Delft University of Technology, Bachelor Seminar

Stop Boiling the Oceans: A Review on Energy Efficient Proof of Work Alternatives

M.R. Comans, O.N. de Haas, R. Jongerius, D.A.J. Oudejans and E. de Smidt
Supervised by: Z. Erkin and P.R. Zimmermann
Delft University of Technology, Delft, The Netherlands

Abstract

Bitcoin's underlying consensus algorithm, Proof of Work, is of inefficient nature. Due to the sheer size that Bitcoin has grown to over the recent years, power consumption has increased so much that the Bitcoin network has been estimated to consume more power than the whole country of Ireland. This paper investigates several alternatives to the Proof of Work consensus algorithm, with a focus on energy efficiency. We found permissioned and permissionless consensus algorithms that offer solutions that consume significantly less energy than Proof of Work.

Keywords: Blockchain, Bitcoin, Byzantine fault tolerant, Consensus algorithms, Cryptocurrency, Energy cost, permissionless, Proof of Stake, Proof of Work

1 INTRODUCTION

Blockchain is a technology that is being used in increasingly more applications and platforms [1]. The idea of a blockchain was first introduced by Haber and Stornetta in 1990 [2], though it was not until Nakamoto applied blockchain as the underlying data structure of the cryptocurrency Bitcoin in 2009 [3]. Bitcoin made it possible to make transactions without the need for a centralized authority (e.g. a bank). During the years that followed, there has been an enormous surge of new cryptocurrencies. At the time of writing, over 2000 different cryptocurrencies are listed on CoinMarketcap.com, a website that gathers cryptocurrency market data from a large selection of exchanges [4].

A downside to Bitcoin is that its system for reaching consensus over its transactions, known as Proof of Work, requires maintainers of the blockchain to put a lot of computational work towards finding a solution to a hard mathematical problem. These maintainers, also known as miners, compete to guess the solution since the first to find it is rewarded. However, this means that the other people who made an attempt essentially wasted all of their computing power put towards solving the problem. In order to maximize their chances, miners guess as fast as possible, which consumes a considerable amount of energy. In Bitcoin, everyone is free to become a miner of the network and the difficulty of the problem, which in turn means the amount of computational work required, scales along with the number of miners participating.

The mining reward and the accessibility of the network have lead to an increase in the number of miners and, thus, the power consumption of the network, which is presented in Figure 1 [5]. In May 2018, the entire Bitcoin network has been estimated to use as much energy as the country of Ireland [5]. Even if all Bitcoin transactions would be processed using hardware running on renewable energy, the network would not be able to sustain itself. A higher amount of transactions would cost a tremendous amount of energy and renewable energy is seasonal while transactions take place year-round [6]. Compared to VISA, the Bitcoin network uses about 300000 times more energy per transaction [7]. Furthermore, most Bitcoin miners use specialized hardware to mine. These custom mining machines cannot be repurposed after their lifespan since they are hardwired to mine Bitcoin [6]. This approach is not sustainable with the looming dangers of climate change. It becomes evident that Proof of Work, and by extension Bitcoin, is not energy efficient and an alternative should be found to replace it.

The main question that this paper tries to answer is: What consensus algorithms that currently exist are more energy efficient alternatives to Proof of Work? To answer this question, the paper surveys what the most commonly used consensus algorithms are among major cryptocurrencies. The process of this study consisted of three phases. First of all, before exploring and comparing the various consensus algorithms, a proper understanding of blockchain technology was required. In order to achieve this, the fundamental papers on subjects such as blockchain and Bitcoin were studied, including the original Bitcoin whitepaper [3] and the paper which initially introduced blockchain as a concept [2].

Secondly, a list of consensus algorithms to be discussed in this paper had to be established. The surveyed consensus algorithms were the ones applied by the 20 most used cryptocurrencies, based on market capitaliza-

