# Energy Efficiency of Distributed Consensus Algorithms

Hanyu Wei
*College of Natural Sciences*
*University of Texas at Austin*
Austin, USA
hywei737@utexas.edu

***Abstract*—Cryptocurrency and blockchain technologies are rapidly growing in energy consumption. The US Energy Information Administration found that in 2024 electricity consumption from cryptocurrency mining in the United States "represents from 0.6% to 2.3% of U.S. electricity consumption" [5] . This paper seeks to compare different commonly used distributed consensus algorithms used in cryptocurrencies, such as Proof of Work, Proof of Stake, Proof of Reserve, and Proof by Agreement based on their energy efficiency, as well as some proposed improvements such as Green-Proof of Work. We analyze their effectiveness at reducing the energy cost of cryptocurrencies while maintaining the security necessary to ensure trust in these cryptocurrencies. We identify that these algorithms have energy costs both in creating a security layer so that only trusted machines vote on their We found that Proof of Work, while the most common consensus algorithm, is also by far the least energy efficient. We also found some novel proposals such as Green Proof of Work, which may reduce energy consumption on paper, but in reality, may increase energy consumption. We recommend that cryptocurrencies turn to alternative algorithms such as Proof of Stake and Proof of Agreement to reduce their energy consumption.**

***Index Terms*—Energy Efficiency, Blockchain, Distributed Consensus Algorithms, Byzantine Generals Problem**

## I. Defining The Problem

During the Great Recession, multiple large banks failed such as Lehman Brothers, which plunged the global economy into a recession. The resulting financial turmoil reduced public trust in traditional banks. Cryptocurrencies were created to provide a novel currency that has decentralized authority, preventing any central authority from debasing the currency. Cryptocurrency has become heavily integrated into the global economy, with Forbes estimating that as of November 11, 2024, the total cryptocurrency market cap is over $3 Trillion USD [15]. Unfortunately, this expansion has come at a massive energy cost, with Schmidt estimating global cryptocurrency energy consumption is around 127 TWh, which "exceeds the entire annual electricity consumption of Norway" [15]. Instead of having a centralized authority that can decide on the veracity of a transaction, traditional distributed consensus algorithms use energy both to send $O(n^2)$ messages to the various distributed verifiers, as well as have the verifiers solve complex mathematical problems to verify the transactions. Newer algorithms promise to reduce energy consumption, but most focus on reducing the energy cost of verification while ignoring the networking cost associated with decentralizing authentication. Section II will describe the necessity of Distributed Consensus Algorithms for Cryptocurrencies. Section III will provide our proposed framework for analyzing the energy efficiency of various Distributed Consensus Algorithms. Sections IV-VI will discuss various distributed consensus algorithms, their implementation history using example cryptocurrencies, and analyze their energy efficiency using our framework. Section VII will compare our Distributed Consensus Algorithms based on their energy efficiency. Section IX will conclude the paper.

## II. Maintaining Trust in Cryptocurrencies Using Distributed Consensus Algorithms

For financial transactions, it is crucial that transactions are verified to prevent fraud such as through double-spending, where a fraudulent actor spends a coin in two separate transactions. In traditional centralized financial systems, there is a central trusted authority such as a Bank that acts as the ultimate validator of transactions. This ensures that there is one record of transactions, preventing double-spending. In cryptocurrencies, transactions are verified through a pool of independent users verifying each other's transactions. Transaction histories are stored on distributed ledgers, where each user stores replicates, and shares transactions across the network. However, this can lead to fraudulent transactions such as double-spending being undetected as different users have a different version of the distributed ledger due to network connection differences. Therefore, we need a way to maintain consensus on which transactions are valid and which transactions are not. Failure to reach a consensus can be disastrous for cryptocurrencies as this can lead to double spending. This can happen either accidentally such as when Stellar forked itself in 2014 due to two sections of the network being unable to reach consensus [24], or intentionally when hackers attacked Bitcoin Gold in 2019 with a 51 percent attack, allowing them to double-spend, creating over $18 million USD in false transactions [26]. The Byzantine Generals Problem provides a model of the problem of maintaining distributed consensus.

### A. Byzantine Generals Problem

Introduced by Leslie Lamport in 1984, the Byzantine Generals Problem identifies how to maintain consensus with malicious actors. In the Byzantine Generals Problem statement,

there are n Byzantine generals trying to attack a fort. To avoid disastrous losses, the generals must either all attack at once or retreat at once, as any attack with only some of the generals would lead to a disastrous defeat. A simple solution is to have the generals vote on the course of action, which would ensure all the generals either attack or retreat together. However, if some of the generals are traitors, we cannot guarantee the accuracy of this vote. If at least 51% of the generals are loyal, then we can ensure that the generals vote for the correct action. A solution the minority of traitors can do to gain 51% of the votes, however, is to create fake generals to vote in the election [1]. This is the Sybil attack, where an attack creates many fake identities that are secretly controlled by one attacker [2]. In terms of cryptocurrency, this is equivalent to users voting on the accuracy of transactions, so a malicious actor spins up a botnet to mass vote for fraudulent transactions. Thus, the goal of the distributed consensus algorithms for cryptocurrencies is to make it extremely difficult for any attacker to gain the majority of verification power, however, this usually comes at the cost of extreme energy consumption to secure the cryptocurrency.

