out of problems to solve. While great for the problem itself, this is devastating for the blockchain as no more blocks can be generated. Additionally, in many cases we do not only not know if we will run out of problems, but also when.

Not equiprobable, meaning not all solutions are equally likely. This is problematic because Proof Of Work builds upon each and every participant having the same likelihood of solving the challenge. This does not imply that such a system is generally impossible, though.

4.2 Proof Of Stake

The most popular contender for Proof Of Work is Proof Of Stake. Generally speaking, instead of using hardware, energy and time as resources, the mechanism uses the tokens of the cryptocurrency itself as a resource. This has the potential benefits of lowering the overall cost of running a cryptocurrency, as it is less computationally intensive and does not require specialized hardware. It also creates so called *Stakeholder incentives*, meaning everyone owning the cryptocurrency has an interest in it staying valuable, therefore reinforcing “good” behavior.

There exist several variations of this scheme:

- Proof Of Stake: Stake (i.e. voting weight) of coin increases as long as the coin is not used
- Proof Of Deposit: coin is frozen for some time, but can be reclaimed
- Proof Of Activity: any online coin (owned by an online node) can win (coins are randomly chosen and have to create the next block signed with their key)

Though, as the famous cryptocurrency researcher Vitalik Buterin noted in 2014, coming up and creating an alternative to Proof Of Work is challenging:

> After years of research, one thing has become clear: proof of stake is non-trivial – so non-trivial that some even consider it impossible. [26]

Buterin is the co-creator and inventor of Ethereum, who has also written Ethereum's whitepaper and it still working on the technology.

4.2.1 Proof Of Stake: Casper

The developers of Ethereum, Buterin amongst them, are currently planning a transition from pure Proof Of Work to a hybrid Proof Of Work-Proof Of Stake based blockchain. As of early 2018, this is still work in progress. *Casper* is a smart contract that will implement and monitor Proof Of Stake on the Ethereum blockchain. Smart contracts are small pieces of code run by every single participating node. Thus, Casper is also executed and most importantly verified by every single node in the network.

The first iteration, now called *naive Proof Of Stake*, suffered from the *nothing-at-stake problem*: it didn't punish participants for validating more than one history (i.e. more than one block), therefore creating splits in the blockchain and undermining its credibility. This problem has been addressed in *modern Proof Of Stake* versions, such as *Casper Friendly Finality Gadget (FFG)*, developed by Vitalik Buterin et al., and *Casper Correct By Construction (CBC)*, developed by Vlad Zamfir et al. [27]

Both Casper versions use the same underlying concept: anyone can bond tokens (the generic term for coins), while decisions leading to a convergence on the blockchain are monetarily rewarded, decisions resulting in a split of the chain are punished.

First, validators (the term for referring to the stakeholders) transfer their chosen stake to Casper, called a *security deposit*. Then, two rounds of voting occur: in the first round the validators *prepare* the next block. This happens by voting which block should be appended to the blockchain next. A block is selected by receiving a least two thirds of staked *Ether*, which is the currency Ethereum uses. In the second round, the validators need to commit to the previously selected block [27]. This procedure is illustrated in Figure 4.1.

Because two rounds of voting occur, but each validator can only vote once per item on the blockchain, this mechanism builds a consensus on the blockchain. After these voting rounds, Casper will then *slash* bad validators and *reward* good validators.

Slashing bad validators is implemented by them losing their stake. Rewarding on the other hand is accomplished by retransferring their stake and additionally

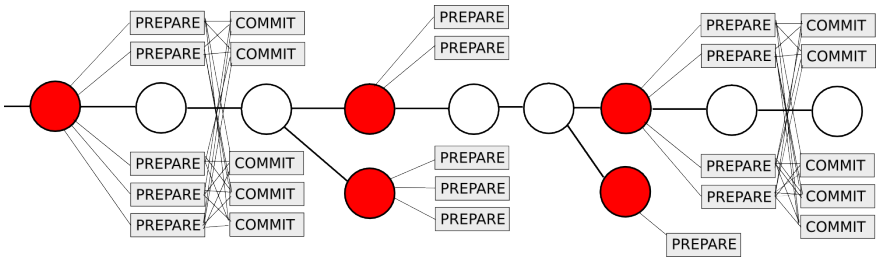


Figure 4.1 – Ethereum voting: Prepares and Commits, Source: [27]

receiving prorated amount of fees of transactions in the block. Due to its nature, the general concept is also referred to as "Consensus By Bet" [28]:

Coins, either real or virtual, need to be spent, and one wants to choose the most likely outcome as to maximize the chance of winning. In this case that means voting for the most likely block to be appended to the blockchain, instead of voting for a block which might bring personal benefit, but is unlikely to be voted for by others.