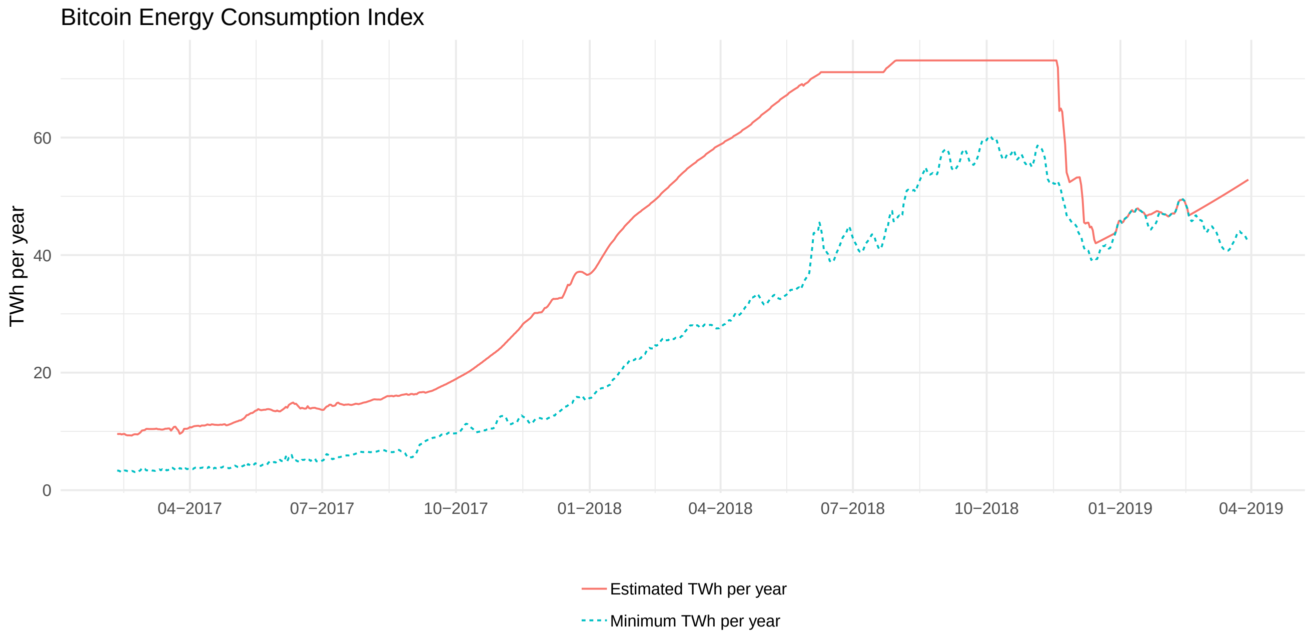

Figure 1: Bitcoin energy consumption index [5]

tion on CoinMarketCap.com [4]. *Tokens* were excluded from this list since these cryptocurrencies do not implement their own blockchain, but instead make use of the infrastructure of others. An overview of these cryptocurrencies as of 31 March 2019 along with their consensus algorithms is given in Table 1.

Finally, the selected consensus algorithms were explored in-depth. Some of the consensus algorithms are very new or in active development, which was clearly recognizable in the poor or rapidly changing documentation. Other channels through which literature was gathered consist of the TU Delft Library's WorldCat Discovery and Google Scholar. Keywords which were used include: the names of the consensus algorithms (Proof of Work, Proof of Stake, et cetera), blockchain, consensus algorithm, cryptocurrency, energy/power usage, attack.

2 AN OVERVIEW OF BLOCKCHAIN

In order to have a virtual currency that can hold value, it should be infeasible to spend the same money twice, just like how it is impossible to spend physical cash twice. Since virtual currency is just data that can be copied or falsified, guaranteeing that money cannot be spent twice is deemed to be a problem. This problem is known as the double spending problem [8]. Traditional digital payment methods solve the double spending problem by having a third party to verify the validity of the transaction. The third party keeps a ledger of all transactions and verifies a new transaction by checking the origin of the funds being moved. The validating third party will only accept a new transaction if the funds were not already spent. This solution works and has done so for hundreds of years. However, it has been argued that this system has its weaknesses. One of the most predominant arguments against this system is that it requires the users to trust the authority verifying the transactions. Moreover, the mediation of third parties could increase transaction costs. Lastly, there is no possibility to make non-reversible transactions, which can be a risk for the parties involved in a transaction. To circumvent trust in a single validating authority, a solution has been presented where each user of the network maintains there own copy of the full ledger. This ledger could then be synchronized with the ledgers of other users using a peer-to-peer synchronization protocol. This would result in a network where trust is spread over the users of the network, instead of trusting a single authority [3].

The transactions of Bitcoin and most other cryptocurrencies are stored on a decentralized, digital ledger. This ledger, called a blockchain, maintains a full history of every transaction made on the network. The blockchain got its name from the fact that transactions are grouped in blocks, with each block referencing the previous block, resulting in a chronological chain of blocks. Figure 2 shows an illustrative representation of a simple blockchain. The blockchain is append-only, which means that new blocks can only be added at the end of the chain. The new block contains a reference to the last block in the form of a hash of that last block. Doing this prevents blocks from being inserted somewhere in the middle of the chain: if a block would be inserted in the middle, than the next block in the chain would still reference the original block and not the newly inserted block.

The balance of a user is not directly stored on the blockchain. Instead, the balance is represented as un-

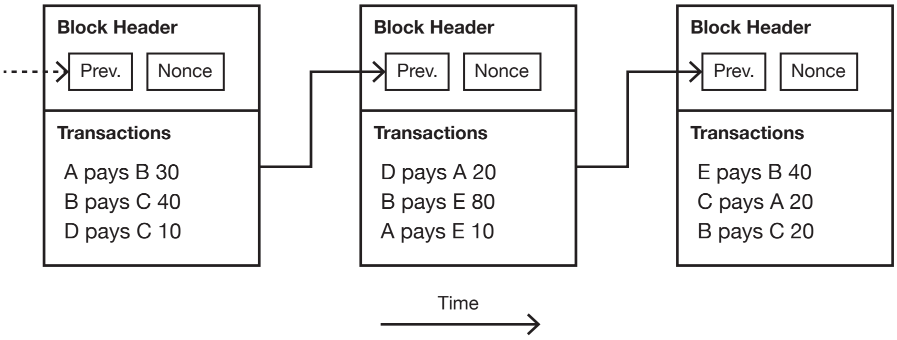

Figure 2: A simple illustrative representation of a basic blockchain

spent transactions which is referred to as unspent transaction output (UTXO). A new transaction made by a user on the blockchain refers to old transactions made to that user in order to determine where the funds come from. A transaction can only be referenced once and should be spent in its entirety, which prevents any double spending. To ensure that only the owner of the UTXO can spend them, every user signs a new outgoing transaction with their private key. This private key originates from a cryptographic public/private key pair. The signature can then be verified by others in the network using the user's public key.