## III. The Energy Cost of Distributed Consensus Algorithms

Centralized financial systems are very energy efficient as transactions can be verified through only checking with a central authority. For example, a Visa transaction requires only 0.00092 kWh [15] of energy to be verified. Cryptocurrencies promised to democratize the financial system by removing this central authority, and instead verifying transactions using distributed consensus algorithms. Unfortunately, distributed consensus is inherently far more energy-inefficient than centralized consensus. We propose a framework for analyzing the energy efficiency of Distributed Consensus Algorithms using the energy they use to maintain security, the energy they use networking to ensure consensus (which is commonly overlooked), and how the energy consumption scales as the cryptocurrency becomes more popular.

### A. The Energy Cost of Security

The most well-known energy cost of distributed consensus algorithms is the security cost. Traditionally, this has involved verifiers solving NP-complete math problems, which is quite obviously energy inefficient. The most energy-inefficient methods require up to o($2^n$) to compute, so the security cost is the most important factor in the energy efficiency of a distributed consensus algorithm. We will describe algorithms that control the security cost as being security efficient and algorithms that do not control the security cost as being security inefficient. Many of the newer algorithms try to be security efficient, however, this energy cost often overshadows the send energy cost of Distributed Consensus Algorithms.

### B. The Energy Cost of Networking

A lesser-understood cost of distributed consensus algorithms is the networking cost. A centralized consensus algorithm only requires O(n) messages to be sent to verify transactions, as only the sender, recipient, and verifier need to know about the transaction. However, since in distributed consensus algorithms verification is distributed amongst all the network participants, the amount of messages we need to send becomes O($n^2$). This may seem like a small change at first but scales quadratically the more network participants there are, which causes popular cryptocurrencies to be energy inefficient. For example, the most energy-efficient algorithm we will look at, Stellar Consensus Protocol, estimates that over 94 percent of their energy cost is from networking.

### C. The Energy Cost of Scalability

The final energy cost of distributed consensus algorithms is the energy cost of scalability. Some distributed consensus algorithms can grow massively without majorly affecting the energy cost, whereas others exponentially increase in energy cost as network participants rise. Since the largest cryptocurrencies are the ones that have disproportionate energy usage, there is no point in a distributed consensus algorithm that is energy efficient at a small scale but energy inefficient when deployed at a global scale.

### D. Our Ranking Criteria

We will describe algorithms that control the networking cost as being network efficient, algorithms that do not control the networking cost as being network inefficient, and algorithms that scale efficiently as scalability efficient. When ranking the algorithms, we take both security cost, networking cost, and scalability cost into consideration to determine the most energy-efficient distributed consensus algorithm.

## IV. Proof of Work

Popularized by Bitcoin, Proof of Work was the first successful distributed consensus algorithm used for cryptocurrencies. It is one of the most important distributed consensus algorithms for cryptocurrencies, with half of the top ten cryptocurrencies using Proof of Work as the consensus algorithm, as shown in Figure 1 [1]. Proof of Work is arguably the most secure consensus algorithm, but it is by far the most energy-intensive consensus algorithm, being security inefficient, network inefficient, and scalability inefficient.

### A. Bitcoin

On 31 October 2008, an anonymous developer commonly known as Satoshi Nakamoto wrote a white paper describing Bitcoin, the first decentralized cryptocurrency and the most widely used cryptocurrency in the world, with a market cap of $157.3 Billion USD. Inspired by Adam Back's Hashcash, bitcoin was the first major implementation of the distributed consensus algorithm proof of work [3]. Bitcoin verifies the entire transaction history of a coin using proof of work. A transaction is conducted by the previous owner digitally signing the timestamp of a transaction and adding the public key of the next owner to the block of the previous transaction. This allows the recipient to verify the transaction by computing the hash using the previous record and their public key.

| Name | Consensus Algorithm | Market Cap | Rank |
|---|---|---|---|
| Bitcoin | Proof of Work | $1,948 B | 1 |
| Ethereum | Proof of Stake | $402 B | 2 |
| Tether | Proof of Reserves | $132 B | 3 |
| Solana | Proof of History | $121 B | 4 |
| BNB | Proof of Staked Authority | $93 B | 5 |
| Ripple | Ripple Consensus Protocol | $91 B | 6 |
| Dogecoin | Proof of Work | $59 B | 7 |
| Cardano | Proof of Stake | $38 B | 8 |
| USDC | Proof of Reserves | $18 B | 9 |
| TRON | Delegated Proof of Stake | $18 B | 10 |
| Stellar | Stellar Consensus Protocol | $13 B | 14 |
| XEM | Proof of Importance | $0.21 B | 278 |

