Olympiads in Informatics, 2020, Vol. 14, 47–60

DOI: 10.15388/ioi.2020.04

Consensus Algorithms for Highly Efficient, Decentralized, and Secure Blockchains

Diego I. ESTEVEZ
University of Waterloo
200 University Ave W, Waterloo, ON N2L 3G1, Canada
e-mail: destevez@uwaterloo.ca

Abstract. This paper about blockchain technology introduces its theory, implementation, and applications while focusing on the types of consensus algorithms. The methodology is linguistic and consists of a comparative analysis of the most popular algorithms. This paper is part of a broader effort to make these concepts more accessible and help develop an environment where students can grow and be part of this technological revolution.

Students with a background in algorithmic programming are uniquely suited to tackle highly impactful questions about the algorithms underpinning blockchains. After reading this paper, students should be able to build their own implementation of a blockchain and start doing research into this technology.

Keywords: blockchain, consensus algorithms, peer-to-peer networks, cryptocurrency.

1. Introduction

1.1. *Bitcoin as Leader of the Decentralized Revolution*

Bitcoin has been a very popular topic recently. It's been the promise of a technological revolution while simultaneously a rather controversial concept for governments and banks around the world. It introduced several concepts such as the blockchain and consensus algorithms, which enabled an effective scheme for decentralized ownership of information and transactions through a peer-to-peer network on the internet.

Bitcoin's core ideas can be summarized as transactions made between users without a middle-man and units of the currency directly owned by the individuals, not regulated by a central party like a bank or a company. The latter is a direct consequence of consensus algorithms, which, through cryptography and game theory, the network uses to verify the interactions and spread the changes across the network. Consensus algorithms

replace the need for trust between users and explicit coordination. In fact, in the words of Satoshi Nakamoto, the pseudonym of the inventor of Bitcoin, "What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party." (Nakamoto, 2008)

1.2. *What the Blockchain is*

The blockchain works as a public ledger that stores the interactions that users make within the network. As explained in the Bitcoin whitepaper by Nakamoto, in this system:

- Sending money between parties represents a change in the balance of the nodes involved. This change is referred to as a "transaction," and, as soon as it's made, it is broadcast to all the nodes.
- Members of the network, otherwise known as nodes or peers, store these transactions locally in a pool of unconfirmed transactions. Eventually, they will group these transactions into a block, a container of transactions, which, in Bitcoin's case, has a memory limit of 1MB.
- A blockchain contains a linear chain of blocks, with each of the blocks containing the cryptographic hash of the previous one. Thus, if any block prior to the current one is changed, its hash will change, and the chain of hashes will break.
- After collecting transactions to build a new block, a node has to confirm it and broadcast it to the rest of the nodes. To confirm the block, and the transactions contained in it, the node needs to meet the criteria of the consensus algorithm of the network. In Bitcoin's case, this algorithm is called Proof of Work (PoW) and asks for nodes to find a "nonce" (a number) that, when concatenated to some information in the block, leads to a hash with a specific number of leading zeroes. Therefore, since the node doesn't know which nonce will work beforehand, it needs to spend computational power by iterating through integers to find this number.
- After finding a nonce that works, nodes broadcast the block to the network along with the nonce. The rest of the nodes will receive this information and verify it. Verifying whether the nonce works is much easier than calculating it in the first place. Therefore, it is easy to check that computational power was spent by someone else.
- Bitcoin applies the principle that the longest chain will always be the "correct" one. This means that if the broadcasted block will lead to a longer chain and is verified, nodes will adopt it and add this new block to their blockchains. It's possible when two blocks are broadcasted at the same time that one block gets to some nodes before the other. Eventually, as nodes keep adding blocks, one of the chains will become the longest and will be adopted by the rest of the network.

The fact that the blocks are connected by their hashes allows for immutability. Additionally, anyone can join the network and get a copy of this immutable ledger, meaning that the information there is public and can't be censored.

Through consensus algorithms like PoW and the principle of the longest-chain, Nakamoto was able to solve the problem of double-expenditure, as well as circumventing some of the shortcoming of peer-to-peer networks, such as users/nodes going offline and the possibility of having nodes with different versions of the chain.

