COMPARATIVE ANALYSIS OF BLOCKCHAIN CONSENSUS ALGORITHM; POW VS POS BASED ON PERFORMANCE AND SECURITY METRICS

OJEKUDO, NATHANIEL AKPOFURE, PhD.
Department of Computer Science
Ignatius Ajuru University of Education
Rumuolumini Port Harcourt, Rivers State
Nigeria

Abstract

Cryptocurrency has attracted great attention in academic, financial and well as technology. Behind the cryptocurrency is the Blockchain, which is a decentralized digital ledger which records all verified transaction on the network. Transactions within the blockchain are verified using one of many consensus algorithms to resolve the problem of reliability involving many distributed nodes which keep copies of the ledger. This paper presents a comprehensive overview on blockchaintechnology and consensus problems. Since the Proof of Work (PoW) and Proof of Stake (PoS) are the most widely used consensus algorithm amongst cryptocurrency. This study attempts to comparatively analyze typical PoW and PoS consensus algorithm based on performance and security metrics. Finally, we present our conclusion based on our analysis, which suggests that PoS algorithm appears to perform better than PoW algorithm especially on performance metrics such as transaction rate, cost and energy and scalability. However, PoW performs better on security metrics such as; cost of attack, coin age accumulation attack, precomputing attack, nothing at stake problem and initial distribution problem. Future Blockchain Consensus algorithm should consider implementing innovative improvements on both performance and security metrics.

KEYWORDS: Cryptocurrency, Proof of Work, Proof of Stake

Introduction

Bitcoin since being introduced in by Nakamoto (2008), it has become a global decentralized cryptocurrency now and led to more than 1633 alternative coins (CoinmarketCap, 2018). The total venture capital of cryptocurrency reached over 248 billion USD at the end of June, 2018 (CoinmarketCap, 2018). Cryptocurrency has attracted great attention in academic, financial and well as technology. The core technology under Bitcoin is the Nakamoto consensus protocol, which plays a key role in maintaining the transaction history of Bitcoin in a public distributed ledger called the blockchain (Gao and Nobuhara, 2017).

Critical to the operation of a distributed ledger is ensuring the entire network collectively agrees with the contents of the ledger; this is the job of the consensus algorithm. The function of a consensus algorithm is to verify that information being added to the blockchain is valid. That is the network is in consensus. This ensures that the next block being added represents the most current transactions on the network, preventing double spending and other invalid data from being appended to the blockchain. In addition, the consensus mechanism keeps the network from being derailed through constant forking. There have been a number of different consensus algorithm devised, each with their own pros and cons. They all serve the same core purpose as described above but differ in methodology. The critical difference between varying consensus algorithm is the method in which they delegate and reward the verification of transactions.

The most popular blockchain implementation of a distributed and trustless consensus algorithm consensus algorithm are the Proof of Work (PoW) and Proof of Stake (PoS) systems (Asolo, 2018). This paper will focus on describing and comparing PoW vs PoS. Note however, that a number of other systems exist, such as Delegated Proof of Stake (DPoS) and Federated Byzantine Agreement (FBA), Proof of Authority, Proof of Importance, etc (Schumann, 2018).

Statement of the Problem

As the cryptocurrency market continues to grow and create awareness among people, institutions and nationalities, a lot of interest have arisen on how the blockchain can be adapted to solve most of the real life problems. Blockchain is a decentralized or distributed ledger, which is the underlying database structure for transactions of Bitcoin and other digital cryptocurrency currencies (BitFury Group, 2015). A key feature of the blockchain is the distributed ledger, which contains a record of all previous transactions. This ledger is called a distributed ledger because it is not stored in a central location, rather it is stored across a network of computer across the world.As such, there exists the problem of reliability, security and performance as many distributed nodes keep copies of the digital ledger.

Aim and Objectives of the Study

The aim of this study is to comparatively analyze blockchain consensus algorithm

Objectives of the Study are:

- To explain Blockchain and its architecture.
- To review Blockchain Consensus Algorithms, Proof of Work and Proof of Stake.
- To review Implementations of PoW and PoS
- Comparatively Analyze Proof of Work and Proof of Stake.