Figure 1. Cryptocurrencies by Market Cap on November 22, 2024 [23]

However, this could allow the sender to "double spend" by signing multiple transactions using the same coin. Prior distributed consensus algorithms failed to combat this attack. Bitcoin is resistant to double spending, by grouping new transactions onto a block in a Merkle Tree. This new transaction is announced to all the verifiers (called miners) and asked to find a nonce value which when added to the block produces a hash with a certain number of leading zero bits. Once a miner has found a nonce, they broadcast it to everyone else, which verifies that the transaction is valid and adds the block to the distributed ledger. This is an extremely energy-intensive process as finding the nonce value is an NP-complete problem [3].

This is how Proof of Work tries to solve the 51 percent attack, by making acquiring voting rights extremely expensive in energy cost, so one miner gaining over 50 percent of computing power would be prohibitively expensive. The value of the Nonce is dynamically adjusted based on the total computing power of the miners to ensure that a limited number of bitcoins are minted. Bitcoin tries to save some hash computations by storing transactions in a Merkle tree [3]. This means that appending a new transaction to the chain only requires the new node and corresponding interior node's hashes to be updated rather than the hash of the entire block, which saves some energy. Merkle trees also allow old blocks to be compacted by pruning old branches that are not needed to compute the hash, saving disk space.

Because Blockchain uses distributed ledgers, this means that each network participant will have a slightly different version of the ledger due to network latency. In addition, it is possible that multiple parts of the network are disconnected, leading to multiple blockchains being formed. If that is the case, the longest chain is assumed to be the correct chain. Transactions on the wrong chain need to be added to the new chain by recomputing nonce values or else their transaction history will disappear from the distributed ledger [3].

### *B. Energy Efficiency Analysis*

Proof of Work remains an inherently energy-inefficient consensus algorithm, as it relies on the expensive energy cost of mining to secure transactions. Bitcoin's white paper specifically calls the usage of energy to mine Bitcoin "analogous to gold miners expending resources to add gold to circulation." Essentially, energy is the cost needed to verify a user as legitimate by making the energy costs for an attacker to be overwhelmingly expensive. In addition, the energy cost of miners who did not find the nonce value first is wasted, but every miner contributes to the difficulty of calculating the nonce, as the nonce difficulty increases based on the total computational power of the whole network. The fork choice rule also harms the energy efficiency of Proof of Work as not only is any mining work on shorter chains wasted, but any transactions on the shorter chains that want to be added to the correct chain need to be recomputed with the correct nonce. Therefore, Proof of Work is clearly security inefficient. Bitcoin is highly network inefficient because o($n^2$) messages have to be sent as any transaction needs to be verified by various distributed miners, and then propagated across the network to various distributed ledgers. Unfortunately, the energy cost of Proof of Work is massive, as Wanecek finds that Bitcoin uses 1,575.93 kWh per transaction, while a Visa transaction uses 0.00092 kWh [15].

Uniquely, Proof of Work demonstrates extreme scalability inefficiency. As a Proof of Work secured cryptocurrency becomes more popular, the network cost and security cost rise exponentially. The network cost would increase due to the number of nodes on the network that need to reach consensus increasing, resulting in more messages being sent. The security cost would increase as more transactions would increase, increasing the transaction fee for a block. This would increase the incentive for miners to mine, increasing the hash rate of the network, which would increase the difficulty of cracking the hash to keep block generation at a constant rate, therefore increasing the energy needed to verify a transaction. This has been shown as Bitcoin's electricity consumption has exponentially increased as it has gotten more popular according to Figure 2. Unfortunately, this energy cost is necessary for the security of the cryptocurrency, as Bitcoin Gold was hit by a 51 percent attack soon after it was forked from Bitcoin due to the energy cost of transactions being very low, causing almost $18 Million in losses [26].

