# Green Mining: toward a less energetic impact of cryptocurrencies

Philippe Jacquet
Nokia Bell Labs
Nozay, France
Email: philippe.jacquet@nokia-bell-labs.com

Bernard Mans
Macquarie University
Sydney, Australia
Email: bernard.mans@mq.edu.au

***Abstract*—While cryptocurrencies continue to gain popularity, their energy cost is increasingly becoming unsustainable. In this paper, we present an innovative scheme which eliminates the burden of the proof of work which is the main cause of the energy waste in cryptocurrencies such as Bitcoin. The scheme is based on a green leader election scheme which guarantees a bounded average number of simultaneous mining whatever the size of the population in competition.**

## I. Introduction

Popular cryptocurrencies have now been effectively in use for nearly ten years (*e.g.* Bitcoin [9]). In addition to the increasing appetite for its key financial aspects (e.g., fast, transparent, secure and private), their potential to facilitate, verify, or enforce the negotiation or performance of various transactions provide an important innovative service that will have an ongoing impact in all aspects of our life.

Actual cryptocurrencies are completely decentralized protocols. They are based on the blockchain concept where transactions are regrouped by blocks and committed in a common distributed directory connected in peer to peer organization. The transactions and the blocks are chained via a chain of hash function which is too expensive to forge and falsify. The blockchain allows that each block and each transaction can be checked back by anyone as if there would be in a gigantic distributed accounting book. The block has also the function of tracing and controling the inflation of the Bitcoin volume and facial value. This is done by giving a reward to the block miner (the action of proposing a block) which is accepted by distributed consensus when the block is "committed" (e.g., in Bitcoin, followed by six newer blocks). A block which is not committed disappears.

One drawback of popular cryptocurrencies such as Bitcoin is the moderation of block mining rate. Since mining reward is a fundamental tool in the control of volume of values, there is a fierce competition in block mining. Some individuals may be tempted to mine multiple blocks in order to augment their revenue or bend the system in their favor (e.g., by double spending).

In order to avoid that too many simultaneous blocks are mined and proposed for commitment, Bitcoin requires a "proof of work" by forcing the block miner to execute a large number of CPU cycles via iterated hash computation before submitting his/her block. The idea of computational puzzle as proof of work is not new, it appears first in the seminal work of Dwork and Noar [5] and was later suggested to avoid Sybil attack by Aspnes et al. [1]. The proof of work provides scarcity and uniqueness that help reduce the individual block mining rate.

However the "Proof of Work" presently costs 100,000,000,000 CPU cycles (around 10 minutes on a current PC) and if one million miners contends at the same time, then the energy cost of mining one Bitcoin exceeds the value of the Bitcoin itself. Furthermore this cost is expected to quadratically grow or at least linearly grow with the number of contending miners leading to a protocol whose energy cost can absolutely not be sustainable in the near future.

In order to avoid these catastrophic consequences, a plethora of alternatives have been suggested, from variants of proofs of stake (e.g., [3], [13]) to various proofs of "something" (e.g., proof of Exercise [12]). Other attempts to tackle the huge waste generated by the Proof of Work as implemented in Bitcoin have been aiming at turning them into work that is actually useful (e.g., solving practical problems have been investigated in [2]) but not really in reducing the amount of work. The main issue is that it is still paramount to reach agreement and prevent double spending while providing a way to mint the currency.

In this paper, we mainly focus on the unsustainable energy cost and propose a cryptocurrency protocol moderated through a "green" mining protocol in such a way that it will remove the high energy cost of the proof of work while keeping the blockchain mining input rate between reasonable limits. Those limits can be arbitrarily tuned *a priori*, while the proof of work CPU parameter must be constantly updated.

Our scheme is inspired by a "green" leader election [8]. At its core our model implements a Leader Election that (in addition to public verifiability) provides liveness and fairness: the unique chain grows and the probability of electing each party is independent to its relative computational power. Our model is "permissionless": there is no explicit membership protocol, anyone can join the system. Our model also offers the safety property (block are eventually committed and cannot be removed).