Blockchain Architecture

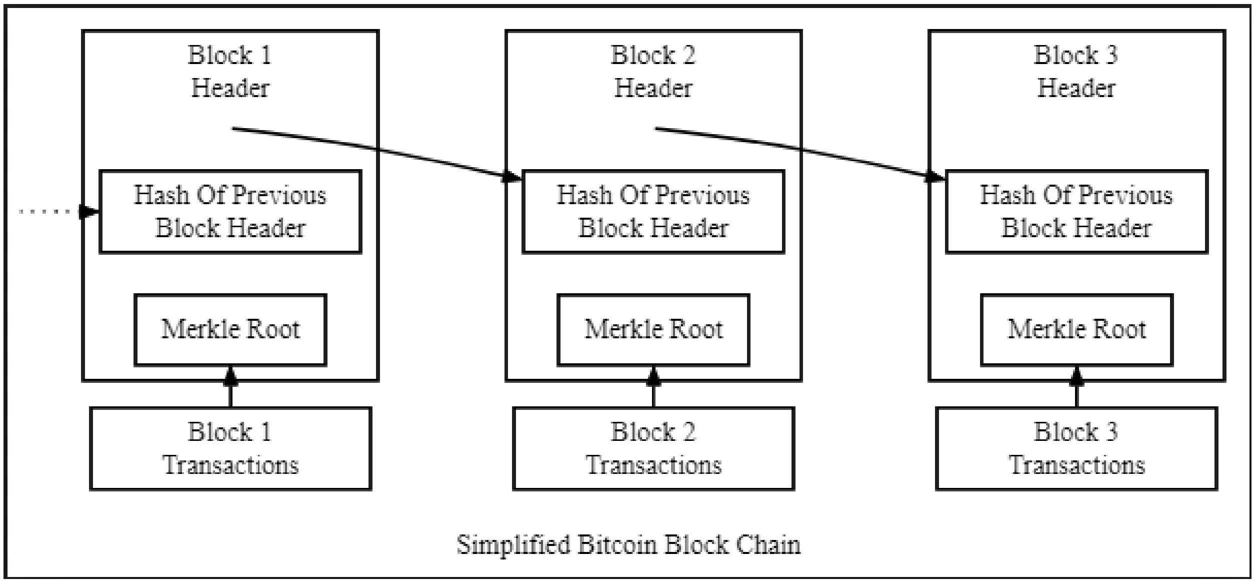


Figure 1-Blockchain which contains sequence of blocks (Bitcoin Developer Guide, 2018)

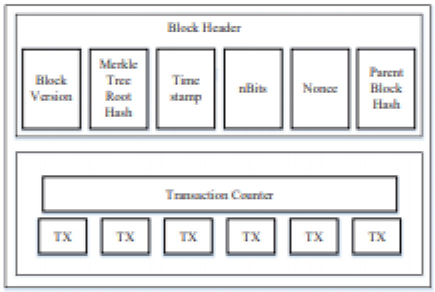


Figure 2- Blockchain Structure (Zheng et al, 2017)

Blockchain is a sequence of blocks, like conventional public ledger which holds a complete list of transaction records (Lee KuoChuen, 2015). Figure 3 above illustrates an example of a blockchain. With a previous block hash contained in the block header of the preceding block, a block has only one parent block. The first block of a blockchain is called genesis block which has no parent block (Zheng et al, 2017). We then explain the blockchain architecture components in details.

A. Block

A block consists of the block header and the block body as shown in Figure 2. In particular, the block header includes:

(i) Block version: indicates which set of block validation rules to follow.

(ii) Merkle tree root hash: the hash value of all the transactions in the block.

(iii) Timestamp: current time as seconds in universal time since January 1, 1970.

(iv) Bits: target threshold of a valid block hash.

(v) Nonce: a 4-byte field, which usually starts with 0 and increases for every hash calculation.

(vi) Parent block hash: a 256-bit hash value that points to the previous block. The body of the block is composed of a transaction counter and transactions. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction. Blockchain uses an asymmetric cryptography mechanism to validate the authentication of transactions (Nomura Research Institute, 2015).

B. Digital Signature

Digital signature based on asymmetric cryptography is used in an untrustworthy environment.