1.3. *Why Scalability is so Important*

In this paper's context, scalability means that the blockchain can handle a high transaction rate without clogging. This is key because, otherwise, it may be possible that applications of this technology will not be fully adopted. As a consequence, the capacity for a positive change of this technology in society will be undermined.

A network that fails to scale will negatively affect any participant's experience, thereby decreasing their retention and potentially incentivizing the use of worse alternatives that are useful in the short-run but centralized in the long-run.

2. The Problem

2.1. *Consensus Algorithms*

As previously explained, consensus algorithms are the mechanism that the network uses to confirm transactions. These algorithms pick the node that will add the next block based on a factor: with Proof of Work, a node with t percent of the total computational power of the network, known as hash rate, has a probability of t percent of finding the nonce and having his block confirmed. With other algorithms like Proof of Stake, a node, also known as validator in this context, is selected to add a block based on its total share of the blockchain's currency

As more transactions are made in the network in a short period of time, the size of the pool of unconfirmed transactions grows. Thus, transaction makers include a small amount of cryptocurrency, known as a transaction fee, as an incentive for validators to include the transaction in a block. This is due to the fact that when a validator confirms a block, it collects all the fees of the transactions contained within it. Therefore, the larger the transaction fee, the higher the incentive and the chance that it will be added soon in a block and confirmed.

Additionally, in Bitcoin's Proof of Work, a node is awarded newly generated currency by the system for every new block that it confirms. However, as time passes, these rewards are halved until they eventually converge to 0. At that point, the network's currency reaches the maximum supply that will ever be available. In Bitcoin's case, this number is about 21 million.

2.2. *The Difficulty of this Issue*

Because of the nature of these algorithms and peer-to-peer networks, there's usually a trade-off between scalability, security, and decentralization.

A consensus algorithm that scales is typically less secure since it is likely to be more fault-tolerant and have more relaxed requirements for validators. Higher validation standards results in more challenging tasks that delay the process.

Keeping a system totally decentralized also has an impact on performance since it requires more users to verify a single transaction and mitigate the chance of a node gaining an edge in the verification process and centralizing power. More complicated tasks by the consensus algorithms lead to a smaller set of nodes that can validate, concentrating power in them and taking away decentralization from the network.

An extremely secure system requires an extremely challenging task to confirm a block. Consequently, nodes need more incentives and resources to pursue this challenge of adding a block. Nodes with the most resources will have a higher chance of becoming successful validators. However, this tends to be a very small minority in very secure systems, which negatively affects the decentralization and scalability of the network.

In terms of security, blockchains in general tend to be vulnerable to 3 attacks: In the 51% attack, the malicious agent creates an unofficial copy of the network's blockchain. The agent then makes a transaction only in the official chain. For this transaction, the agent would have exchanged the cryptocurrency for something else. However, the problem arises when this user has so much computational power that, while the transaction still remains unconfirmed in the main chain, it can add blocks on its nonofficial chain faster than the rest of the network on the official chain. Thus, the nonofficial blockchain would eventually become larger than the official one, which, if broadcast, the rest of nodes would adopt based on the principle that the longest chain is the correct one. This means that the agent can now double-spend the money since the unofficial chain, which is the one that nodes would now have, does not contain the transaction that he or she had made. The double expenditure problem is called the Byzantine Generals problem.

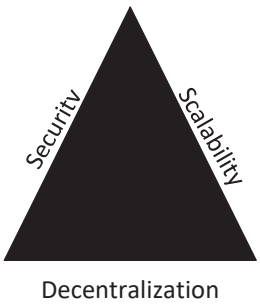

Fig. 1. There's a trilemma of trade-offs between security, scalability, and decentralization in blockchains consensus algorithms.

Another famous attack is Sybil, where the malicious agent tries to fill the network with its own nodes to gain a larger total share. In PoW, the probability of mining the next block is equal to the share of the total computational power. Thus, the number of nodes that the agent controls is irrelevant. What matters is the share of the absolute CPU/GPU power.