We present our "green" protocol in Section II and its performance analysis in Section III.

## II. The Green Coin mining protocol

### A. Parameters and format

Our Green Coin protocol is based on the following parameters:

- an integer $k$;
- a vector of increasing probabilities of length $k+1$: $P_0, P_1, \ldots, P_k$, with $P_k = 1$;
- an upper bound $N$ on the maximum admissible number of contending block, it can be set as large as possible.

Typical values for these parameters would be $k = 8$, $N = 2^{32}$ are examples. The green coin leader protocol consists in authorizing the mining of an average number of order $N^{1/k}$ out of $n \leq N$ contending blocks.

As with Bitcoin, each block of the Green Coin protocol will carry some values obtained by hash function. We denote $h$ the length of the hash value fields in the block. A typical value is $h = 256$ (32 bytes).

Contrary to Bitcoin where there is only one kind of block, Green Coin relies on two kinds of blocks:

- the *regular* block containing transactions;
- the *empty* block which does not contain any transactions.

The regular block format is similar with Bitcoin block format but with the difference that it does not contain a *nounce* field but instead a *call* field. The call field is similar to a hash value field. The following table gives an abstract format listing the most significant fields:

| | |
|---|---|
| 1 | hash of previous block |
| 2 | date |
| 3 | list of transactions |
| ... | ... |
| 4 | call value |
| 5 | hash of the block |

The empty block looks similar but the transaction list is replaced by a nounce field of same size as a hash value field.

| | |
|---|---|
| 1 | hash of previous block |
| 2 | date |
| 3 | nounce value |
| 4 | call value |
| 5 | hash of the block |

### B. Mining protocol

Initially, to simplify our presentation, we will assume that empty slots are mined by a central entity and on a restricted depositary. We will discuss later how the protocol can be made distributed or how to face forgery.

A block can be mined only if its hash value is smaller than the call value of the previous block. The call field of a regular block has always the same value which is $2^{h+1}P_0$. If no regular block is mined, after a certain lapse of time the central entity mines an empty block. The empty block hash value should not necessarily be smaller than the call value of the previous block, but any regular block should. The call value of the empty block should have value $2^{h+1}P_1$. The use of the nounce field will be described later.

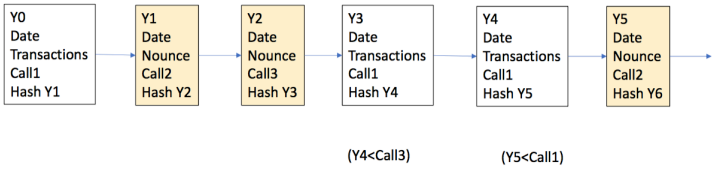


Fig. 1: A sequence of blocks, empty blocks are grayed.

As long as no new regular block is mined, the central entity mines empty block with successive call values $2^{h+1}P_2, \ldots, 2^{h+1}P_k$. After $k$ mined empty blocks the call values sequence restarts.

As there is no nounce field in the regular block there is no proof of work computation required. The block are authorized via the call values which have the effect of regulating the mining rate. Since the non authorized blocks do not need to be submitted there is also a regulation of the traffic generated by the block mining.

If a call fails, *i.e.* no regular block is mined, then a new call will be made via a new empty block. Since $P_k = 1$, the last call value is the max-value $2^{h+1}$ (in fact $2^{h+1} - 1$, otherwise it would exceed the field length), which means that every block is eligible (as long as it contains the hash value of the last empty block). Therefore any competition of block mining will result in at least one block mined. Figure 1 displays an example of block sequences in the blockchain. The chain is enforced by the hash values $Y_0, Y_1, \ldots$. The value of a regular block hash should always be smaller than the call value of the previous block, regardless it is empty or regular. Empty blocks do not need to follow this rule.