Each user owns a pair of private key and public key. The private key that shall be kept in confidentiality is used to sign the transactions. The digital signed transactions are broadcasted throughout the whole network. The digital signature involves two phases: signing phase and verification phase. For instance, a user Bob wants to send another user Alice a message.

1) In the signing phase, Bob encrypts her data with her private key and sends Alice the encrypted result and original data.
2) In the verification phase, Alice validates the value with Alice's public key. In that way, Bob could easily check if the data has been tampered or not.

C. Key Characteristics

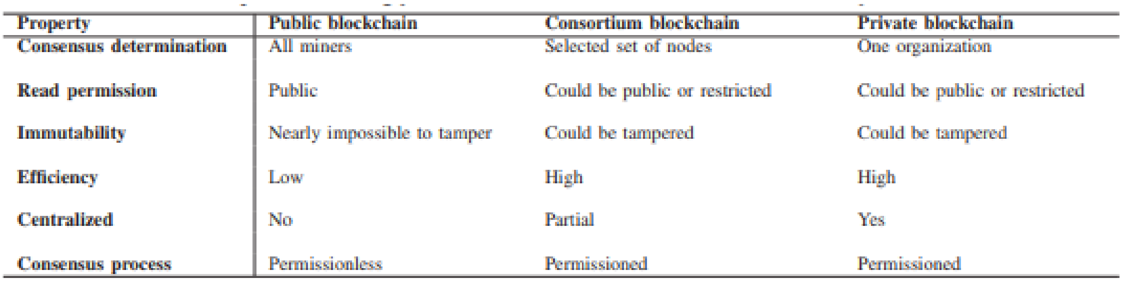
Basically, blockchain has following key characteristics.

- Decentralization. In conventional centralized transaction systems, each transaction needs to be validated through the central trusted agency (e.g., the central bank), inevitably resulting to the cost and the performance bottlenecks at the central servers. Contrast to the centralized mode, third party is no longer needed in blockchain. Consensus algorithms in blockchain are used to maintain data consistency in distributed network.
- Persistency. Transactions can be validated quickly and invalid transactions would not be admitted by honest miners. It is nearly impossible to delete or rollback transactions once they are included in the blockchain. Blocks that contain invalid transactions could be discovered immediately.
- Anonymity. Each user can interact with the blockchain with a generated address, which does not reveal the real identity of the user. Note that blockchain cannot guarantee the perfect privacy preservation due to the intrinsic constraint (Zheng et al, 2017).

- Auditability. Bitcoin blockchain stores data about user balances based on the Unspent Transaction Output (UTXO) model: Any transaction has to refer to some previous unspent transactions. Once the current transaction is recorded into the blockchain, the state of those referred unspent transactions switch from unspent to spend. So transactions could be easily verified and tracked.

Taxonomy of blockchain systems

Today, blockchain systems are categorized generally into three types: public, private and consortium blockchain (Buterin, 2015). For public blockchain, all records are visible to the public and everyone could take part in the consensus process. Differently, only a group of pre-selected nodes would participate in the consensus process of a consortium blockchain. As for private blockchain, only those nodes that come from one specific organization would be allowed to join the consensus process. A private blockchain is regarded as a centralized network since it is fully controlled by one organization. The consortium blockchain constructed by several organizations is partially decentralized since only a small portion of nodes would be selected to determine the consensus. The comparison among the three types of blockchains is listed below in Table I.



Property	Public blockchain	Consortium blockchain	Private blockchain
Consensus determination	All miners	Selected set of nodes	One organization
Read permission	Public	Could be public or restricted	Could be public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralized	No	Partial	Yes
Consensus process	Permissionless	Permissioned	Permissioned

Table 1 - Comparison among public, consortium and private (Zheng et al, 2017)

- *Consensus determination*. In public blockchain, each node could take part in the consensus process. And onlya selected set of nodes are responsible for validating the block in consortium blockchain. As for private chain, it is fully controlled by one organization and the organization could determine the final consensus.
- *Read permission*. Transactions in a public blockchain are visible to the public while it depends when it comes to aprivate blockchain or a consortium blockchain.
- *Immutability*. Since records are stored on a large number of participants, it is nearly impossible to tamper transactionsin a public blockchain. Differently, transactions in a private blockchain or a consortium blockchain could be tampered easily as there are only limited number of participants.
- *Efficiency*. It takes plenty of time to propagate transactions and blocks as there are a large number of nodes on public blockchain network. As a result, transaction throughput is limited and the latency is high. With

fewervalidators, consortium blockchain and private blockchaincould be more efficient.