Peer-to-peer networks are also vulnerable to denial of service attacks (DoS), where a node is bombarded with packets. In this case, the node gets flooded and cannot operate normally.

2.3. *The Purpose of the Question & Approach*

One of the most pressing questions is which consensus algorithm has the right balance between security, scalability, and decentralization for permissionless, open blockchains. In this qualitative paper, I try to answer that question by providing a linguistic framework to make a comparative analysis of the most popular algorithms that fall in this category.

3. The Implementation

To show how blockchains work, I will develop a basic version from scratch using Python. The snippets shown throughout the document are either in Python 3 or JSON.

Fig. 2 shows a reduced version of a Bitcoin transaction that highlights its most important features: the input amount, the fee for miners, a hash identifying the transaction, and the addresses of both the sender and the recipient. Nodes collect these transactions into their pools of unconfirmed transactions and then select some of these to go into their next block.


```
{
   "hash":"8639c99fafd10ef8cd0b1c0499eb8983b4bc7810642589df374a0f87bb337ff4",
   "received-time":"2020-06-18 21:26",
   "input-amount":10.02440268,
   "fee":0.00006780,
   "sender":"1Ptv5qNTg6bpoMrH8zKqpiSA62jC3i76Nr",
   "recipient":"35EAYWQmq7nwBYqkYNLZKZf5WAY4sXs7BT"
}
```


Fig. 2. A simplified model of a Bitcoin transaction in JSON.


```
{
   "hash":"000000000000000000001072a36c38d3c9e6cf1b3bc85d457606a39830574ba8c0".
   "previous-block":"53cc5f7efb064a603a1dca0ca9747c716ce4862e32e99f762a209b",
   "timestamp":"2020-06-18 22:54",
   "index":635362,
   "nonce":318525442,
   "transactions": {...}
}
```


Fig. 3. A simplified model of a Bitcoin block in JSON.

Fig. 3 is a simplified version of Bitcoin's Block 635362, which stores 2,386 transactions. Since each block contains the hash of the previous block, it would require an enormous power to change a value in a block prior and rebuild the chain. Thus, with every additional block, the chain is reinforced as it would require more power to modify the data.

The snippet in Fig. 4 contains a further simplified version of a transaction and block, as well as functions to instantiate the first block (genesis), add new blocks and transactions, and hash a block.

Since this blockchain is not part of any peer-to-peer network, it doesn't have any consensus algorithm or mechanism to prevent double-expenditure. However, several nodes need to have this chain and be able to synchronize it in real-time. Thus, next, we will see the most popular algorithms and implement Proof of Work, the most simple and common one.

```
from time import time
import hashlib
import json

class Blockchain:
    def __init__(self):
        self.unconfirmed_transactions = []
        self.chain = []
        self.new_block(previous_hash='1', nonce=1) # Create the genesis block

    def new_block(self, nonce, previous_hash):
        block = {
            'index': len(self.chain) + 1,
            'timestamp': time(),
            'transactions': self.current_transactions,
            'nonce': nonce,
            'previous_hash': previous_hash or self.hash(self.chain[-1]),
        }

        # Reset the current list of transactions
        self.current_transactions = []
        self.chain.append(block)
        return block

    def new_transaction(self, sender, recipient, amount):
        self.current_transactions.append({ 'sender': sender,
                                           'recipient': recipient,
                                           'amount': amount})
        return self.last_block['index'] + 1

    @staticmethod
    def hash(block):
        block_string = json.dumps(block, sort_keys=True).encode()
        return hashlib.sha256(block_string).hexdigest()
```

Fig. 4. A Python 3 implementation of a basic blockchain[1].

[1] This implementation is based on that described in the article: Van Flymen, D. (2017, September 25). Learn Blockchains by Building One. From https://medium.com/@vanflymen/learn-blockchains-by-building-one-117428612f46

4. Popular Algorithms

4.1. *Proof of Work*

Proof of Work (PoW) consists of combining the header of the block (that is, the part that includes the hash of the previous block and a Merkle tree) with the nonce to get a hash that meets a condition stipulated by the network. In Bitcoin's case, the task's complexity gets automatically adjusted every 2,016 blocks (or about 14 days) so that one block is confirmed, on average, every 10 minutes. To increase the difficulty, the network demands more zeroes in the hash, thereby exponentially increasing the computational power needed to confirm a block.