### *C. Green PoW – Is It An Improvement?*

Given the energy wastefulness of Proof of Work, there have been many proposed changes to Proof of Work to reduce energy consumption. One example of this is Green-PoW proposed by Lasla et al [2]. Green-PoW proposes the mining process is split up into two epochs. In the first round, all miners are allowed to compete and mine a block like in normal Proof of Work. In addition, a small subset of miners who did not find the nonce first are elected to mine the next

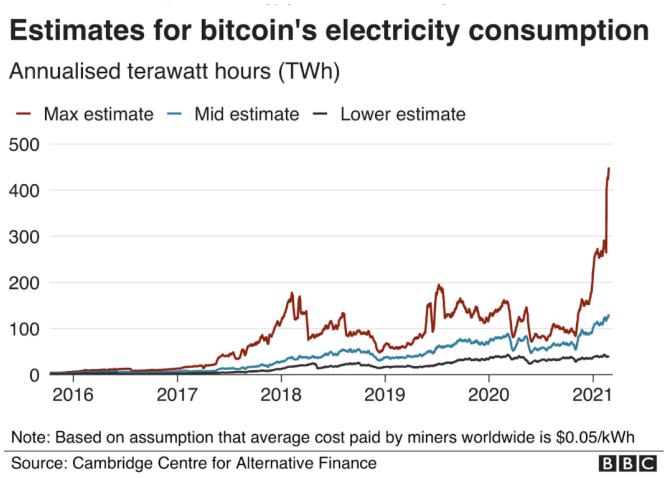


Figure 2. Bitcoin energy usage

block in the next epoch. This means that the losers do not "waste" energy, as mining in the first round ensures they can be selected to mine in the power-save round. In the power-save round, since the hash rate of the network is lower, the hash difficulty decreases to ensure a consistent block generation rate. This power-save round also decreases fork occurrence during the power-save round because there are fewer miners who could be unaware of a block generation. However, it could be possible no block is generated in the power-save round if the power-save miners get disconnected from the network or their hash rate is not high enough to find the correct nonce in the block generation time. In that case, the power save round would time out, and the blocks would go to the next full power round to ensure the blocks get added to the chain. Green-PoW is still highly resistant to Sybil attacks as any attacker that wants to compete in both rounds under multiple fake identities must have enough computing power to win the mining in the first round to identify its fake accounts as miners in the power-save round [2].

### D. Energy Efficiency Analysis

On the surface, Green-PoW seems like a very promising energy-efficiency improvement over Proof of Work. Lasla et al. claim that Green-PoW can reduce energy consumption by nearly 50 percent compared to traditional Proof of Work without degrading the security of Proof of Work, by allowing the fastest miners to turn off their machines in the power-save round [2]. On the surface, this makes sense, as the fastest machines likely consume the most power. This also means that the loser's energy cost is not wasted as much as they can still be chosen for the power-save round, thus giving Green-PoW security efficiency. Because there are fewer miners in the power-save round, there will be fewer fork occurrences as fewer of the power-save miners will be unaware of a block generation, which is good for energy efficiency as that ensures less mining work is done on incorrect chains. An unintentional consequence is that we will reduce the number of messages sent in the power-save round as only a subset of the machines need to know about the block to validate it, thus providing network efficiency. This also provides scalability efficiency as the power-save round reduces the hash difficulty to account for the lower hash rate and there are fewer miners competing in the power-save round regardless of the size of the network. Therefore, it may seem like Green-PoW is a good solution to our problems. However, this assumption is naive thanks to Jevon's paradox. Since in Green-PoW miners only make money by actively mining blocks, a miner not selected for the power-save round is incentivized to switch to mining another cryptocurrency because otherwise, they would be losing the opportunity to earn money, thus leading to more energy consumption overall as they can mine multiple blockchains at once. Therefore, we do not recommend Green-PoW as an energy-saving alternative to regular Proof of Work.

## V. Proof of Stake

To try to combat the energy cost of Proof of Work, Proof of Stake was introduced as an alternative distributed consensus algorithm. Implemented most notably by Ethereum, the second most popular cryptocurrency by market cap, and Cardano, the eighth most popular cryptocurrency by market cap, Proof of Stake tries to reduce energy consumption by validating transactions through coin ownership rather than computational power, which makes it security efficient. However, Proof of Stake does not improve the network cost, because a majority of validators still need to know about the transaction and vote on the veracity of the transaction.

### A. Ethereum

Ethereum is the second-largest cryptocurrency by market capitalization and the largest cryptocurrency using Proof of Stake. When Ethereum was first introduced in 2013 by Vitalik Buterin, it utilized Proof of Work, however, Ethereum was upgraded to Proof of Stake on September 15, 2022, in an update called "the merge" to improve its energy efficiency [9]. Proof of Stake aimed to secure the network and validate transactions through coin ownership rather than computational power. In Proof of Stake, validators participate by placing a "stake" of coins. As a block is announced to be verified, a validator is chosen pseudo-randomly based on the size of their stake and the age of their stake. Validators more invested (e.g. that have more coins staked for a longer period of time) are more likely to be chosen as they are assumed to be more invested in the coin and have an incentive to validate honestly. This validator then decides on the validity of a block. Other validators attest to the accuracy of a block, and once a majority agree that the block is valid it is added to the chain, and the validator earns a transaction fee and some minted coins.