- *Centralized*. The main difference among the three types of blockchains is that public blockchain is decentralized, consortium blockchain is partially centralized and private blockchain is fully centralized as it is controlled by asingle group.
- *Consensus process*. Everyone in the world could join the consensus process of the public blockchain. Differentfrom public blockchain, both consortium blockchain and private blockchain are permissioned.

Since public blockchain is open to the world, it can attract many users and communities are active. Many publicblockchains emerge day by day. As for consortium blockchain, it could be applied into many business applications. Currently Hyperledger (Zheng et al, 2017)is developing business consortium blockchain frameworks. Ethereum also has provided tools for building consortium blockchains (Ray, 2018).

The Consensus Problem

The consensus is a problem in distributed computing wherein nodes within the system must reach an agreement given the presence of faulty processes or deceptive nodes.

A. The Byzantine Generals Problem

The Byzantine Generals Problem is a problem concerning communication failure (Lamport, Shostak & Pease, 1982). Briefly, how can each node ("general") in a system be certain that the information they are receiving is valid?

In the original problem, the situation of n Byzantine generals preparing to attack a fort is proposed. Each general has the option to attack the fort or retreat; however, it is vital that all generals agree upon the same course of action, as a half-hearted attack would be disastrous. To complicate matters, the generals are far apart, only able to communicate through messengers, which may not successfully deliver their messages, and some of these generals are traitorous and will actively attempt to deceive the others (Bach, Mihaljevic & Zagar, 2018).

B. Byzantine Fault Tolerance (BFT)

Byzantine Fault Tolerance (BFT) is a category of replication algorithms that attempts to solve the problem of reaching consensus when nodes can generate arbitrary data. Castro & Liskov (2002) described BFT can guarantee the safety (the chance that something negative will happen in the system) and liveness (the chance that progress will be made within the system) of a system given that no more than

$\lfloor (n - 1) \div 3 \rfloor$(1)

replicas are faulty over the system's lifetime, where n is the total number of replicas within a system. BFT can handle up to 33% of nodes being faulty. Typically, up to

$3f + 1$(2)

replicas in order to provide safety and liveness in a system, where f is the total number of faulty replicas contained within said system. Correia,

Veronese & Lung (2010) however, suggests at least one known BFT implementation is able to reduce this to
required replicas. 2f + 1

C. Delegated Byzantine Fault Tolerance (dBFT)

Delegated Byzantine Fault Tolerance (dBFT) is a variant of standard BFT. In the NEO whitepaper (NEO, 2018), this fault tolerance algorithm splits clients within a Peer to Peer system into two separate types: bookkeepers and ordinary nodes. Ordinary nodes do not take part in determining consensus but, rather, vote (hence the "delegated") on which bookkeeper node it wishes to support. The bookkeeper nodes that were successfully elected are then included in the consensus process.

In this process, a random bookkeeper node is selected to broadcast its transaction data to the entire network. Should at least 66% of the other bookkeepers agree that the transaction data is valid, it is committed permanently to the blockchain and another round of consensus is started with another randomly selected bookkeeper (Bach, Mihaljevic & Zagar, 2018).

High-Profile Consensus Algorithm Implementation

Currently, there exist over 1,600active cryptocurrencies (that is, actively tradeable on the global market), "high-profile" in this context is indication of a cryptocurrency's market cap. Although cryptocurrency market values are in a constant flux state, this ranking schema was determined to be the fairest in ordering the currencies (and the algorithms behind them).