Since the first validator, known as miner in PoW, who discovers the nonce receives a reward (typically newly generated currency and all the transaction fees in the block), there's an incentive for nodes to compete to confirm the blocks as fast as possible. The node that can spend the most computational power has the highest chance of finding it first.

Some reasons why the difficulty of the task increases are to make up for advancements in technology that could make the task trivial and to decrease the chance that a nonce is found by chance. Additionally, the fact that finding the nonce requires so much 'work' means that it's hard for a single agent to gather 51% of the computational power of the network and be in the position to implement an attack.

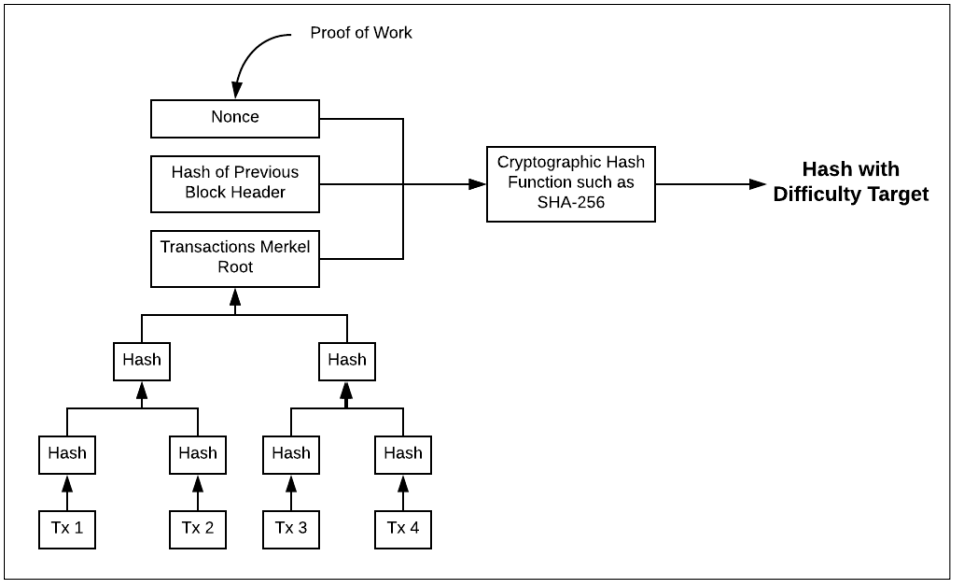

Fig. 5. Diagram illustrating the behaviour of the PoW algorithm, which consists of finding the cryptographic hash of a nonce, the hash of the previous block, and the transaction Merkle root and evaluating the resulting hash against a condition. (Kumar, 2018).

```
class Blockchain:
    ...
    @staticmethod
    def valid_nonce(last_nonce, nonce, last_hash):
        guess = f'{last_nonce}{nonce}{last_hash}'.encode()
        guess_hash = hashlib.sha256(guess).hexdigest()
        return guess_hash[:4] == "0000"

    def proof_of_work(self, last_block):
        last_nonce = last_block['nonce']
        last_hash = self.hash(last_block)

        nonce = 0
        while self.valid_nonce(last_nonce, nonce, last_hash) is False:
            nonce += 1
        return nonce
```

Fig. 6. The implementation of PoW into our basic blockchain model (van Flymen, 2017).

4.1.1. *The Implementation*

To implement PoW, we need the logic to compute nonces and verify them. We will continue using the example above (Fig. 6) that takes a further simplified approach to PoW.