### C. Distributed empty block mining

The main issue around the mining of empty block is that other user could be tempted to mine empty blocks. Having several parallel empty blocks is not a problem *per se* since, without nounce, they will be identical. But a user may like to accelerate the protocol in order to get to the call value $2^{h+1} - 1$ which allows to mines blocks without restriction. Notice that in this case all regular blocks in competition would be simultaneously eligible and letting them go and mine at the same time would not give a competitive advantage to those who caused this. It should be noted that the empty block itself, as carrying an empty list of transactions, provides no monetary advantage to those who mine it.

Anyhow an excessive flow of blocks to be mined may cause a congestion problem which is exactly the proof of work concept that we want to solve. Of course the most suitable way to fight this syndrome is to give the authority to a single entity the right of mining empty blocks and store them on a single fully secure depository.

But a single authority as secure as possible is vulnerable to cloning, hacking or hijacking. For this reason we propose to introduce a nounce field in the format of the empty block. The empty block miner would need to produce a proof of

work which will delay the time last between two empty block mining. That way a secondary forged empty block miner will be still delayed by the proof of work. This is a safe solution when we consider that the acceleration of empty block mining is the most dangerous threat of the present scheme.

## III. Performance analysis

### A. The classic leader election via collision

The goal of leader election (e.g., [7], [11]) is to select one among $n > 0$ players, by proceeding through a number of rounds. Note that using rounds is natural here as it is also commonly accepted that the Blockchain model incorporates some degree of synchrony assumptions, otherwise if network delay can be unbounded then the adversary can boost its power just by making the non-faulty parties incur much higher delays [4].

The rounds of our leader election are made via coin tossing. We assume that all competitors has a coin whose head probability is $p$ and tail probability $q = 1 - p$. We want to proceed to a leader election among $n \geq 1$ competitors. At the beginning all the $n$ competitors transmit. If it results into a collision then each competitor tosses the coin, only those who get a head contend for the second round, we call them the survivors. If the second round results in a collision, then a second coin tossing occurs and a subset of survivors contend, and the protocol continues until none survive. In this case we take as result the last non empty set of survivors.

There is an additional procedure in order to reduce the survivor population in order to get a single leader. In general it is a collision algorithm but it is not useful here because we allow a subset of survivors as long as this subset is of reasonable size.

In fact the average size of the last non empty survivor set can be computed precisely using analytic combinatorics to be $\frac{q}{p \log(1/p)}$ with some fluctuations of small amplitude. We notice that this value is finite and independent of the initial number contenders. This average size can also be tuned to be small or large by tuning the parameter $p$. It can be done by compromising with the average number of rounds which tends to $\log_{1/p} n$ when $n \to \infty$. If we wanted to imitate this process with the call values in the empty block one should consider a sequence $(P_0, P_1, \ldots)$ such that $P_\ell = p^\ell$ represents an exponential descending staircase. However this will not work as expected since after the first call one would have already $n$ mined blocks. Still we can be inspired by this protocol in order to design the ascending exponential staircase. This is indeed made possible because we can safely assume that there is a large yet fixed limit $N$ to the number of simultaneous contenders.

In this case we suppose that $P_0 = \frac{1}{N}$ and that $P_\ell = \frac{1}{N} p^{-\ell}$ for $\ell \leq k$. In order to have $P_k = 1$ one must have $p = \frac{1}{N^{1/k}}$. If $N = 2^{32}$ and $k = 8$ we will have $p = \frac{1}{16}$ and the call sequence will be:

| call value rank | value |
|---|---|
| 0 (initial) | $2^{224} - 1$ |
| 1 | $2^{228} - 1$ |
| 2 | $2^{232} - 1$ |
| 3 | $2^{236} - 1$ |
| 4 | $2^{240} - 1$ |
| 5 | $2^{244} - 1$ |
| 6 | $2^{248} - 1$ |
| 7 | $2^{232} - 1$ |
| 8 | $2^{256} - 1$ |

We denote $\mathbf{M}_n$ the average number of regular mined block when $n$ blocks are in competition after a regular block. An important but classic assumption in this paper is that successive calls of a hashing function are independent. Of course *stricto sensu* this is not true since hash value determinations are deterministic computations, but traditionally the better successive hash values imitates independent random variables, the better is a hash function. In fact this argument is the foundation of the resilience of blockchain made of successive hash computation against attack.