Currency Name	**Consenus Algorithm**	**Market Cap**
Bitcoin	Proof ofWork	$106.6B
Ethereum	Proof ofWork*	$43.6B
BitcoinCash	Proof ofWork	$21.4B
Cardano	ProofofStake	$3.5B
Nxt	ProofofStake	$81M

Table 2 - Top 100 cryptocurrency (CoinmarketCap, 2018)

*PlannedswitchtoProofofStakesometimein2018

1) Proof of Work (PoW)

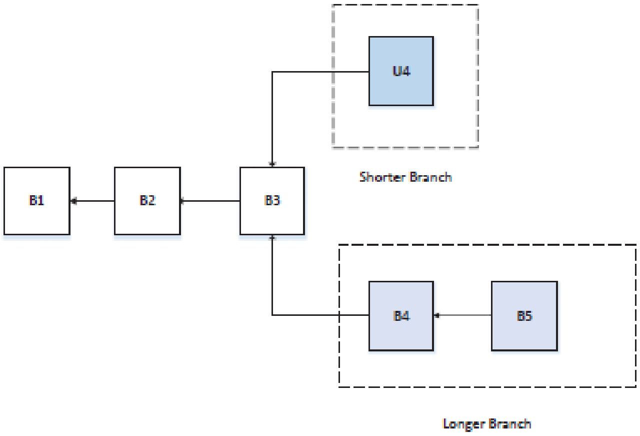
In a decentralized network, a node has to be selected to record the transactions. The easiest way is random selection. However, random selection is vulnerable to attacks. Proof of Work (PoW) was the first blockchain consensus algorithm used in blockchain (Witherspoon, 2018).The Bitcoin White Paper proposed the use of a Proof of Work system to prevent an entity from gaining a majority control over the network(Schumann, 2018).

In Proof-of-Work model, for a node to publish a block of transactions, a lot of computational work has to be done to prove that the node is not likely to attack the network. Zheng et al, 2017. The computational work must be difficult for the client but easy for the server/network to verify (Greenfield, 2017).

The type of computational work miners must solve has the following key features that define the Proof of Work system:

- The puzzles are asymmetric, meaning it is difficult for miners to solve but the correct answer is easily verified by the network (Greenfield, 2017).
- The computational work have no skill involved, they require brute force. The only way for a miner to improve their odds of solving a puzzle is to acquire additional computational power; something that is very energy and capital intensive.
- The computational work parameters are periodically updated in order to keep the block time consistent. The Bitcoin protocol, for example, has a block generation target time of 10 minutes. For instance, if the average block time over two weeks has decreased to below 10 minutes, the network will automatically increase the difficulty. This increases the number of calculations and the average time required for the puzzle to be solved.

According to Nakamoto (2008) the PoW model require all blocks, with the exception of the first block created by the system (the "genesis block"), have a hash which consists of the previous block's hash alongside the nonce required to create the necessary zero bits. The genesis block is an exception as it has no previous block to point to: its hash is entirely zeroes.



.

Figure 3 - Blockchain Fork (Zheng et al, 2017)

In Fig. 3 above, a scenario of blockchain fork (the longer branch would be admitted as the main chain while the shorter one would be deserted) calculations. In PoW, each node of the network is calculating a hash value of the block header. The block header contains a nonce and miners would change the nonce frequently to get different hash values. The consensus requires that the calculated value must be equal to or smaller than a certain given value. When one node reaches the target value, it would broadcast the block to other nodes and all other nodes must mutually confirm the correctness of the hash value. If the block is validated, other miners would append this new block to their own blockchains. In Bitcoin, nodes that calculate the hash values are known as *miners* and the PoW procedure are called *mining*.

In the decentralized network, valid blocks might be generated simultaneously when multiple nodes find the suitable nonce nearly at the same time. As a result, branches may begenerated as shown in Figure 3. However, it is unlikely that two competing forks will generate next block simultaneously.

In PoW protocol, a chain that becomes longer thereafter is judged as the authentic one. Consider two forks created by simultaneously validated blocks U4 and B4. Miners keep mining their blocks until a longer branch is found. B4, B5 forms a longer chain, so the miners on U4 would switch to the longer branch Miners have to do a lot of computer calculations in PoW, these works waste too much resources (Zheng et al, 2017).