In this fragment, three new functions are present:

- *Hash* computes the hash of a JSON block using SHA-256.
- *Valid_nonce* takes a suspected nonce, the last confirmed nonce, and the hash of the last block. Then, it joins them together, computes the hash, and checks whether the first 4 digits of the hash are all zeroes. This is a slightly different version of PoW than the one described previously, but the same principles still apply.
- *Proof_of_work* receives the hash and the nonce of the last block and looks for the nonce by iterating through a loop.

4.1.2. *Analysis*

Given that PoW requires so much computational power, it is a very safe consensus algorithm. For critical networks like Bitcoin, the probability of implementing a 51% attack is very low because it would require immense computational power. On the other hand, the chances of finding the nonce by sheer luck are also slim.

Scalability-wise, given the limit on block size and frequency, significant traffic increases (i.e., Bitcoin bubble in late 2017) lead to “bottlenecks” in the pool of unconfirmed transactions. This means that it might take many blocks (or days) until a specific transaction gets confirmed. Thus, the average transaction fee also must increase to incentivize miners to include the transaction in a block.

In PoW, decentralization is negatively impacted as the network grows. As the task’s difficulty increases, it becomes more unaffordable for the average user to take part in the verification process because she or he might lack a powerful enough com-

puter to compete. The algorithm centralizes mining power in users who have high-end computers (ASICs mainly) and are located in countries where electricity is comparatively cheap.

The consumption of so much energy for PoW has a negative environmental impact, which cannot be ignored in the long run. In fact, Bitcoin alone (which implements PoW) uses the same amount of energy in a year as Denmark (Vashchuk & Shuwar, 2018). Besides, all this computational work is wasted: finding a nonce doesn't contribute or add to society in any meaningful way. Alternatives have been proposed that make use of this computational power to solve heavy computations in scientific research, thereby contributing to humanity (Wahab & Mehmood, 2018).

PoW is still a relevant algorithm that enabled many of the first applications of blockchain technologies (Bitcoin and Ethereum, to name a few) and provided a very secure method of verifying transactions. However, a system like PoW may be utterly inapplicable to applications that require transactions to be confirmed quickly and can scale to billions of users. A social network on the blockchain using PoW, for instance, would be unfeasible.

In the long-run, mainstream applications won't be able to use PoW because of the system's tendency for centralization, the lack of scalability, and the exponentially increasing costs of computation. PoW can still be very practical for small projects and as a means to generate the currency, but the community must realize that it will have to adopt another system if the application grows significantly. Furthermore, given that electricity won't be paid using the cryptocurrency, the currency's price will be negatively affected as mining costs increase.

4.1.3. *Task*

The best way to fully understand how a consensus algorithm works is by implementing one. At the beginning of the section, a basic model of a blockchain was shown using Python. Now, you can implement PoW using the information on this paper.

4.2. *Proof of Stake*

With PoW, a node has a probability of being selected to mine a block directly proportional to its share of the total computational power of the network. With Proof of Stake (PoS), computational power is replaced by the amount of the cryptocurrency, or tokens, that the node has. In most implementations of PoS, the user has to freeze some funds to be considered by the consensus algorithm as a possible validator, or minter in this context, where the higher these funds are and the longer they have been frozen, the higher the chance that the user is selected to add the next block. Other users have to attest that the block is correct and, if that's the case, the transaction fees of the block are distributed to the minter and the verifiers, and the frozen funds are returned. If the validator adds an invalid block, it is economically penalized by losing a part or all of his

or her frozen funds. Consequently, it is in the best interest of the wealthy nodes, which have the highest chance of being selected, to be honest, as being dishonest would have them lose a significant amount of money. For nodes to be considered as minters, they need to be online 24/7. Any computer with access to the internet can participate.

In PoS, the network takes a virtual approach to consensus. Instead of requiring a physical investment such as CPU/GPU power, the algorithm pseudo-randomly determines it based on a virtual investment, the amount of the native cryptocurrency that the node is willing to freeze. Thus, Proof of Stake is an energetically low-consumption alternative to Proof of Work and seeks to reward the nodes' commitment to the network.