Proof of Stake protects against a 51 percent attack as an attacker would need to own more than 51 percent of the total coins staked to mount a 51 percent attack, however, this would be prohibitively expensive for an attacker to obtain. Ethereum's implementation of Proof of Stake refines the process by using a fork-choice rule that prioritizes blocks

with the highest weight of "attestations," or validator votes, which increases resilience against attacks [10]. Proof of Stake may also suffer from the nothing-at-stake problem, where bad actors can validate multiple forks at once to increase their chance of earning the reward, but this can fragment the distributed ledger. To combat nothing-at-stake, Ethereum punishes validators that publish multiple contradictory messages using slashing, which takes away some of the stake of validators that have been validating incorrect transactions [10].

### B. Energy Consumption Analysis

Unlike Proof of Work, where miners compete to solve cryptographic puzzles and expend significant energy to generate each block, Proof of Stake's validator selection relies on stakes instead of competitive hashing, making block generation and consensus less energy-intensive [8]. This means that validators do not need insanely powerful and power-hungry data centers to validate transactions, so Proof of Stake is a security-efficient distributed consensus algorithm. However, Proof of Stake does not target the networking cost of distributed consensus algorithms. Every transaction requires all the validators to know about and attest to the validity of the transaction, thus Proof of Stake still requires o($n^2$) messages to be sent, so it is network inefficient. Proof of Stake has a medium stability cost, as the energy cost of verifying transactions rises as the network becomes more popular, but is a polynomial increase rather than the exponential increase as in Proof of Work. Proof of Stake has demonstrated itself as a viable alternative to proof of work that reduces energy costs without harming security. Empirically, Ethereum reduced its energy consumption by 99.95% by switching to proof of stake from proof of work and has not suffered any major security incidents [9]. A transaction on Ethereum takes only 0.03 kWh [14], which is significantly improved from 1,575.93 kWh per transaction needed for Bitcoin [15].

## VI. Proof of Authority

Proof of Authority is a distributed Consensus Algorithm used notably by VeChain [25]. Proof of Authority tries to tackle the scalability cost of earlier distributed consensus algorithms by relying on a limited number of trusted validators to validate transactions. However, this means Proof of Authority is highly centralized, which is antithetical to the original goal of cryptocurrencies.

### A. Vechain

In 2017, Gavin Wood introduced the Distributed Consensus Algorithm Proof of Authority, which has become used by Vechain. Proof of Authority is a consensus model that relies on validators' reputations rather than stake ownership or computational power. Proof of Authority assigns the role of validators to a small, pre-selected group of authorized signers, which is set in the blockchain's genesis block. These signers, often verified through Know Your Customer (KYC) processes, are well-known validators chosen by the community [11]. Rather than staking their coins like in Proof of Stake, validators' public reputations are effectively "staked," meaning that any misconduct or failure is directly attributable to them, which incentivizes reliability and honesty. However, this model sacrifices decentralization: with only a few known validators, Proof of Authority systems are vulnerable to censorship and may lack the resilience and openness valued in traditional blockchain systems. When a block is proposed, a validator is chosen in a round-robin order to verify the transaction. The result of the validators approve or deny the validation, and if a majority agrees then the transaction is added to the blockchain. Proof of Authority avoids the 51 percent attack as validators are well-known, so any dishonest validator would be publically known and could be easily purged from trusted validators.

### B. Energy Consumption Analysis

From an energy perspective, Proof of Authority is highly efficient. Since all a validator has to do is vote on whether a transaction is valid or invalid, security costs for Proof of Authority are extremely low. In addition, the network costs for Proof of Authority are extremely low as there are a fixed number of validators, so the number of messages needed to be sent is constant. Therefore, Proof of Authority is easily scaleable without massive energy costs. Since the number of validators is fixed, any transaction would only need the same number of votes on the accuracy regardless of the size of the network, so it is scalability efficient.

## VII. Proof of Importance

Proof of Importance is a distributed consensus algorithm introduced by the NEM blockchain for its native cryptocurrency, XEM. Proof of Importance was designed to remove the incentive to hoard coins which is present in Proof of Stake.

### A. NEM

One issue that Proof of Stake faces is the rich-get-richer problem. In a pure Proof of Stake network, "whales" of the network might be encouraged to simply hoard their pool of currency, as their voting power and resulting rewards simply increase over time. Designed to address these centralization issues seen in Proof of Stake, Proof of Importance assigns each account an "importance" score that reflects both their coin holdings and their activity within the network. Accounts with higher importance scores are more likely to be selected to "harvest" or validate a block, encouraging users to participate in transactions rather than simply hoard currency.