2) Proof of Stake (PoS)

PoS (Proof of stake) is an energy-saving alternative consensus algorithm to PoW. In PoS, consensus is achieved in a deterministic way, where the validator of a new block is chosen, depending on its wealth, also defined as stake or by coin age. Another key distinction about Proof of Stake is that under Proof of Stake there is no new coin creation (mining). Instead, all of the coins are created in the very beginning. This means that in the PoS models there is no block reward, so, the miners take the transaction fees as opposed to newly minted coins (Schumann, 2018).

With Proof of Stake, there is no computational work, instead, the Validator of a new block is chosen in a deterministic way based on their stake. The stake is how many coins/tokens one possesses. For example, if one-person **A** were to stake 100 coins and another person **B** staked 500 coins, the person staking 500 coins would be 5 times more likely to be chosen as the next block validator.

Unfortunately, as the mining cost is nearly zero, attacks might come as a consequence. Miners in PoS have to prove the ownership of the amount of currency. It is believed that people with more currencies would be less likely to attack the network. However, the selection based on account balance is quite unfair because the single richest node is bound to be dominant in the network. As such, many solutions are proposed with the combination of the stake size to decide which one to forge the next block.

The target amount that a validator needs to contribute in order to mint a new block is determined by the system under the following condition:

$$proofhash < coins x age x target$$

the ***proofhash*** is the obfuscation sum that depends on a stake, the unspent output, and the current time. *Coins* are the number of coins a miner has spent for the mining privilege, ***age*** is the age of the coins that have been spent and ***target*** is the required amount of coins specified by the network through a network a difficulty adjustment process similar to PoW's implementation.

Nxt and Cardano are example of Implementation PoS as given in table 4 above.

Nxt is a pure proof of stake cryptocurrency that began in 2013. Unlike Peercoin, Nxt does not use proof of work to create new coins; the entire available supply of 1 billion coins (NXTs) was present in the system from the genesis block. Thus, the only incentive to mint blocks is to collect transaction fees (BitFury Group, 2015).

Cardano's Ouroboros - The Inventors of Cardano's Ouroboros protocol took the PoS algorithm described above and added additional security measures to ensure persistence and liveness within their system (Kiayias, Russell, David & Oliynykov, 2017). Namely, as described in the whitepaper, this implementation includes a delegation process for the electing of stakeholders and takes snapshots of current stakeholders in what they have labeled an "epoch." Each epoch, new stakeholders are elected by having a subset of the current stakeholders randomly decide who the stakeholders will be in the next epoch (Bach, Mihaljevic & Zagar, 2018).

Comparative Analysis of Proof of Work and Proof of Stake

Different consensus algorithms have different advantages and disadvantages. Table 3 below gives a comparison between different PoW and PoS consensus algorithms based on a set of performance and security metrics.

	Metric	**POW**	**POS**
Transaction Rate (TPS)	Performance	Low, <100	High, <1000
Cost of Paticipation		Computer Power	nodes to buy some initial cryptocurrency
Scalability of Peer Network		High	Very High
Requirement for Operation		Computing Power	Coin Age or Stake
Cost and Energy		High	Low
Initial Distribution Problem	Security	Insignificant	Very significant
Selfish Mining		Susceptible	Impossible
Cost of Attack		Very Expensive and Difficult to achieve	Less Difficult and Expensive to achieve
Coin Age Accumulation Attack		Impossible	Possible
PreComputing Attack		Impossible	Possible
Nothing at Stake Problem		Impossible	Possible
Adversary Tolerance		<=25% computing power	<51% stake

Table 3 - Comparison of POW and POS algorithms based on performance and security metrics.

A. Performance Metrics

a. **Transaction rate or Transaction per Second** – Transaction rate theoretically considers number of transactions that can be Transaction rate is higher with platforms that can confirm transactions immediately and reach consensus fast. PoW approaches are probabilistic and have to spend significant amount of time solving the cryptographic puzzle. Therefore, these models have high transaction latencies and therefore a low transaction rate of typically <100. PoS can confirm transactions fast and are expected to support high transaction rates. POS have higher transaction rate of <1000. (Mingxiao et all, 2017).

b. **Cost of participation** - PoW requires expending energy, which is a resource that is external to the consensus protocol, while PoS requires nodes to buy some initial cryptocurrency to generate a security deposit for declaring interest and bonding with the platform.