In terms of decentralization, there's undoubtedly a propensity for wealthy nodes to get wealthier, as they would have a higher chance of being selected to validate a block and earning the transaction fees. The system can still be gamed by trying to hold as much currency for as long as possible, but the algorithm by slightly randomizing its choice for minter guarantees a minimum decentralization level and that the same agent will not always be chosen. Some networks, however, require a minimum amount to be staked, limiting an average person's capacity to become a minter. But, most of the time, this barrier tends to be smaller than buying high-end computers and paying significant electricity bills. Thus, there's a lower chance of disparaging economic inequality among the nodes.

Scalability-wise, PoS outperforms PoW. Furthermore, a system called sharding has been proposed that improves the scalability very significantly. In sharding, the network is split into different chains, on which validators work independently, and blocks are processed simultaneously.

However, a significant drawback of this algorithm is that, given that PoW doesn't award minters with newly generated currency like in PoW, there's no way for supply to be created of the cryptocurrency or token. Therefore, to produce the tokens of the blockchain, a variation of PoS should be implemented that also distributes these tokens fairly across the network. An option could be to use PoW to generate currency at the beginning and then switch to PoS once a number of tokens is met.

Unfortunately, pure PoS is less secure than PoW, as the incentive to not work on multiple chains is not present. In PoW, working on parallel chains would require a lot of computational power. In PoS, however, the nodes could simultaneously stake on multiple chains and increase the chance of being selected as minter. To solve this issue, called "nothing at stake," nodes that stake on various chains could be penalized by losing their frozen funds. Another vulnerability was exposed by Bitcoin researcher Andrew Poelstra, who showed that pure Proof of Stake (without any extra logic or procedures) is reversible, leading to the possibility of previous blocks being altered and enabling double-expenditure (Poelstra, 2015). Fortunately, the chance of a 51% attack is less realistic in PoS because the attack would require the agent to acquire a huge percentage of the tokens of the network. Based on economic supply and demand, each extra token purchased would become slightly more expensive. Thus, for blockchains worth billions of dollars, this would cost immensely.

4.3. *Delegated Proof of Stake*

Delegated Proof of Stake (DPoS) is a popular algorithm that consists on the community deciding on the nodes that will add the next blocks. The individuals who possess currency can vote for witnesses and delegates. Witnesses are the nodes that add blocks, and the network keeps a list of the nodes that receive the highest number of votes and picks the top X that will be adding the blocks. This number X changes by blockchain, and the network iterates through the top X witnesses giving each a few seconds to add its block. If the witness doesn't add the block in time, she or he gets penalized. Delegates, who are also voted on, check the validity of these nodes, ensure the network's well-being, and can propose modifications to the blockchain. In DPoS, there's continuous voting of delegates and witnesses.

DPoS positions itself as a system of decentralized governance. Holders of the cryptocurrency or token are the voters whose votes are weighted by how much currency they hold. To incentivize stakeholders to vote for them, witnesses have to share a percentage of the transaction fees they earn with their voters. As a consequence, witnesses have to appear trustworthy and that they will add many blocks and on time. On the other hand, if any party acts dishonestly, it can be immediately expelled by the voters.

This scheme removes the random factor of PoS and concentrates the validators among nodes with the highest credibility. The centralization parameter is trustworthiness, not wealth, which may be beneficial for the network because there's a competition to gain credibility among users. There are no significant barriers to entry, like investing in a high-end computer, and a node doesn't need capital to increase its chance of becoming a witness or delegate. This means that there's likely to be a higher degree of opportunity and economic equality among the nodes. Proponents of DPoS campaign that the fact that it is easy to enter the network means that the system is more decentralized than PoW or PoS. However, this forgets that voters need wealth to have some meaningful voting power. It is the wealthy users that have the most significant influence through their votes and those that essentially control the direction of the network.

As in PoS, the fact that the algorithm is not bound by a physical means or network's rules on block rate improves the capacity for scaling. But, opposite to PoS, the nodes don't have to be online all the time to participate and don't need to have the full chain. Additionally, the fact that minting power is concentrated in a few users significantly improves the rate at which blocks are added. Indeed, networks that select fewer witnesses will be the fastest, but this improvement in scalability will come at the direct cost of decentralization, and, by extension, security.