There is one more problem: how do nodes agree on the blocks that should be added to the chain? For example, an attacker may spend his coins in an online shop, but subsequently broadcast a fraudulent version of the blockchain without said transaction, while the shop will broadcast the *correct* blockchain. Since there is no single validator to trust, the nodes in the network cannot know which version is correct. In other words, a consensus has to be reached by the network to decide on the blockchain that the network believes to be the *correct* chain.

In Bitcoin, this consensus is reached using the Proof of Work (PoW) consensus algorithm. This algorithm will be explained more thoroughly in section 3.1.1, but essentially new blocks are created by solving a highly computationally difficult problem, which ensures the validity of all transactions inside the block. The chance one solves the problem is directly depending on the amount of computing power available in the network. Therefore, PoW provides security as long as one party does not control 51% of the total computing power in the network. Due to the computational investment in the blockchain, it is infeasible to modify or delete blocks from the blockchain, since this requires redoing all of the work being done on all succeeding blocks. This is another important property of blockchain in combination with Proof of Work: immutability. However, there is one considerable downside to this approach: because it has to be computationally infeasible for one entity to control the majority of the network, the problem to solve should require a lot of work.

A last important property of a blockchain is whether it is permissioned or permissionless. Permissionless blockchains are completely open to participate in and no permission is required from anyone to do any operation on the blockchain such as mining. No one knows the identities of others in the network, and trust is purely based on game-theoretical incentives. Permissioned blockchains, on the other hand, require permission to participate in the network. This removes the need for introducing artificial incentives since nodes can only be added to the network if they are trusted to vote honestly [9].

3 CONSENSUS ALGORITHMS

Consensus algorithms are used by cryptocurrencies to decide on the state and order of transactions in a network of nodes. There exist two main types of consensus algorithms in our selection. The first is proof-based consensus algorithms. The second type is Byzantine fault tolerance-based consensus algorithms.

3.1 Proof-based algorithms

In proof based consensus algorithms a user has to prove that the user can produce a new block. The proof should be verifiable by all other nodes in the network and it should be hard to fake a proof. This section will provide the proof-based algorithms from Table 1 and two more algorithms, Proof of Elapsed Time and Proof of Capacity, since these have working implementations.

3.1.1 Proof of Work

One of the most (in)famous consensus algorithms is Proof of Work (PoW). It is the most used consensus algorithm in the top 20 coins by market capitalization (see Table 1). The first cryptocurrency that used PoW to decide over the order and state of transactions in a blockchain was Bitcoin [3]. In PoW, blocks in the blockchain can be created by performing computationally expensive calculations.

The incentive for creating blocks is the block reward. A node in the network receives the reward if it is the first to solve a mathematical "puzzle". The goal of this puzzle is to find a number called the nonce. Nodes that try to solve this puzzle are called miners. The miners have to find a hash of the block which, when represented

Table 1: Top 20 coins and their consensus algorithms according to coinmarketcap.com on March 31, 2019

#	Cryptocurrency	Consensus Algorithm	Notes
1	Bitcoin	Proof of Work [3]	
2	Ethereum	Proof of Work [10]	Planning to use Proof of Stake to finalize the Proof of Work blockchain [11].
3	XRP	Byzantine Fault Tolerance [12, Chapter 6]	Formerly known as Ripple.
4	EOS	Delegated Proof of Stake & Byzantine Fault Tolerance [13]	
5	Litecoin	Proof of Work [12, Chapter 6]	
6	Bitcoin Cash	Proof of Work	
7	Stellar	Federated Byzantine Agreement [14]	
8	Cardano	Proof of Stake [15]	
9	TRON	Delegated Proof of Stake [16]	
10	Bitcoin SV	Proof of Work [3]	
11	Dash	Proof of Work [12, Chapter 6]	
12	Monero	Proof of Work [12, Chapter 6]	
13	IOTA	Directed Acyclic Graphs [17]	DAGs are not discussed in this paper.
14	Tezos	Proof of Stake [18]	
15	NEO	Delegated Byzantine Fault Tolerance [19]	
16	Ontology	Verifiable Byzantine Fault Tolerance [20]	Combination of PoS, BFA and Verifiable Random Functions (VRF). VBFT is not discussed in this paper.
17	Ethereum Classic	Proof of Work [12, Chapter 6]	
18	NEM	Proof of Importance [12, Chapter 6]	
19	Zcash	Proof of Work [12, Chapter 6]	
20	VeChain	Proof of Authority [21]	

in binary, starts with a certain amount of zeros. In other words, the hash has to be smaller than a certain target value. The nonce is part of the block and is thus hashed together with the other contents of the block. If the hashed value is not below the target value, the miners change the nonce and try again. Due to the properties of a hashing algorithm, even if this nonce changes slightly, the resulting hash changes drastically. The output can be seen as a sequence of coin flips. The chance to get one zero at the beginning of the hash is 50%, to have two consecutive zeros 25% and so on. The more zeros are required, the more challenging the puzzle is. The process of solving this puzzle is called mining. This process can also be seen as a lottery between the miners. Once the miner has found a particular nonce so that the hash of the block is below the target, the miner can broadcast the block to the network, after which everyone can check if the nonce found by the miner is valid, by calculating the hash themselves. If the nonce is valid, the other nodes can decide to add the found block to their local blockchain and the process of finding a block starts over. The miner gets a reward when a block has been found, which incentivizes people to mine blocks and do the computational work.