In Proof of Importance, importance is calculated using vested XEM (the portion of XEM held long enough to be "vested") and the position/relevance of an account within the network topology. As a prerequisite, an account must have 10,000 XEM to partake in the Proof of Importance importance calculations. As outlined in the NEM whitepaper [12], the first step in the Proof of Importance calculation is the :outlink matrix," which is a measure of the net flow of XEM to/from an account. All the outgoing transactions of at least 1,000 XEM without 30 days are collected, weighted greater the more the more recent it was. A similar value is calculated

for the incoming transactions. The difference of these values are normalized and then placed into a matrix. Next, similar to how search engines rank websites, the NCDawareRank algorithm (similar to PageRank) ranks the nodes importance in the network, taking into account the outlink matrix. Finally, the NCDawareRank score is combined with the amount of vested XEM to find the importance score, where it is used to determine the likelihood of harvesting a block.

The usage of the NCDawareRank algorithm and using net flow as a metric for importance helps to combat Sybil attacks. For instance, in an attack similar to link spamming, an account sends XEM to other accounts and has the other accounts send funds back, feigning importance in the network. However, as compared to PageRank, NCDawareRank is more resistant to such link-spamming attacks [19]. In addition, using net flow as a metric helps take into account the behavior of others. For instance, an attacker cannot just send XEM to many accounts to boost importance, as the incoming XEM sent to the other accounts will cancel out the outgoing XEM sent by the attacker [12]. Lastly, establishing a minimum balance of 10,000 vested XEM for harvesting and coin age also aims to increase the unprofitability of malicious attacks in a similar manner as Proof of Stake.

### B. Energy Consumption Analysis

Proof of Importance is more energy-efficient than Proof of Work, as it avoids the energy-intensive computations that Proof of Work requires. However, Proof of Importance suffers a worse performance compared to Proof of Stake because, in addition to staking, Proof of Importance requires additional network analysis algorithms to verify the importance of a node. Therefore, Proof of Importance has a moderate security efficiency. Like Proof of Stake, Proof of Importance does not deal with the networking cost. Since like in Proof of Stake, transactions are verified across the network we need O($N^2$) messages to be sent, so it is network inefficient. As the network size grows, or network activity increases, the computational costs for these network analysis algorithms grow quickly, causing an increase in energy costs. Therefore, Proof of Importance is scalability inefficient.

## VIII. Proof of Agreement

Proof of Agreement is the final distributed consensus algorithm we will discuss. It is implemented most notably by the Stellar Consensus Protocol in the Stellar cryptocurrency.

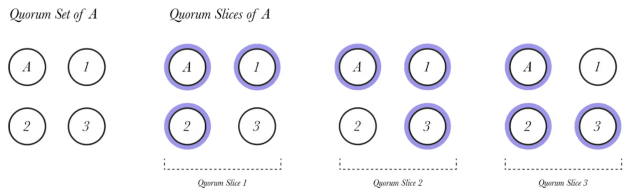


Figure 3. Example of quorum and quorum slices [13]

### A. Stellar

Stellar was originally created by Jed McCalebe and Joyce Kim in 2014 on the Ripple Consensus Protocol, but unintentionally forked itself in December 2014 due to a failure to maintain consensus [24]. The Stellar Consensus Protocol was designed by David Mazières in 2015, as a replacement consensus algorithm with decentralized control, good performance, and low financial/energy requirements in mind [22]. To achieve this, nodes in the Stellar network are organized into quorums and quorum slices. A quorum is a set of trusted nodes, and a quorum slice is a subset of the quorum that is needed to convince a node of a particular state/transaction (pictured in Fig. 3). At a high level, processing transactions is divided into two sections, the nomination protocol and the ballot protocol.

In the nomination protocol, candidate transaction sets are proposed to be included in the ledger. After a node confirms its first candidate, it stops voting to nominate any new transaction sets [13]. However, it can still accept/confirm previously selected transaction sets (ensuring that the nodes eventually converge on a candidate set) [13]. Once the nodes believe that the nomination protocol has converged on a transaction set, the ballot protocol begins, where they vote to commit or abort the set. This stage makes use of federated voting, in which agreement on a certain statement is broken into two steps. First, the nodes vote to accept a statement. To accept a statement, the node must have never accepted a conflicting statement (as a rational node would not change its stance on accepting a statement) and a large enough portion of the node's quorum slice must vote to accept transaction [21]. Then, the nodes vote to confirm the statement, essentially voting on the fact that the first vote succeeded. For a given node, this means it views a statement a as unknown, accepted, or confirmed. The two stages of voting are needed because other functioning nodes may be unable to assess the current statement due to missing information/unresolved dependencies, or that the current node is a bad actor and accepting contradictory statements [22]. Should the nodes vote to commit the composite value, it is "externalized" for the slot.

| System | Electricity consumption per transaction | |
|---|---|---|
| Bitcoin [57] | 1 575.93 | kWh |
| Ethereum [58] | 107.75 | kWh |
| VISA [59] | 0.00092 | kWh |
| Stellar (this study) | 0.00022 | kWh |

Figure 4. Comparison of Stellar Consensus Protocol per transaction cost with other systems [20]. (Note this is pre-merge Ethereum)

### B. Energy Consumption Analysis

Similar to Proof of Stake, the energy efficiency of the Stellar Consensus Protocol is far better than the most commonly used Proof-of-Work, as consensus is not reached by racing to finish NP-complete operations.