Let $\mathbf{M}_n^\ell$ denote the number of blocks mined after the $\ell$th empty block under the condition that all the previous empty blocks resulted into no regular block mined. We have $\mathbf{M}_n = \mathbf{M}_n^0$ and since $P_k = 1$: $\mathbf{M}_n^k = n$. Since the fact that the successive calls to hash values are assumed independent, the number of blocks called by the $\ell$th empty block is a binomial random variable $B(n, P_\ell)$ of probability $P_\ell$. For $\ell < k$ when the binomial variable is $B(n, P_\ell) = 0$ which occurs with probability $(1 - P_\ell)^n$, no block is mined after the $\ell$th block and $\mathbf{M}_n^\ell = \mathbf{M}_n^{\ell-1}$. Thus for all integers $m$, $m > 0$:

$$\begin{aligned} P(\mathbf{M}_n^\ell = m) &= \binom{n}{m} P_\ell^m (1 - P\ell)^{n-m} \\ &\quad + (1 - P_\ell)^n P(\mathbf{M}_n^{\ell+1} = m) \end{aligned} \tag{1}$$

and

$$P(\mathbf{M}_n^\ell = 0) = \delta(n). \tag{2}$$

where $\delta(n) = 1$ if $n = 0$ and zero otherwise. Let $M_n^\ell(u) = E[u^{\mathbf{M}_n^\ell}]$.

*Lemma 1:* We have for $\ell < k$

$$\begin{aligned} M_n^\ell(u) &= (1 + P_\ell(u - 1))^n \\ &\quad + (1 - P_\ell)^n M_n^{\ell+1}(u) \end{aligned}$$

and $M_n^k(u) = u^n$.

*Proof:* This is a direct application of equation (1) ∎

*Lemma 2:* We have the identity

$$E[\mathbf{M}_n] = nP_0 + \sum_{\ell=1}^{\ell=k} nP_\ell \prod_{j<\ell} (1 - P_j)^n \tag{3}$$

*Proof:* This is a direct application of previous lemma by using $E[u^{\mathbf{M}_n^\ell}] = \frac{\partial}{\partial u} M_n^\ell(1)$ ∎

*Theorem 1:* We have the estimate $E[\mathbf{M}_n] = O(N^{1/k})$ for all $n \leq N$.

*Remark:* the optimal value is $k = O(\log N)$ but lower values are also interesting.

*Proof:* From equation 3 we get the inequalities

$$E[\mathbf{M}_n] \leq nP_0 + \sum_{\ell=1}^{k} nP_\ell (1 - P_{\ell-1})^n \quad (4)$$

$$\leq nP_0 + \sum_{1 \leq \ell \leq k} nP_\ell \exp(-P_{\ell-1} n). \quad (5)$$

The quantity $\sum_{1 \leq \ell \leq k} nP_\ell \exp(-P_{\ell-1} n)$ is equal to $\sum_{1 \leq \ell \leq k} \frac{n}{N} p^\ell \exp(-p^{\ell-1} n/N)$. This quantity is smaller than $f(n/N)$ with $f(x) = \frac{1}{p} \sum_{\ell \in \mathbb{Z}} g(p^\ell x)$ with $g(x) = xe^{-x}$. The function $f(e^x)$ is periodic of period $\log p$ and therefore is bounded. Since $p = N^{-1/k}$ we get $E[\mathbf{M}_n] = O(N^{1/k})$ since $n/N \leq 1$ because $n \leq N$. ■

*Theorem 2:* We have the more precise estimate

$$E[\mathbf{M}_n] \leq \frac{n}{N} + AN^{1/k}. \quad (6)$$

with $A = \frac{1}{\log(1/p)} \sum_{k \in \mathbb{Z}} |\Gamma(1 + 2ik\pi/\log p)|$, $p$ denoting $N^{-1/k}$ and where $\Gamma(.)$ is the Euler "Gamma" function.