The core idea of Proof of Work relies on the fact that a substantial amount of work has been put into a block. The consensus of the correct blockchain can be reached by trusting the longest chain because this chain will have the most work put into it. this is also where the name PoW comes from. When two nodes mine a block at the same time, there will be a fork in the blockchain. Eventually, one of these two forked chains will become longer and will be accepted as the correct chain. As long as the majority of the computing power in the network is held by honest nodes, their chain will remain the longest. This also prevents bad actors from double spending or making dishonest transactions, as they would have to maintain their malicious blockchain on their own and compete with honest nodes. As long as one party does not control the majority of the computational power of the network, so more than half, sustaining a transaction where double spending is performed is infeasible. Another advantage of PoW is that every block references the block that comes before it. This ensures that the content of blocks that are already in the blockchain cannot easily be altered: if the contents of a block somewhere in the blockchain changes, all of the hashes and corresponding nonces of the following blocks would also need to be recalculated. So, if a malicious actor wants to change a block in the past, he would have to find a new nonce for that block up until the current block, all while the rest of the network is mining new blocks. This is why it is infeasible to change a blockchain without having more than half of the computational power in the network, which in turn also makes blocks immutable after enough blocks are appended to the blockchain.

3.1.2 Proof of Stake

Proof of Stake (PoS) was first introduced in a Bitcoin Forum thread as an energy efficient alternative for Proof of Work [22]. Ppcoin (also known as PeerCoin) was the first to implement this system a year later [23]. Reaching consensus on a new block is based partly on one's stake in the system. The main idea behind this is that anyone with a large stake would not want to include fraudulent transactions, as that would essentially jeopardize the chain along with their profits. Since PoS does not require expensive hardware, it can be considered a fairer approach to reaching consensus.

Within Proof of Stake, there are a few common ways to determine who creates a block:

Pure

In a pure Proof of Stake system, the effective balance of a user is used to determine the chance that that user can create the next block. An example is the Nxt platform [24]. Nxt holds a total of 1 billion coins and is initially distributed from a genesis account, that has a negative balance of 1 billion. In order to create a new block, every active user signs the *generation signature* of the current block with their public key. The hash of this signature is compared to a target value. This target value is calculated for every user based on a base value, the time since the last block and the effective balance of the user. This is energy efficient, since every user can only generate one hash, and this hash will be compared to the target value, which changes only every second. Once the target for a user has become so large that their calculated hash is below it, they can append the block to the blockchain. Contrary to PoW, PoS algorithms usually do not include mining rewards, and instead, the incentive comes from transaction fees that the block creator can collect.

Coin age

Coin age is defined as the multiplication of the amount of currency and the holding period [23]. For example: if someone received 2 coins and held them for 50 days, the coin age accumulated by that person is 100. If that person chooses to spend the 2 coins after the 50 days, the coin age accumulated by that person is consumed. In PeerCoin, the probability that a node can create the next block is determined using the coin age of that user[23]. Specifically, users can create a hash of the new block only every second, since a timestamp is included in the block header and no nonce is present, and this hash can be compared to a target value that is calculated using the coin age of that user.

Leased

Leased Proof of Stake (LPoS) is a variant of the classic Proof of Stake consensus where nodes that have a balance can lease their balance to other nodes, which improves their chance to be selected to create the next block. In return, the lessee pays a fee to the lessor. An example of a cryptocurrency that utilizes this algorithm is the Waves Platform [25].

Slashing

Ethereum introduced their Proof of Stake algorithm, Casper, as an extra security layer on top of the current Proof of Work blockchain [11]. Casper is not meant to provide consensus over the individual blocks, but rather to provide consensus over the finality of checkpoints within the blockchain. In Casper, each validator deposits coins as stake and votes on the correct next checkpoint. However, the deposit of validators can be lost when they vote for checkpoints in multiple forks. This penalty system is called slashing.

Delegated

In Delegated Proof of Stake (DPoS), users in the network can vote for block producers, which create and verify blocks in the blockchain [16]. The voting power of the users is proportional to the amount of blockchain currency they hold. In a round-robin order, each of the elected producers will create a block once, while the other producers vote on the validity of the block. After all elected producers have created a block once, new producers will be elected.

Current Proof of Stake implementations do have certain issues, highlighted in an article by Bentov, Gabizon, and Mizrahi [26]. One flaw that they mention, is that Proof of Stake is prone to *rational forks*. If a stakeholder in the system is rational, it will be inclined to maintain and solve blocks on multiple forked chains, which would ultimately result in a divergent network. In other words, consensus will never be reached. Ethereum Casper prevents this by slashing the deposit of a validator when they vote for multiple forks [11].

A 51% attack requires notably more resources in PoS than in PoW. It is harder to acquire more than 50% of the stake in the network than computing power, as buying more stake becomes more expensive due to demand and supply. Especially in the case of pure PoS, where all the coins are distributed on conception and there are no block rewards. The transaction fees of new blocks have a low return on investment compared to block rewards.

3.1.3 Proof of Importance

Proof of Importance (PoI) is similar to Proof of Stake (PoS), except that it is not solely derived from the size of an account's balance [27]. It incorporates other behaviors that are believed to be positive for the holistic economy. NEM, the only cryptocurrency in the top 20 (see Table 1) which currently uses PoI, does this by accounting for three factors: vesting, transaction partners, and number of and size of transactions in the last 30 days. By using this approach, NEM avoids the incentive to simply hoard coins, which in PoS systems can be worthwhile.

3.1.4 Proof of Authority

Proof of Authority is comparable to Proof of Stake, but instead of coins, nodes have to put their identity at stake. The idea is that when you put your identity at stake, you gain the right to create blocks. However, everything you do is public, both the benefits you gain but also the bad things you do. So not only your identity but also your reputation is at stake, which should incentivize people to maintain the network [28].