The Stellar Consensus Protocol relies on networking to nominate, accept, and confirm transaction sets. In fact, networking contributes about 94% of the total energy cost [20]. However, since only the quorum slice is needed to verify a transaction rather than the whole network, the constant factor in the $O(N^2)$ messages for consensus is reduced, producing some impressive energy efficiency improvements.

According to experiments by Wanecek, the per transaction cost for the Stellar Consensus Protocol has been found to be around 0.222 Wh, or 0.000222 kWh [20] (pictured in Fig. 4), which is much lower than that of Ethereum at around 0.03 kWh [14]. In addition, according to Cole et al., increasing the size of quorums did not cause a significant increase in the number of network messages sent [21], which could mean that Stellar Consensus Protocol is scalability efficient.

## IX. Energy Analysis

We summarize the energy efficiency of the various distributed consensus algorithms based on Figures 5 and 6. We can see that security cost has the largest impact on the energy efficiency of a distributed consensus algorithm. Networking efficiency has a comparatively lower impact on the energy efficiency of distributed consensus algorithms. Scalability efficiency is an important caveat to mention, although it is very difficult to compare as the example cryptocurrencies are wildly different in popularity. By using our energy efficiency framework, we rank the distributed consensus algorithms based on energy efficiency as Proof of Work, Green-PoW, Proof of Importance, Proof of Stake, Proof of Agreement, and then finally Proof of Authority.

## X. Conclusion

While highly secure and the backbone of the largest cryptocurrencies such as Bitcoin, Proof of Work is the worst distributed consensus algorithm in terms of energy consumption. Green-PoW may seem like an improvement at first, but due to Jevon's Paradox may increase the total energy usage. Proof of Importance is less energy expensive. It derives the importance of a node based on its network activity/topology and vested coins, which translates to voting weight. While this provides an incentive not to hoard coins, the extra overhead of network analysis, especially once the network has grown large, causes its energy efficiency to suffer. After that, Proof of Stake is the next least energy-expensive algorithm, being popularized by Ethereum. Proof of Stake effectively reduces the energy cost of transactions by using staking for consensus, but it does not touch the networking cost. Proof of Agreement is the second to least energy-expensive, being the first algorithm to effectively deal with both the networking cost of sending transaction messages and the computing cost of verifying transactions through quorum slices. Proof of Authority is the least energy-intensive algorithm, although by delegating trusted nodes it essentially recreates a centralized system, breaking the purpose of a decentralized system.

We believe that Proof of Agreement is the best distributed consensus algorithm for energy-efficiency as it significantly decreases the energy cost of transactions while maintaining the benefits of being a distributed algorithm. It does so by attacking both sides of the energy consumption problem-reducing the messages we need to send and reducing the computation cost of verifying transactions. Proof of Stake is the next best alternative as it effectively reduces the computation cost of verifying transactions, but it does not deal with the networking costs needed to maintain consensus.

| Consensus Algorithm | Security | Networking |
|---|---|---|
| Proof of Work | High | High |
| Green-PoW | High | High |
| Proof of Stake | Low | High |
| Proof of Authority | Low | Low |
| Proof of Importance | Medium | High |
| Proof of Agreement | Low | Medium |
| Visa Transaction | Low | Low |

Figure 5. Analysis of Consensus Algorithm in Security Costs and Networking Costs

| Consensus Algorithm | Scalability | Energy Cost |
|---|---|---|
| Proof of Work | Low | 1,575.93 kWh |
| Green-PoW | High | unknown |
| Proof of Stake | Low | 0.03 kWh |
| Proof of Authority | High | unknown |
| Proof of Importance | Low | unknown |
| Proof of Agreement | High | 0.000222 kWh |
| Visa Transaction | High | 0.00092 kWh |

Figure 6. Analysis of Consensus Algorithms in Scalability and Energy Cost per Transaction

## References


[1] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," IEEE Xplore, May 01, 2018. https://ieeexplore.ieee.org/abstract/document/8400278

[2] N. Lasla, L. Al-Sahan, M. Abdallah, and M. Younis, "Green-PoW: An energy-efficient blockchain Proof-of-Work consensus algorithm," Computer Networks, vol. 214, p. 109118, Sep. 2022, doi:https://doi.org/10.1016/j.comnet.2022.109118.