*Proof:* We can give an accurate estimate of the maximum value of function $f(x)$. The function $f(e^x)$ is periodic of period $\log p$ and therefore has a Fourier decomposition:

$$f(e^x) = \sum_{k \in} g_k e^{2ik\pi x/\log p} \quad (7)$$

where $g_k$ is the Fourier coefficient.

We have

$$g_k = \int_{\log p}^{0} f(e^x) e^{2ik\pi/\log p} dx \quad (8)$$

$$= \frac{1}{\log(1/p)} \int_0^\infty g(x)^{2ik\pi/\log p - 1} dx \quad (9)$$

$$= \frac{1}{\log(1/p)} \Gamma(1 + 2ik\pi/\log p) \quad (10)$$

and then use the upper bound $|f(e^x)| \leq \sum_{k \in \mathbb{Z}} |g_k|$.

In passing we get the mean value of $f(e^x)$ to be equal to $g_0 = \frac{1}{\log(1/p)}$. ■

Figure 2 shows the quantity $E[\mathbf{M}_n]$ versus $n$ for various parameters. Notice that the sequence of bumps reflects the periodic fluctuations as function of $\log n$ analyzed in the upper bound. Figure 3 shows the same quantities obtained by simulation, each point being simulated 1,000 times.

We can give a close expression of the distribution of $\mathbf{M}_n$.

*Lemma 3:* For all complex number $u$:

$$E[u^{\mathbf{M}_n}] = (1 + P_0(u-1))^n - (1 - P_0)^n + \sum_{0 < \ell < k} \prod_{j < \ell} (1 - P_j)^n \times ((1 + P_0(u-1))^n - (1 - P_0)^n) + u^n \prod_{j < k} (1 - P_j)^n.$$

*Proof:* By application of lemma 1. ■

Figure 4 displays the various shape of the sequences $P(\mathbf{M}_n > m)$ versus integer $m$ for various parameters. Notice that the

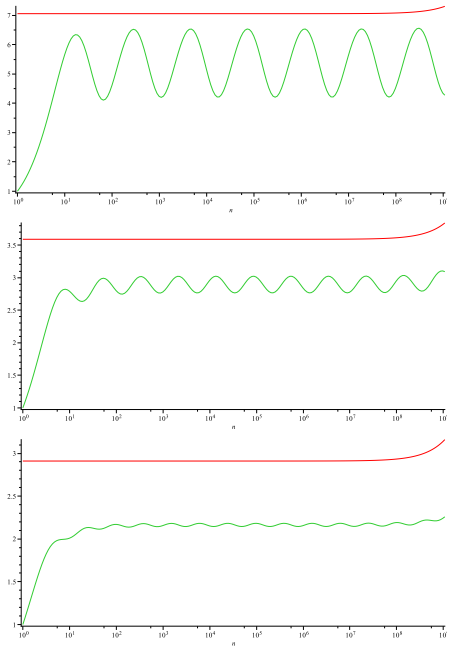

Fig. 2: The quantity $E[\mathbf{M}_n]$ and the upper bound given by (6) in red versus $n$ for $N = 2^{32}$, (top) for $k = 8$, (middle) $k = 12$, (bottom) $k = 16$.

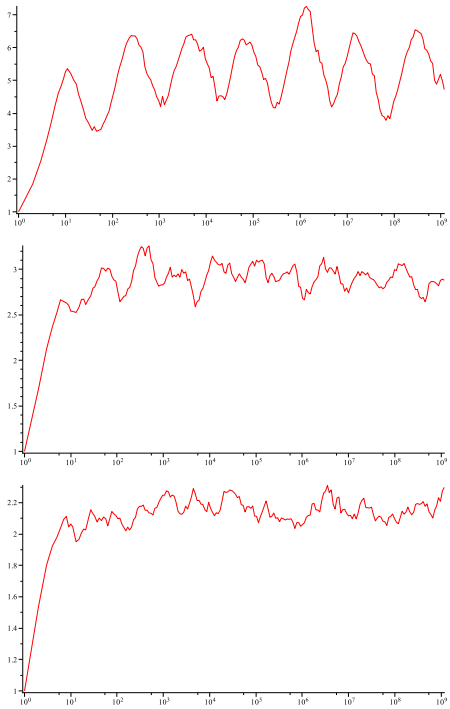

Fig. 3: The same as in figure 2 $E[\mathbf{M}_n]$ with $N = 2^{32}$ but obtained with 1,000 simulations per point, (top) for $k = 8$, (middle) $k = 12$, (bottom) $k = 16$.