For this system to work, a few requirements have to be met according to POA network:

- The identities have to be real and valid, which requires a good validation process.
- The right to become a validator should be hard to obtain in order for it to be valued.
- Establishing the authority needs to be done using the same procedure for all validators in order to ensure integrity.

Examples of cryptocurrencies that use this consensus algorithm are VeChain [21] and POA Network [28]. In Proof of Authority, all nodes have equal rights and equal opportunities to create blocks. In order to ensure security, the node selection process should not be deterministic, but (pseudo-)random instead. Instead of choosing the longest chain like in Proof of Work as the truth, the nodes pick the chain with the highest number of witnesses (in VeChain called *Accumulated Witness Number*) [21].

Proof of Authority is also vulnerable to 51% attacks, but this requires more than half of the nodes in the network to agree on a malicious chain. Another concern is an attack where someone forks the blockchain and then tries to convince everyone in the network to use that chain instead. In VeChain, this problem is solved by implementing a fixed interval between blocks and by again implementing the Accumulated Witness Number. The branch with the highest number of witnesses will be chosen as the truthful branch.

3.1.5 Proof of Elapsed Time

Proof of Elapsed Time (PoET) is a consensus algorithm for permissioned networks developed by Intel [29]. A key aspect of PoET is Intel's Software Guard Extensions (SGX) programming manual. SGX serves as a Trusted Execution Environment (TEE), which allows specific, trusted code to run independently of the application that it runs in. It produces a signed attestation from an application which is rooted in the processor and guarantees that the code has been initialized correctly in a trusted environment. While this feature provides the fundamentals of PoET, it also brings a barrier of entry resulting in the permissioned nature of PoET. PoET is an efficient alternative to PoW that removes the computationally intensive process and replaces it by stochastically electing participants to be the next block creator. Essentially, each node in the network is given a random timer object. If a timer of a node is the first to expire, that node will become the block leader and is allowed to create the next block. The main advantage to PoET is the fact that it is extremely energy efficient, because when the timer is running, the user's CPU can be idle or used for other tasks. There is no need for complicated stake or incentive architecture. However, the disadvantages of PoET include its inherent permissioned nature and reliance on hardware security.

3.1.6 Proof of Capacity

Proof of Capacity (PoC) is currently being used by only one cryptocurrency, Burstcoin. Furthermore, the team behind Burstcoin is also the one developing the model [30]. In essence, instead of calculating a hash with a different nonce as fast as possible like in Proof of Work, PoC involves using the free space on the hard drive to store pre-calculated solutions. The calculation of these solutions is too difficult to do in real-time, especially because the average time between new blocks in Burstcoin is 4 minutes. By having more free storage space available, it is possible to store a larger number of solutions, called plots, resulting in higher chances to have the solution to the most recent block stored. Some plots are faster than others, meaning that simply having a correct solution is not enough. Even though the only current PoC model is based on a Directed Acyclic Graph data structure, it would still be possible on a blockchain implementation which is why it is still mentioned in this paper.

There are some clear advantages to PoC compared to PoW. For one, storage is far more energy efficient than computing power, leading to less energy consumption. Storage is also much more accessible, as most devices have some form of storage, however, there could emerge a new arms race over storage space instead of computational power. Apart from the fact that the technology is still in its early stages, another disadvantage is that the storage used by plots is unavailable for other uses.

3.2 Byzantine Fault Tolerance-based algorithms

The Byzantine Generals problem [31] is a classic problem in distributed computing. The problem, put simply, is that multiple generals of the Byzantine army have, together with their respective legions, surrounded a city. They must decide, in unison, whether to attack or not. If some generals attack without the others following suit, their siege will fail. The generals and their legions are far apart, meaning they have to communicate via messengers. However, when general A sends a messenger to general B he cannot be sure that his messenger arrived, as the enemy could have intercepted him. When the messenger arrives, general B can send a messenger back to tell A that he received the message, however, now this "acknowledge" message is uncertain to arrive. Furthermore, one or more of the generals may be traitors, who will try to confuse the others. There are several parallels which can be drawn between this problem and the consensus problem of a blockchain, which is why some cryptocurrencies have implemented a variation of Byzantine Fault Tolerance to reach consensus [32] [33].

Byzantine Fault Tolerance is the ability of a distributed network to function appropriately, such that the network correctly and consistently reaches consensus despite bad actors either propagating incorrect information or failing to send information at all. The aim is to weaken the influence of malicious nodes in order to prevent full system failure and ensure a correct consensus is made by the honest nodes. There are three main variations of solutions to this problem in use by cryptocurrencies: Practical Byzantine Fault Tolerance (PBFT), Federated Byzantine Agreement (FBA) and Delegated Byzantine Fault Tolerance (DBFT).

3.2.1 Practical Byzantine Fault Tolerance (PBFT)

One of the first solutions to the Byzantine Generals problem was Practical Byzantine Fault Tolerance [33]. The PBFT model works by providing a practical Byzantine state machine replication that accepts Byzantine faults (i.e. malicious nodes) by making the assumption that there are independent node failures and manipulated messages propagated by specific, independent nodes. Essentially, all the nodes in the PBFT network are ordered in a sequence with one primary node, the leader, and the rest as backup nodes. All the nodes are constantly communicating with each other, trying to reach consensus on the state of the network through a majority. For each broadcast, every node has to prove that message came from a specific peer node, but it also needs to verify that the message was not modified during transmission. This is done using a modified implementation of message authentication codes. Traditional signing systems like what Bitcoin uses would be too inefficient because the volume of messages is much larger.