The fact that the age of coins is not taken into account is beneficial, as it removes the incentive for not moving wealth or trying to hack old, unused accounts that are poorly secured. Furthermore, the fact that voting power is proportional to wealth means that it would be extremely costly to get so much of it. Interestingly, while the "nothing at stake" problem is still present with this algorithm, if a user participates in multiple chains simultaneously, it will harm his or her reputation, essentially socially penalizing this user for the behavior.

For the entire system to work correctly, there needs to be an active, unorganized base of stakeholders that votes for the witnesses and delegates in the network's best interest. However, there is no guarantee of this. In fact, the tendency of the network towards centralization risks security, as it incentivizes the creation of cartels or groups to conspire together. For witnesses and delegates, this would be much more viable as there are few of them, and, by conspiring together, they could get away with malicious acts like double-spending. Networks that have a lower requirement of delegates to validate a block are especially vulnerable to one of these attacks, as it would be easier to meet that number. This phenomenon can also be extended to voters, who could conspire or be bribed to elect specific delegates and witnesses. There's certainly nothing that assures that the witnesses and delegates will never act maliciously.

4.3.1. *Solution as a Layer 2*

DPoS scales much better than PoS or PoW and can allow many applications that require several transactions per minute, such as blockchain video games (i.e., Crypto Kitties) or social networks (i.e., Steemit). However, DPoS should only be used for applications that can sacrifice security and decentralization for scalability. DPoS should not be the foundational consensus algorithm for systems that need very significant security and that must be trustless between nodes, such as those that manage financial transactions or health certificates.

An advantageous approach is to use DPoS as a Layer 2. In this scheme, a foundational blockchain runs a consensus algorithm like PoS or PoW and another blockchain linked to it, such as DPoS, serves to give scalability. Since PoW and PoS are relatively safe algorithms, there's reasonable level of integrity and security offered by the base chain. If the blockchain is attacked, the base chain can be used to revert to the last healthy block.

In my opinion, the community will inevitably have to resort to a Layer 2 solution in the long-run for many applications. Through hybrid protocols like this one, a stable network is achieved, where we can enjoy the benefits of the ultra scalability of DPoS while maintaining a sensible security level.

5. Conclusion

In my opinion, these algorithms serve different purposes, which, in large part, make it difficult to replace one by another.

Proof of Work is an algorithm that maximizes security to the point that is impossible with another consensus algorithm, and it is the easiest to maintain and implement. However, it becomes inefficient in the long run and cannot support a network like Bitcoin if the rate at which the number of users grows is maintained. Thus, I see PoW as an alternative that can serve to back financial systems and cryptocurrencies in the short/medium run, but it is a system limited by its own nature. Additionally, while it generates decentralization in the short term, it ultimately leads to centralization. By rewarding

clusters of high-performing computers, the system creates an incentive for the formation of cartels and groups that centralize block mining and validation.

Proof of Stake allows for long-term scalability by sacrificing some decentralization and security. There's likely to be less mobility of the tokens between the nodes as they are incentivized to freeze funds for as long as possible. The fact that wealthy nodes are likely to become wealthier is a risk for many cryptocurrencies, especially those that have a small market capitalization. For these blockchains, it would be cheap to buy a high percentage of the supply.

The Ethereum network, a blockchain for hosting decentralized applications (Dapps), is moving from PoW to Casper, their own implementation of PoS. Their objective is to increase the scalability of the network, which currently consists of 11,000 nodes and a $21 billion market capitalization; with PoW, the system is unable to sustain the Dapp ecosystem in the long run. Thus, the developers are willing to sacrifice some of the network's security and decentralization to achieve the project's purpose.

Delegated Proof of Stake is too risky for networks like Ethereum and Bitcoin that store so much value and still seek to maintain a general degree of decentralization. Nevertheless, DPoS can improve the scalability of blockchains by orders of magnitude, thereby enabling crucial applications for the growth of the blockchain ecosystem. Using a hybrid protocol that combines DPoS and PoW could be a very promising option for applications that still need security and scalability.