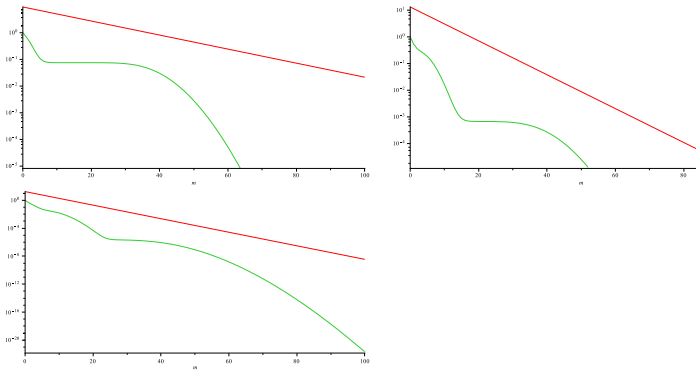

Fig. 4: Computed values of $P(\mathbf{M}_n > m)$ with upperbound given by (11) in red versus $m$ for $n = 10,000$, with $N = 2^{32}$, (left) for $k = 8$, (right) $k = 12$, (bottom) $k = 16$.

sequence of descending bumps in the plots comes from the mixture of the different binomial distributions. We can also give an estimate of the exponential tail distribution of $\mathbf{M}_n$ by the following theorem about large deviations.

*Theorem 3:* Let $p = N^{-1/k}$. For all integer $m > 0$, we have the estimate

$$P(\mathbf{M}_n > m) \leq \frac{k + e^p}{(1+p)^m} \tag{11}$$

*Proof:* We have $P_\ell = p^{k-\ell}$ and

$$\begin{aligned} E[u^{\mathbf{M}_n}] \quad \leq \quad & (1+p^k(u-1))^n - (1-p^k)^n + \\ & \sum_{0<\ell<k} \left( \left(1+p^\ell(u-1)\right)^n - (1-p^\ell)^n \right) (1-p^{\ell+1})^n \\ & + u^n (1-p)^n \end{aligned}$$

Let $u = 1 + p$ we have $1 + (u-1)p^\ell = 1 + p^{\ell+1}$ and therefore $(1 + (u-1)p^\ell)(1 - p^{\ell+1}) = 1 - p^{2(\ell+1)} \leq 1$. Thus $E[u^{\mathbf{M}_n}] \leq (1 + p^{k+1})^n + k$. Since $(1 + p^{k+1})^n \leq \exp(np^{k+1}) = \exp(pn/N)$. Since $n < N$ we have $\exp(np/N) \leq \exp(N^{-1/k})$. We conclude the proof with the observation that for all integers $m$: $u^m P(\mathbf{M}_n > m) \leq E[u^{\mathbf{M}_n}]$ ■

Figure 5 displays the histograms of 1,000 simulations of $\mathbf{M}_n$ versus $n$ for $N = 2^{32}$ and $k = 8, 12, 16$.