For this model to work, the assumption is made that the number of malicious nodes can never be larger or equal to one-third of the total nodes in the network. With a large n nodes in the network, it becomes increasingly unlikely for more than $n/3$ nodes to be malicious. As long as this is true, the algorithm effectively provides both liveness and safety, which in turn means that eventually the replies received by clients from their requests are correct due to linearizability. Each round of consensus has 4 stages:

1. A client sends a request to the leader node to invoke an operation.
2. The leader node broadcasts the request to the backup nodes.
3. The backup nodes execute the request and send a reply to the client.
4. The client waits for $n/3 + 1$ replies from different nodes with the same result; this is the consensus state.

The requirements for the nodes are that they start in the same state and are deterministic. The final result is that all honest nodes form a consensus on the order of the operations (e.g. transactions) and they either accept it or reject it. The leader is changed every round in a round-robin type competition. If the network suspects the leader node is malicious, it can also be replaced by the next in line if a majority of honest nodes votes for this. The main drawback of PBFT is that it does not scale well. The model only performs efficiently with a small group of validators, because the validator nodes are constantly communicating.

3.2.2 Federated Byzantine Agreement (FBA)

An aspect that most consensus protocols desire is decentralized control, and one could argue that relying on a cryptocurrency that does not use a decentralized consensus protocol is not an improvement over traditional banking systems in terms of trust. An approach to form a consensus that is decentralized based on Byzantine Fault Tolerance is Federated Byzantine Agreement. FBA was introduced by Stellar [14], but based on the original Ripple Consensus Protocol [34]. The Stellar foundation invented the Stellar Consensus Protocol (SCP), a construction for FBA.

In a PBFT system, there is a list of appointed validators, which is often chosen by a central authority. Therefore, even though the validators might not be from one authority, they are selected by one, making it centralized in some sense. FBA takes a different approach by omitting the need for such a central validator list. Instead, in an FBA system, each validator has a set of peer validator nodes that they trust, which is known as a *Quorum Slice*. In a network of validators, such quorum slices will overlap and will result in a *Quorum* where consensus can be reached through linked trust. A consensus is reached iteratively by rounds of voting. In these rounds, transactions that do not reach a certain threshold are discarded and a new round is started with the remaining transactions. Validators continuously attempt to agree on a specific subset of all candidate transactions, until one final subset is agreed on by a supermajority. This will be the set that is added to the ledger or blockchain.

FBA allows for nodes to freely join the network as a validator, but to actually contribute as a validating node in the network, peer nodes need to include it in their quorum slice, making FBA semi-permissioned.

3.2.3 Delegated Byzantine Fault Tolerance

The NEO cryptocurrency uses a consensus algorithm called Delegated Byzantine Fault Tolerance (DBFT) [19]. In this system, the holders of NEO coins can vote for bookkeepers. These bookkeepers vote for new blocks in the blockchain by using a protocol based on Practical Byzantine Fault Tolerance [35].

4 ENERGY EFFICIENCY ANALYSIS

Proof of Work and hybrid approaches using Proof of Work are the only consensus algorithms where mining requires a significant amount of work and thus power usage. Besides the expensive calculations that are required for mining, the majority of effort is wasted, since multiple miners attempt mining a new block, yet only one miner can add his block to the blockchain and claim a reward.

Proof of Stake, however, is significantly more efficient, since miners do not have to hash indefinitely, but in a limited search space in the case of PeerCoin [23] or only once per block in the case of Nxt [24]. As far as known to the authors, there does not exist any estimation of power consumption of Proof of Stake networks. Another advantage is that no special hardware is required for producing new blocks, which results in less wasted electronics. The delegated version of Proof of Stake is even more energy efficient. Only a selected amount of nodes can produce new blocks, therefore, fewer nodes are performing the computations. However, there is an overhead of electing producers. Proof of Importance and Proof of Authority share the same advantages as Proof of Stake, since the only difference is how the user's stake is determined.

Networks that use Byzantine Fault Tolerance-based consensus algorithms also have low theoretical power consumption compared to PoW, because they do not solve a hashing puzzle at all. These networks reach consensus through continuous communication. In PBFT, only a few nodes in the network are validators and have to arrive at a consensus. However, the validators have to attach their signature and Message Authentication Codes to their messages. This approach is efficient for small or private networks. PBFT does not scale as well as proof based algorithms, since the volume of messages would get too large. Permissioned networks, in general, are more energy efficient than permissionless networks since less nodes have to be active and verify the transactions. FBA solves this problem by using Quorum Slices, because each node only has to communicate with a small subset of trusted nodes.

5 DISCUSSION

As discussed in section 3, many different consensus algorithms exist, trying to solve the different issues that arise from other implementations. It is practically impossible to pick a single approach as the best one since the field is still in rapid development. Cryptocurrencies are sometimes even trying to switch consensus algorithms, such as Ethereum [11]. We can discuss some important considerations though in order to aid research on different consensus algorithms and to indicate areas where future work is needed.