Indeed, each algorithm has its trade-offs and advantages and lies somewhere in a spectrum where there's no absolute "best." While more variations and new protocols will emerge, PoW will likely continue being the most popular option for its simplicity and security. On the other hand, PoS has more use-cases and seems to provide the most reasonable balance between decentralization, scalability, and security. It would not generate bottlenecks in the medium/short run, and security is usually not at high risk in most medium or small scale projects. Still, since it can't be used to generate the initial supply of the currency, another system like PoW will have to be used at the beginning.

With the information in this paper, students with a background in programming should have a clear picture of the state of the art of blockchain technologies and an understanding of the theory, implementation, and applications of these technologies.

In the future, problems related to blockchain architecture and algorithms could serve as an inspiration and a training resource for Olympiads because of their potential to be intellectually stimulating and to contribute to our knowledge and society.

References

De Quénetain, S. (2017). Delegated Proof of Stake: The crypto-democracy. Retrieved June 25, 2020, from `http://www.blockchains-expert.com/en/delegated-proof-of-stake-the-crypto-democracy-2`

Ferdous, M. S., Chowdhury, M. J. M., Hoque, M. A., & Colman, A. (2020). Blockchain consensus algorithms: A survey. ArXiv:2001.07091 [Cs]. Retrieved from `http://arxiv.org/abs/2001.07091`

Konstantopoulos, G. (2020). Understanding Blockchain Fundamentals, Part 2: Proof of Work & Proof of Stake. Retrieved June 25, 2020, from `https://medium.com/loom-network/understanding-blockchain-fundamentals-part-2-proof-of-work-proof-of-stake-b6ae907c7edb`

Kore, A. (2018). Building a blockchain. Retrieved June 25, 2020, from https://medium.com/@akshaykore/building-a-blockchain-7579c53962dd

Kumar, A. (2018). Fig. 2. Proof of Work in Bitcoin Blockchain [Digital image]. Retrieved June 26, 2020, from www.vitalflux.com/bitcoin-blockchain-proof-work

Larimer, D. (2017). DPOS Consensus Algorithm - The Missing White Paper. Retrieved June 25, 2020, from http://www.steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper

Li, C., & Palanisamy, B. (2020). Comparison of decentralization in dpos and pow blockchains. ArXiv:2002.02082 [Cs]. Retrieved from http://arxiv.org/abs/2002.02082

Panda, S. S., Mohanta, B. K., Satapathy, U., Jena, D., Gountia, D., & Patra, T. K. (2019). Study of Blockchain Based Decentralized Consensus Algorithms. *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*. DOI:10.1109/tencon.2019.8929439

Poelstra, A. (2015). On Stake and Consensus.

Thin, W. Y., Dong, N., Bai, G., & Dong, J. S. (2018). Formal Analysis of a Proof-of-Stake Blockchain. *2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*. DOI:10.1109/iceccs2018.2018.00031

Van Flymen, D. (2017). Learn Blockchains by Building One. Retrieved June 26, 2020, from https://medium.com/@vanflymen/learn-blockchains-by-building-one-117428612f46

Vashchuk, O., & Shuwar, R. (2018). Pros and cons of consensus algorithm proof of stake. Difference in the network safety in proof of work and proof of stake. *Electronics and Information Technologies, 9*. DOI:10.30970/eli.9.106

Wagner, K., Keller, T., & Seiler, R. (2019). A Comparative Analysis Of Cryptocurrency Consensus Algorithms. *Proceedings of the 16th International Conference on Applied Computing 2019*. DOI:10.33965/ac2019_201912l026

Wahab, A., & Mehmood, W. (2018). Survey of consensus protocols. ArXiv:1810.03357 [Cs]. Retrieved from http://arxiv.org/abs/1810.03357

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

D. Estevez is a 19-year-old Mathematical Physics student at the University of Waterloo. In the past, he founded a social network for citizen journalism, an NGO that was funded by Microsoft, Uber, and Dell, and sent capsules to the stratosphere that broke several records. He's also doing research at Democracy Mars into blockchain governance.