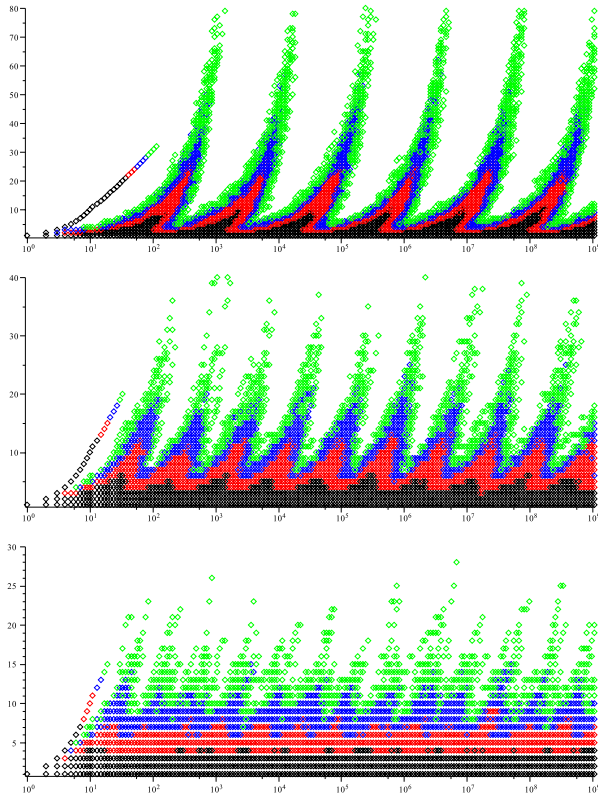

Fig. 5: various values of $\mathbf{M}_n$ versus $n$ simulated 1000 times black values appeared more than 64 times, red more than 16 times, blue more than 4 times, green more than once, with $N = 2^{32}$, (top) for $k = 8$, (middle) $k = 12$, (bottom) $k = 16$.


## ACKNOWLEDGMENT

We are indebted to Daniel Augot for earlier discussions on Blockchain systems.


## REFERENCES


[1] James Aspnes, Collin Jackson, and Arvind Krishnamurthy, *Exposing computationally-challenged Byzantine impostors*, Technical Report YALEU/DCS/TR-1332, Yale University Department of Computer Science, July 2005.

[2] Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. *Proofs of useful work*, 2017.

[3] Iddo Bentov, Charles Lee, Alex Mizrahi, and Meni Rosenfeld, *Proof of activity: Extending bitcoin's proof of work via proof of stake*, ACM SIGMETRICS Performance Evaluation Review, 42(3):34–37, 2014.

[4] Christian Decker and Roger Wattenhofer, *Information Propagation in the Bitcoin Network*, P2P, 13th IEEE International Conference on Peer-to-Peer Computing, Trento, Italy, September 2013.

[5] Cynthia Dwork and Moni Naor, *Pricing via processing or combatting junkmail*, CRYPTO, 12th Annual International Cryptology Conference on Advances in Cryptology, pp. 139–147, London, UK, 1993.

[6] Juan Garay and Leonardos Kiayias, *The bitcoin backbone protocol: Analysis and applications*, EUROCRYPT, pp. 281–310, 2015.

[7] P.J. Grabner and H. Prodinger, *Sorting algorithms for broadcast communications: Mathematical analysis*, Theoretical computer science, 289(1), pp. 51-67, 2002.

[8] Philippe Jacquet, Dimitris Milioris and Paul Mühlethaler, *A Novel Energy Efficient Broadcast Leader Election*, Mascot, 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, San Francisco, August, 2013, pp. 495–504.

[9] S. Nakamoto, *Bitcoin: a peer-to-peer electronic cash system,* 2008, Available: http://www.bitcoin.org

[10] Rafael Pass, Lior Seeman, and Abhi Shelat, *Analysis of the blockchain protocol in asynchronous networks*, EUROCRYPT, LNCS 10211, pp. 643– 673, 2017.

[11] Helmut Prodinger and Guy Louchard, *The Asymmetric Leader Election Algorithm with swedish stopping: A probabilistic analysis*, Discrete Mathematics & Theoretical Computer Science 14(2), pp. 91-128, 2012.

[12] Ali Shoker, *Sustainable blockchain through proof of exercise*, NCA, 16th IEEE International Symposium on Network Computing and Applications, Cambridge, 2017, pp. 393-401.

[13] G. Wood, *Ethereum: A secure decentralised generalised transaction ledger*, 2015, yellow paper.