One of the main goals of these cryptocurrencies is decentralization. An interesting question is whether decentralization really is the goal which will solve all of the problems with current centralized payment systems. One can also ask if true decentralization is even a realistic possibility. In the case of Bitcoin, there is no centralized authority in the pure sense: anyone with the appropriate hardware can become a miner. The vast majority of blocks are mined by large mining pools though, with the four largest pools dominating well over 51% threshold [36]. Even if 51% control over the network is never reached, it still means that the large mining pools have significant control over the network, which is exactly what Bitcoin originally wanted to avoid.

This closely relates to the question of whether a cryptocurrency network should be fully permissionless. We would argue that, to some degree, a permissioned network is an interesting solution that can solve a lot of problems that we are facing with current solutions. The energy usage of permissioned blockchains is low as already indicated in section 4. Permissioned networks also have the benefit of being faster, since only some selected nodes have to vote [9]. Apart from the reduced energy consumption and the higher speed, permissioned networks also have a clear governance structure [37]. Updates to the network can be rapidly rolled out opposed to permissionless networks where every node essentially acts in their own self-interest. Nevertheless, permissioned networks also have problems. It is a more centralized approach than a purely permissionless network. This can also impact the security of a permissioned blockchain. Although taking control of the blockchain is not possible for an arbitrary malicious person, nodes can still collude to change the blockchain. However, we would argue that distributing the control over a number of trusted, independent, varied organizations, which are unlikely to collude, would result in a network more decentralized than many of the current networks in Table 1. These organizations do not have to consist of only banks and financial institutions, but could also include organizations with different interest like human rights organizations or the United Nations. This would also help combat the other main issue with permissioned networks: censorship [37]. If the nodes in the network are carefully selected to represent the needs of all people that have an interest in the network, we think this risk is negligible.

In our opinion, a particularly interesting permission system is what we referred to in 3.2.2 as semi-permissionless. This is for example implemented by Stellar, which allows everyone to join the network, but trusted nodes have to start trusting new nodes in the network. We recommend further research to be conducted on the different ways of implementing (semi-)permissioned systems, while possibly considering the inclusion of various carefully selected organizations.

Other further research to investigate could be whether or not Bitcoin users are even aware of the scale of the energy consumption of Proof of Work. Furthermore, it is important to know if users would actually adopt another cryptocurrency that would have objectively better mechanisms if presented, or would stay with Bitcoin. It is thought that most of the people utilizing the cryptocurrency market are primarily interested in investing, rather than using an alternative transaction system [38]. To answer these questions, one should study researches from a more psychological viewpoint.

To form a well-substantiated decision on which consensus algorithm suits the situation best, more factors should be taken into account. Centralization is important to consider. This mostly comes down to personal preference. Full decentralization is often desired by those with a more libertarian opinion, yet plenty of people might rather prefer some governance over the network. This paper did not extensively review security risks, as that would require a separate study to sufficiently analyze the differences among consensus algorithms. There is a lack of research in the security aspects, since all these technologies are state-of-the-art. Some other more practical aspects are transaction rate and confirmation time, which are also important if it were to be used as a replacement for traditional electronic payment.

Further research is necessary in the power consumption in current networks, which has been indicated in the previous section 4. As far as known to the authors, currently there do not exist any estimations of power consumption in the networks besides Bitcoin and Ethereum.

6 CONCLUSION

In this paper, we presented a set of alternatives to the energy inefficient Proof of Work consensus algorithm as used by Bitcoin. The consensus algorithms reviewed in this paper offer solutions that consume significantly less energy than Proof of Work-based consensus algorithms, as they do not rely on computationally expensive calculations to select a block creator. To make a decision on what consensus algorithm is best, more factors besides energy efficiency as discussed in this paper should be taken into account. Furthermore, we proposed other potential research topics to improve upon the work done in this paper, such as the question of whether people using Bitcoin are generally aware of the power consumption of the Bitcoin network. Finally, a discussion about permissioned and permissionless blockchains has been introduced, which deserves attention in future work.

REFERENCES

[1] Market Research Future, *Blockchain Technology Market Research Report - Global Forecast to 2023*, 2018. [Online]. Available: https://www.marketresearchfuture.com/reports/blockchain-technology-market-1708 (visited on 03/31/2019).

[2] S. Haber and W. S. Stornetta, "How to time-stamp a digital document," in *Conference on the Theory and Application of Cryptography*, Berlin, Heidelberg: Springer, 1990, pp. 437–455.

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

[4] CoinMarketCap, *All Cryptocurrencies*, 2019. [Online]. Available: https://coinmarketcap.com/coins/ (visited on 03/31/2019).

[5] A. de Vries, "Bitcoin's growing energy problem," *Joule*, vol. 2, no. 5, pp. 801–805, 2018.

[6] A. de Vries, "Renewable energy will not solve bitcoin's sustainability problem," *Joule*, 2019, ISSN: 2542-4351. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S254243511930087X.

[7] Digiconomist, *Bitcoin Energy Consumption Index*, 2019. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption (visited on 03/31/2019).

[8] U. W. Chohan, *The double spending problem and cryptocurrencies*, 2017. [Online]. Available: https://ssrn.com/abstract=3090174.

[9] J. Mattila, "The blockchain phenomenon," *Berkeley Roundtable on the International Economy (BRIE)*, pp. 2016–1, 2016. [Online]. Available: https://brie.berkeley.edu/sites/default/files/juri-mattila-.pdf.

[10] G. Wood, *Ethereum: A secure decentralised generalised transaction ledger (eip-150 revision)*, 2016. [Online]. Available: http://gavwood.com/paper.pdf.

[11] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *CoRR*, vol. abs/1710.09437, 2017. [Online]. Available: http://arxiv.org/abs/1710.09437.