c. **Scalability of peer network** - Scalability of the consensus models is its ability to reach consensus when number of peering nodes are constantly increasing. POS have higher scalability compared to POS.

d. **Requirement for operation of Blockchain -** POW blockchain requires miners to have high computing power to solve difficult computer problems to create new blocks. POS blockchain selects the validator to mint new blocks based on the coin age or stake of the token that possess.

e. **Cost and Energy -** The computational PoW required to operate a Proof of Work system is very energy intensive. The Bitcoin network, for example, requires an annual energy consumption comparable to that of Colombia (57.6 TWh annually) (Digiconomist, 2018). In addition, the competitive nature of mining means an increasing amount of money is being invested into more powerful mining computers, which in turn will require more and more energy to be supplied. As such the role of mining is becoming increasingly reserved for large-scale operations. On the other hand, Proof of Stake systems do not require mining or the accompanying energy hungry processing power. As a result, Proof of Stake systems require a mere fraction of the energy to run. The lower energy costs also make the role of validating more accessible to anyone in the community, whereas the role of mining is becoming increasingly reserved for large-scale operations.

B. Security Metrics

f. **Initial Distribution Problem -** In a proof of stake system, there is always a concern that the initial holders of coins will not have an incentive to release their coins to third parties, as the coin balance directly contributes to their wealth. In Bitcoin and other PoW systems, early adopters of the technology are at the same position as the rest of the users: in order to mine coins, they need to improve their hardware continuously and optimize resource consumption. In a proof of stake system, a user that acquired 10% of the coins when the system was just launched (e.g., for $1,000) is at an advantage compared to users with the same funds when the system has gained popularity and $1,000 translates to 0.01% of all coins (BitFury Group, 2015).

g. **Selfish Mining–** In selfish mining, an attacker selectively reveals mined blocks in order to waste computational resources of honest miners. Selfish mining is specific to POW consensus. Since there are no expensive resources involved in block generation in the case of PoS consensus, the attack is ineffective for PoS currencies (BitFury Group, 2015).

h. **Cost of Attack –** To execute a successful attack with 100% probability, the attacker needs to control more than 50% of the resources used to secure the system (computational power in case of PoW; liquidity in case of PoS) for the duration of the attack. Thus, in the case of proof of stake, the attacker does not need to own more than a half of the currency – he only needs to gain the privilege of accessing 50% of the currency in circulation. For instance, an attack on Bitcoin lasting for 1,000 blocks would require $4 million at very least (and, unlike a short range attack, it would be highly visible as observed network hash rate would drop in half for an extended time). In earlier versions of proof of stake, the cost of attack would be much lower (BitFury Group, 2015).

i. **Coin Age Accumulation Attack –** For POS blockchain like Peercoin and other systems using coin age instead of wealth as a measure of user's stake. In the first versions of Peercoin blockchain, coin age was uncapped. This means that by waiting long enough, an attacker could potentially accumulate enough coin age to effectively overtake the network. For instance, an attacker owning 5% of all coins could split his money into multiple outputs and wait until the age of his coins would become 10 times more than average. After that, the attacker could mint multiple blocks in sequence with high probability to perform double-spending or other malicious activity. If multiple users are attempting this attack, it would lead to deterioration of the network.

j. **Precomputing Attack –** These type of attack are not possible with POW model, but are very possible with PoS model.

k. **Nothing at Stake Problem –** With a proof of work algorithm, such behavior is irrational. By splitting the resources on multiple branches, a miner diminishes the probability to find a block on each of them; the optimal strategy in a PoW system is always to mine on a single branch. The basic proof of stake algorithm does not discourage forking. When a blockchain forks, the rational behavior for all users of the network is to mint blocks on both branches. The probability to find a block does not decrease if the user is attempting to mint on multiple blockchain branches. As such a validator will receive a duplicate copy of their stake on the newly forked blockchain (Schumann, 2018). If a validator signs off on both sides of the fork, they could potentially claim twice the amount of transactions fees as a reward and double spend their coins.

l. **Adversary tolerance** – The fraction of the network that can be compromised without the consensus being affected**.** PoW systems could help miners to gain more revenue by only 25% of the hashing power, while PoS require 51%.