[3] S. Nakamoto, "Bitcoin: a Peer-to-Peer Electronic Cash System," bitcoin.org, Oct. 2008. Available: https://bitcoin.org/bitcoin.pdf

[4] "Blockchain's Energy Crisis — SAP Insights," SAP. https://www.sap.com/sea/insights/viewpoints/blockchains-energy-crisis.html

[5] M. Morey, G. McGrath, and H. Minato, "Tracking electricity consumption from U.S. cryptocurrency mining operations - U.S. Energy Information Administration (EIA)," www.eia.gov, Feb. 01, 2024. https://www.eia.gov/todayinenergy/detail.php?id=61364

[6] I. Eyal and Sirer, Emin Gun, "Majority is not Enough: Bitcoin Mining is Vulnerable," arXiv.org, 2013. https://arxiv.org/abs/1311.0243

[7] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," 2012. Available: https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf

[8] P. Wackerow, "Proof-of-stake (PoS)," ethereum.org, Jan. 26, 2022. https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/

[9] "One year later: How the proof of stake has changed Ethereum," www.home.saxo, Sep. 21, 2023. https://www.home.saxo/content/articles/cryptocurrencies/one-year-later—how-proof-of-stake-has-changed-ethereum-21092023

[10] “Upgrading Ethereum — 2.3.3 LMD Ghost,” Eth2book.info, 2014. https://eth2book.info/capella/part2/consensus/lmd_ghost/#p_59 (accessed Nov. 12, 2024).
[11] “Proof-of-authority (PoA) — ethereum.org,” ethereum.org, 2024. https://ethereum.org/en/developers/docs/consensus-mechanisms/poa/
[12] “Whitepapers,” NEM Documentation, 2019. https://docs.nem.io/pages/Whitepapers/docs.en.html (accessed Nov. 12, 2024).
[13] “Stellar Consensus Protocol (SCP) — Stellar Docs,” developers.stellar.org, Mar. 13, 2024. https://developers.stellar.org/docs/learn/fundamentals/stellar-consensus-protocol
[14] “The Merge brings down Ethereum’s network power consumption by over 99.9%,” Cointelegraph. https://cointelegraph.com/news/the-merge-brings-down-ethereum-s-network-power-consumption-by-over-99-9
[15] “Stellar — Diving into Energy Use on Stellar: Blockchain Payment Efficiency Examined,” stellar.org, Sep. 02, 2021. https://stellar.org/blog/developers/diving-into-energy-use-on-stellar-blockchain-payment-efficiency-examined
[16] J. Schmidt, “Why Does Bitcoin Use So Much Energy?,” Forbes Advisor, Jun. 07, 2021. https://www.forbes.com/advisor/investing/cryptocurrency/bitcoins-energy-usage-explained/
[17] J. Rowlatt, “How Bitcoin’s vast energy use could burst its bubble,” BBC News, Feb. 27, 2021. Available: https://www.bbc.com/news/science-environment-56215787
[18] A. Bada, A. Damianou, C. Angelopoulos, and V. Katos, “Towards a Green Blockchain: A Review of Consensus Mechanisms and their Energy Consumption.” Available: https://eprints.bournemouth.ac.uk/36968/1/GREEN_BLOCKCHAIN.pdf
[19] Athanasios N Nikolakopoulos and John D Garofalakis. Ncdawarerank: a novel ranking method that exploits the decomposable structure of the web. In Proceedings of the sixth ACM international conference on Web search and data mining, pages 143–152. ACM, 2013.
[20] W. Wanecek, “Electricity Consumption of a Distributed Consensus Algorithm,” Lub.lu.se, 2021. https://lup.lub.lu.se/student-papers/search/publication/9059429 (accessed Nov. 23, 2024).
[21] R. Cole and L. Cheng, “Modeling the Energy Consumption of Blockchain Consensus Algorithms,” IEEE Xplore, Jul. 01, 2018. https://ieeexplore.ieee.org/abstract/document/8726725/ (accessed Apr. 06, 2022).
[22] D. Mazières, “The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus.” Available: https://cdn.sanity.io/files/e2r40yh6/production-i18n/39856a57fa0c6e7d646b7db88f48f17688693fe4.pdf?dl=stellar-consensus-protocol.pdf
[23] CoinMarketCap, “Cryptocurrency Market Capitalizations — CoinMarketCap,” CoinMarketCap, 2022. https://coinmarketcap.com
[24] “Stellar — Safety, liveness and fault tolerance—the consensus choices,” *Stellar.org*, Dec. 05, 2014. https://stellar.org/blog/foundation-news/safety-liveness-and-fault-tolerance-consensus-choice (accessed Nov. 23, 2024).
[25] “DEVELOPMENT PLAN AND WHITEPAPER.” Available: https://www.vechain.org/assets/whitepaper/whitepaper-1-0.pdf
[26] J. J. Roberts, “Bitcoin Spinoff Hacked in Rare ’51% Attack’,” *Fortune*, May 29, 2018. https://web.archive.org/web/20190116061454/http://fortune.com/2018/05/29/bitcoin-gold-hack/ (accessed Nov. 23, 2024).