[12] D. L. K. Chuen and L. Low, *Inclusive FinTech: Blockchain, Cryptocurrency and ICO*. World Scientific, 2018.

[13] block.one, *EOS.IO Technical White Paper v2*, 2018. [Online]. Available: https://github.com/EOSIO/Documentation/blob/a95c3236b8fd94c8546954ce63367df29de33a02/TechnicalWhitePaper.md (visited on 03/31/2019).

[14] D. Mazieres, *The Stellar consensus protocol: A federated model for internet-level consensus*, 2015. [Online]. Available: https://www.stellar.org/papers/stellar-consensus-protocol.pdf.

[15] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*, Springer, 2017, pp. 357–388.

[16] TRON Foundation, *Advanced Decentralized Blockchain Platform*, 2018. [Online]. Available: https://tron.network/static/doc/white_paper_v_2_0.pdf (visited on 03/31/2019).

[17] S. Popov, *The tangle*, 2018. [Online]. Available: http://www.descryptions.com/Iota.pdf.

[18] L. Goodman, *Tezos - A self-amending crypto-ledger White paper*, 2014. [Online]. Available: https://tezos.com/static/white_paper-2dc8c02267a8fb86bd67a108199441bf.pdf (visited on 03/31/2019).

[19] The Neo Project, *NEO White Paper*, 2018. [Online]. Available: https://github.com/neo-project/docs/blob/3c5530197768d98a1abf7eeed0119a8f4e99e7cc/en-us/whitepaper.md (visited on 03/31/2019).

[20] The Ontology Team, *Ontology Launches VBFT, a Next-Generation Consensus Mechanism, Becoming one of the First VRF-Based Public Chains*, 2018. [Online]. Available: https://medium.com/ontologynetwork/ontology-launches-vbft-a-next-generation-consensus-mechanism-becoming-one-of-the-first-vrf-based-91f782308db4 (visited on 03/31/2019).

[21] VeChain Foundation, *Defining the VeChainThor Blockchain Consensus — Proof of Authority*, 2018. [Online]. Available: https://medium.com/vechainofficial/defining-the-vechainthor-blockchain-consensus-proof-of-authority-8cf3f51a5fa0 (visited on 03/31/2019).

[22] Anonymous, *Proof of stake instead of proof of work*, 2011. [Online]. Available: https://bitcointalk.org/index.php?topic=27787 (visited on 03/24/2019).

[23] S. King and S. Nadal, *Ppcoin: Peer-to-peer crypto-currency with proof-of-stake*, Aug. 2012. [Online]. Available: https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf.

[24] Nxt Community, *Whitepaper:nxt*, 2014. [Online]. Available: https://nxtwiki.org/wiki/Whitepaper:Nxt (visited on 03/10/2019).

[25] Waves Platform, *What is Waves Platform*, 2019. [Online]. Available: https://docs.wavesplatform.com/en/ (visited on 03/22/2019).

[26] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies Without Proof of Work," in *Financial Cryptography and Data Security*, Berlin, Heidelberg: Springer, 2016, pp. 142–157, ISBN: 978-3-662-53357-4.

[27] NEM Foundation, *NEM Technical Reference*, 2018. [Online]. Available: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf (visited on 04/01/2019).

[28] POA Network, *Proof of Authority: consensus model with Identity at Stake*, 2017. [Online]. Available: https://medium.com/poa-network/proof-of-authority-consensus-model-with-identity-at-stake-d5bd15463256 (visited on 04/01/2019).

[29] Intel Corperation, *PoET 1.0 Specification*, 2016. [Online]. Available: https://sawtooth.hyperledger.org/docs/core/releases/1.0/architecture/poet.html (visited on 04/01/2019).

[30] R. S. Seán Gauld Franz von Ancoina, *The burst dymaxion*, 2017. [Online]. Available: https://www.burst-coin.org/wp-content/uploads/2017/07/The-Burst-Dymaxion-1.00.pdf.

[31] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[32] B. Chase and E. MacBrough, "Analysis of the XRP Ledger Consensus Protocol," *CoRR*, vol. abs/1802.07242, 2018. arXiv: 1802.07242. [Online]. Available: http://arxiv.org/abs/1802.07242.

[33] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proceedings of the 3rd Symposium on Operating System Design and Implementation*, 1999, pp. 173–186.

[34] D. Schwartz, N. Youngs, and A. Britto, *The Ripple Protocol Consensus Algorithm*, 2014. [Online]. Available: https://ripple.com/files/ripple_consensus_whitepaper.pdf.

[35] E. Zhang, *A Byzantine Fault Tolerance Algorithm for Blockchain.* [Online]. Available: https://docs.neo.org/en-us/basic/consensus/whitepaper.html (visited on 04/08/2019).

[36] *Bitcoin Hashrate Distribution*, 2019. [Online]. Available: https://www.blockchain.com/pools (visited on 04/01/2019).

[37] S. Zheng, *Crypto simplified: Explaining permissioned blockchains*, 2018. [Online]. Available: https://www.theblockcrypto.com/2018/12/10/crypto-simplified-explaining-permissioned-blockchains/ (visited on 04/08/2019).

[38] F. Glaser, K. Zimmermann, M. Haferkorn, M. C. Weber, and M. Siering, "Bitcoin-asset or currency? revealing users' hidden intentions," *ECIS 2014 (Tel Aviv)*, 2014. [Online]. Available: https://ssrn.com/abstract=2425247.