Proof Of Stake is so complex because it not only changes the consensus algorithm, but also the entire economics around the blockchain – also known as *cryptoeconomics* [29].

A general concern with Proof Of Stake methods is often referred to as the *rich get richer scheme*: as one's voting power and dividends of the transaction fees proportionally increase with the staked amount, wealthy owners of coins gradually become richer. While many researchers, including the authors of various Proof Of Stake schemes themselves, argue this is not the case, only the future can bring certainty on how the economics of blockchains play out.

4.3 Delegated Proof Of Stake

This consensus mechanism was first implemented by the BitShares blockchain and has been updated several times since. Broadly speaking, it is comparable to the U.S. electoral college system. Under Delegated Proof Of Stake, stakeholders elect a number of *witnesses* to generate blocks [30]. Witnesses serve the role of validating signatures and timestamping transactions by including them in blocks, therefore they are BitShares equivalent of miners (without the intensive work, of course). Each account is allowed one vote per share per witness, this process is known as *approval voting*. The top N witnesses by total approval are selected, where N is defined such that at least 50% of voting stakeholders believe there is sufficient decentralization. When stakeholders express their desired number of witnesses, they have to cast a vote for at least that many witnesses. The elected witnesses are paid for their service of producing blocks. The pay rate is set by the stakeholders via their elected

delegates. If a witness does not produce a block in their time slot, then that time slot is skipped, they are not paid, and may be voted out in the future, as they did not fulfill the promise to their voters. A single witness can't sign invalid blocks as the block needs confirmation by the other witnesses as well [30].

The slate of active witnesses is updated daily when the votes are tallied. The witnesses are shuffled and each witness is given a turn to create one block at a fixed schedule, in the case of BitShares this is every 2 seconds [30].

All network parameters (such as fees, block intervals or transaction sizes) can be tuned via elected delegates, resulting in a fully democratic system and, like those in the real world, a very complex one, too. But anyone can monitor network health by observing the witness participation rate and BitShares has maintained 99% witness participation [30]. When witness participation dwindles, users of the network can react by simply allowing more time for transactions to confirm. The low number of witnesses enables a high performance level of effectiveness in creating the blocks and general maintenance of the blockchain.

Optionally - this is not related to Delegated Proof Of Stake - each transaction in the BitShares network may include the hash of a recent block. If this is done, the signer of the transaction can be confident that their transaction may not be applied to any blockchain that does not include that block, providing an additional security benefit. This implies the stakeholders themselves directly certify the long-term integrity of the transaction history [30].

4.4 Proof Of Burn

Another experimental consensus mechanism is Proof Of Burn, originally conceived by Ian Stewart. The underlying idea is sending coins to an irretrievable address is rewarded with proportionate votes on the blockchain for the "spent" coins, consequently the used resource here is a cryptocurrency. This seems extremely counterintuitive a first, but we need to compare "spending", i.e. burning, coins with buying mining hardware and running it. Instead of investing in resources for Proof Of Work, we are directly investing our currency into the blockchain.

In Slimcoin "Proof-of-work is used as a mean for generating the initial money supply. As time passes and as the network accumulates a sufficient supply of coins, proof-of-work mining will become less necessary. Therefore, the network will rely more on proof-of-burn and proof-of-stake" [31].

Advantages of Proof Of Burn include very low energy consumption, especially when compared to Proof Of Work, and no need to invest in hardware. Additionally, the authors argue, there are lesser artificial price swings, due to there being no

influence of mining pools or new mining hardware, and no simple *rich get richer schemes*, because Proof Of Burn rewards entrepreneurial risk instead of wealth [32].

One application of Proof Of Burn is bootstrapping a cryptocurrency: Users who "burn" the original coin (take Bitcoin as an example) by sending it to an irretrievable address are awarded with coins of new currency (i.e. Altcoins). As all transactions are stored on the blockchain, this process is fully transparent and can be verified at any time. This can be considered as a form of extreme, one-time investment.

4.5 Proof Of Luck

Proof Of Luck is the final consensus mechanism we take a look at in this paper. It was conceived by Mitar Milutinovic, Warren He, Howard Wu and Maxinder Kanwal from University of California in Berkeley. Proof Of Luck utilizes the Trusted Execution Environment (TEE) of modern CPUs as a resource, as there is only one per CPU and it is therefore a limited resource. Accordingly, it relies entirely on the security and correctness of TEE provided by the manufacturer.