Conclusion

This paper has reviewed the blockchain architecture, consensus problem. The two most popular implementation of consensus problem-PoW and PoS models were examined. A comparative analysis was conducted on PoW and PoS based on performance and security metrics. PoS blockchain appears to perform better than PoW blockchain especially on performance metrics such as*transaction rate, cost and energy and scalability*. However, PoW performs better on security metrics such as; *cost of attack, coin age accumulation attack, precomputing attack, nothing at stake problem* ***and*** *initial distribution problem* When adopting Blockchain to solve a business problem, it is important to look at the scale of the intended network, the relationships between participants, both functional and non-functional aspects (such as performance and confidentiality) and security before determining the right platform and the right consensus model to use.

Future Blockchain Consensus algorithm should consider implementing innovative improvements on both performance and security metrics. It is hoped that this paper sheds light on the blockchain consensus algorithms, clearly highlighting the strength and weakness of POW and POS consensus models and finally helps in decision making.

References

Asolo, B. (2018). Blockchain Consensus Algorithms Explained. *Mycryptopedia*.

Bach, L., Mihaljevic, B., & Zagar, M. (2018). Comparative analysis of blockchain consensus algorithms. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* Opatija,: IEEE.

Bitcoin Developer Guide. (2018). *Simplified Bitcoin Blockchain.*

BitFury Group. (2015). *Proof of Stake versus Proof of Work*. Version 1.0. BitFury Group.

Buterin, V. (2015). On Public and Private Blockchains. *Ethereum Blog.*

Castro, M., & Liskov, B. (2002). Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems*, *20*(4), 398-461. doi:10.1145/571637.571640

CoinmarketCap. (2018). Crypto Currency Market Capitalizations. *Coinmarketcap.com.*

Correia, M., Veronese, G., & Lung, L. (2010). Asynchronous Byzantine consensus with 2f+1 processes. *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10.* doi: 10.1145/1774088.1774187

Digiconomist. (2018). Bitcoin Energy Consumption Index - Digiconomist. Retrieved from https://digiconomist.net/bitcoin-energy-consumption

Gao, Y., & Nobuhara, H. (2017). A Proof of Stake Sharding Protocol for Scalable Blockchains. *Proceedings of the Asia-Pacific Advanced Network*, *Vol 44* (ISBN 978-4-9905448-7-4).

Greenfield, R. (2017). Explaining How Proof of Stake, Proof of Work, Hashing and Blockchain Work Together. Retrieved from https://medium.com/@robertgreenfieldiv/explaining-proof-of-stake-f1eae6feb26f.

Kiayias, A., Russell, A., David, B., & Oliynykov, R. (2017). Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. *Advances in Cryptology – CRYPTO 2017*, *10401*, 357-388. doi: 10.1007/978-3-319-63688-7_12.

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, *4*(3), 382-401. doi:10.1145/357172.357176.

Lee KuoChuen, D. (2015). *Handbook of digital currency* (1st ed.). Amsterdam: Elsevier.

Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., &Qijun, C. (2017). A review on consensus algorithm of blockchain. *2017 IEEE International Conference On Systems, Man, And Cybernetics (SMC)*. doi: 10.1109/smc.2017.8123011.

Nakamoto, S. (2008). A Peer-to-Peer Electronic Cash System. *Bitcoin.org*.

NEO. (2018). NEO White Paper. Retrieved from http://docs.neo.org/en-us/

Nomura Research Institute. (2015). *Survey on Blockchain Technologies and Related Services FY2015 Report*. Japan's Ministry of Economy, Trade and Industry.

Ray, J. (2018). Consortium Chain Development. *Github*.

Schumann, T. (2018). Consensus Mechanisms Explained: PoW vs. PoS. *Hackernoon*.

Witherspoon, Z. (2018). A Hitchhiker's Guide to Consensus Algorithms – Hacker Noon. Retrieved from https://hackernoon.com/a-hitchhikers-guide-to-consensus-algorithms-d81aae3eb0e3

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *2017 IEEE International Congress on Big Data (Big Data Congress)*. doi:10.1109/bigdatacongress.2017.85.