It combines *Proof Of Work*, *Proof Of Time* and *Proof Of Ownership*, and in their whitepaper the authors argue that it "offers low-latency transaction validation, deterministic confirmation time, negligible energy consumption, and equitably distributed mining" [33].

The basic concept of Proof Of Luck is again a round based voting consensus. Each round is allocated a time slot, and participants have to generate a random number from within the TEE. As this process takes places inside a so called *enclave*, it is ensured that the generated number is a least pseudo random and the number was generated within the specified time slot, which prohibits generating multiple random numbers.

All participants then share their random number and the highest number wins, therefore allowing the winning participant to create the next block.

There are many security concerns with Proof Of Luck: correctness of the TEE, trust towards the hardware manufacturers, presence of cheap, ubiquitous TEEs, only to name a few. Ergo it can not really be considered for public, sensitive applications, but it is an interesting experiment showing where blockchains and consensus mechanisms can be taken.

Chapter 5

Conclusion

Bitcoin, and in particular Proof Of Work, solves a really hard computer science problem: establishing trust between largely anonymous parties without requiring trust. Not only does it solve this problem, as we have seen by comparison to other consensus mechanisms, it also solves it very elegantly.

As Tomaso Aste, from the University College London Centre for Blockchain Technologies, wrote in his paper "The Fair Cost of Bitcoin Proof of Work":

> I conclude that the current cost, although large, is of a justified order of magnitude for an anonymous systems operating between untrustful parties. [8]

Comparing a transaction system like Bitcoin or Ethereum can not only be done on a energy level, the staff, paper work and legal systems attached to traditional banks also need to be taken into consideration. Not to mention the costs of security vans carrying physical currency, energy to melt metal to mint coins, and even the costs of building brick & mortar banks.

We do not think Bitcoin can scale indefinitely with Proof Of Work. Looking at the energy demand is just one example, but also the rampant transaction fees or the ever growing size of the blockchain.

However, Cryptocurrencies and Blockchains are here to stay, anyway. It would come close to a miracle, if the first major iteration of this concept (i.e. Bitcoin) was free of issues. These will be corrected and solved in further iterations of the concepts.

The applications for blockchains are vast and we might not have come up with the ultimate use-case for blockchains, yet. Meanwhile, the technology is rapidly evolving: Bitcoin's *Lightning Network* and *Segregated Witness Benefits*, alternative and hybrid consensus mechanisms, such as currently planned by the developers of the Ethereum blockchain, are all just examples of the rapid rate of progress. Maybe

a blockchain using a hybrid consensus mechanism will become prevalent. Maybe an entirely new consensus mechanism will be conceived and succeed.

The current position of blockchain technology can be compared to the state of networked computers in the 70s and early 80s of the last century. Not a huge amount of value delivered at the time, yet very few could have foreseen the impact of the internet and personal computing would have on the world. If blockchains do present a new leap forward in how we communicate, the electricity used will not be wasted, as they are an invaluable tool for the future of distributed computing.

List of Acronyms

CPU	Central Processing Unit
GPU	Graphics Processing Unit
FPGA	Field-Programmable Gate Array
ASIC	Application-specific Integrated Circuit
SHA-1	Secure Hash Algorithm 1
SHA-256	Secure Hash Algorithm 2 256 Bits
GIMPS	Great Internet Mersenne Prime Search
FFG	Friendly Finality Gadget
CBC	Correct By Construction
TEE	Trusted Execution Environment
NSA	National Security Agency

List of Figures

List of Tables

Bibliography

[1] (2017-12-09) Bitcoin FAQ. [Online]. Available: https://bitcoin.org/en/faq

[2] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4/3, pp. 382–401, 07 1982. [Online]. Available: http://lamport.azurewebsites.net/pubs/byz.pdf

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] The Bitcoin Core developers. (2017-12-09) Bitcoin core. [Online]. Available: https://github.com/bitcoin/bitcoin

[5] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full SHA-1," 2017. [Online]. Available: https://shattered.io

[6] J. Kelly. (2016) Bitcoin 'miners' face fight for survival as new supply halves. Thomson Reuters. [Online]. Available: https://www.reuters.com/article/us-markets-bitcoin-mining/bitcoin-miners-face-fight-for-survival-as-new-supply-halves-idUSKCN0ZO2CW

[7] (2017-12-14) Bitcoin block reward halving countdown. [Online]. Available: http://www.bitcoinblockhalf.com/

[8] T. Aste. (2016) The fair cost of bitcoin proof of work. [Online]. Available: https://ssrn.com/abstract=2801048

[9] (2017-11-25) Antminer S9. [Online]. Available: https://www.antminereurope.com/antminer-s9/

[10] (2017-12-14) Bitcoin stats. [Online]. Available: https://blockchain.info/stats

[11] (2018) Bitcoin wiki: Difficulty. [Online]. Available: https://en.bitcoin.it/wiki/Difficulty

[12] K. J. O'Dwyer and D. Malone, "Bitcoin mining and its energy footprint," 2014. [Online]. Available: http://karlodwyer.com/publications/pdf/bitcoin_KJOD_2014.pdf

[13] (2017-12-12) Bitcoin energy consumption index. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption

[14] N. Kirsch. (2017) Geforce GTX 1070 ethereum mining. [Online]. Available: http://www.legitreviews.com/geforce-gtx-1070-ethereum-mining-small-tweaks-great-hashrate-low-power_195451

[15] (2017-12-14) Ethereum network hashrate chart. [Online]. Available: https://etherscan.io/chart/hashrate

[16] (2017-12-12) Ethereum energy consumption index. [Online]. Available: https://digiconomist.net/ethereum-energy-consumption

[17] (2017-12-14) Bitcoin (BTC) price stats and information. [Online]. Available: https://bitinfocharts.com/bitcoin/

[18] (2017-12-14) Ethereum (ETH) price stats and information. [Online]. Available: https://bitinfocharts.com/ethereum/

[19] A. Shehabi, S. J. Smith, D. A. Sartor, R. E. Brown, M. Herrlin, J. G. Koomey, E. R. Masanet, N. Horner, I. L. Azevedo, and W. Lintner, "United States data center energy usage report," 2016. [Online]. Available: https://eta.lbl.gov/publications/united-states-data-center-energy

[20] B. Eckhouse. (2017-11-30) Google, biggest corporate buyer of clean power, is buying more. Bloomberg New Energy Finance. [Online]. Available: https://www.bloomberg.com/news/articles/2017-11-30/google-biggest-corporate-buyer-of-clean-power-is-buying-more

[21] P. Higgins. (2014) EFF photograph of NSA's Utah data center. [Online]. Available: https://en.wikipedia.org/wiki/File:EFF_photograph_of_NSA%27s_Utah_Data_Center.jpg

[22] S. Oberbeck. (2013) NSA Bluffdale Center won't gobble up Utah's power supply. [Online]. Available: http://archive.sltrib.com/story.php?ref=/sltrib/news/56493868-78/power-center-electricity-utah.html.csp

[23] "Zeitreihen zur Entwicklung der erneuerbaren Energien in Deutschland 1990 - 2016," 2017. [Online]. Available: http://www.erneuerbare-energien.de/EE/Redaktion/DE/Downloads/

zeitreihen-zur-entwicklung-der-erneuerbaren-energien-in-deutschland-1990-2016.pdf

[24] J. Liu, M. Goraczko, S. James, C. Belady, J. Lu, and K. Whitehouse, "The data furnace: Heating up with cloud computing." USENIX, 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/the-data-furnace-heating-up-with-cloud-computing/

[25] (2017-11-25) About Primecoin. [Online]. Available: http://primecoin.io/about.php

[26] V. Buterin. (2014-10-03) Slasher ghost, and other developments in proof of stake. [Online]. Available: https://blog.ethereum.org/2014/10/03/slasher-ghost-developments-proof-stake/

[27] ——, "Casper the friendly finality gadget: Basic structure," 2017. [Online]. Available: https://github.com/ethereum/research/

[28] ——. (2015-12-28) Understanding serenity, part 2: Casper. [Online]. Available: https://blog.ethereum.org/2015/12/28/understanding-serenity-part-2-casper/

[29] ——. (2016-12-30) A proof of stake design philosophy. [Online]. Available: https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51

[30] (2017-12-09) Delegated proof-of-stake consensus. [Online]. Available: https://bitshares.org/technology/delegated-proof-of-stake-consensus/

[31] P4Titan, "Slimcoin whitepaper," 2014. [Online]. Available: www.slimcoin.club/whitepaper.pdf

[32] (2017-12-09) What is proof of burn (eli5)? [Online]. Available: http://slimco.in/proof-of-burn-eli5/

[33] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in *Proceedings of the 1st Workshop on System Software for Trusted Execution*, ser. SysTEX '16. ACM, 2016, pp. 2:1–2:6. [Online]. Available: http://doi.acm.org/10.1145/3007788.3007790

Source Material and Resources available at:

`https://gitlab.cs.fau.de/in76yleb/seminar-blockchain`