Comparative Analysis of Blockchain Consensus Algorithms

L. M. Bach, B. Mihaljević, and M. Žagar
Rochester Institute of Technology, Croatia
leo.bach@mail.rit.edu, branko.mihaljevic@croatia.rit.edu, martin.zagar@croatia.rit.edu

Abstract - **Cryptocurrencies have seen a massive surge in popularity and behind these new virtual currencies is an innovative technology called the blockchain: a distributed digital ledger in which cryptocurrency transactions are recorded after having been verified. The transactions within a ledger are verified by multiple clients or "validators," within the cryptocurrency's peer-to-peer network using one of many varied consensus algorithms for resolving the problem of reliability in a network involving multiple unreliable nodes. The most widely used consensus algorithms are the Proof of Work (PoW) algorithm and the Proof of Stake (PoS) algorithm; however, there are also other consensus algorithms which utilize alternative implementations of PoW and PoS, as well as other hybrid implementations and some altogether new consensus strategies. In this paper, we perform a comparative analysis of typical consensus algorithms and some of their contemporaries that are currently in use in modern blockchains. Our analysis focuses on the algorithmic steps taken by each consensus algorithm, the scalability of the algorithm, the method the algorithm rewards validators for their time spent verifying blocks, and the security risks present within the algorithm. Finally, we present our conclusion and some possible future trends for consensus algorithms used in blockchains.**

Keywords – blockchain; consensus algorithms; cryptocurrency; consensus problem

I. INTRODUCTION

A blockchain is an open, distributed ledger that can record transactions between two parties efficiently and in a permanent, verifiable way [1]. The most famous implementation of one being Bitcoin's, created in 2008 by a person or group working under the pseudonym "Satoshi Nakamoto" [2]. According to the original Bitcoin whitepaper, the goal of this new technology was to enable the creation of a "peer-to-peer version of electronic cash [which] would allow online payments to be sent directly from one party to another without going through a financial institution." In Bitcoin's implementation, this is achieved by timestamping every transaction within said peer-to-peer (P2P) network and hashing them into an ever-growing chain of transaction blocks. This hashing is accomplished by validators (i.e., "miners") which are peers within the network that participate in the creation of new blocks [3]. Assuming no individual or group of validators controls more than 25% of the computing power used to hash these blocks, all transactions within the chain are trusted as valid.

As there can be a potentially unlimited number of validators in any given P2P network, consensus algorithms must be utilized for there to be any cooperation between them. The most widely adopted of these is the Proof of Work (PoW) algorithm implemented by Bitcoin; however, there are numerous other means by which a network can achieve consensus, such as the algorithms that will be overviewed further in this paper.

This paper is organized as follows: firstly, problem of reaching consensus in a distributed system is explained in brief. Afterwards, the consensus systems used by the top cryptocurrencies (ranked by current market share) is overviewed. The different algorithms are compared with the Proof of Work algorithm in terms of scalability and energy efficiency. Theoretical systems that propose interesting solutions to current consensus problems are discussed and evaluated based on their feasibilities in terms of implementation. Finally, the limitations of the research conducted within this paper are discussed and avenues for further research are provided.

II. 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, first described in [4], is a problem concerning communication failure. Namely, 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.

B. Byzantine Fault Tolerance (BFT)

Byzantine Fault Tolerance (BFT) is a category of replication algorithms that aim to solve the problem of reaching consensus when nodes can generate arbitrary data. As described [5], 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

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

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

$$3f + 1 \quad (2)$$

replicas in order to provide safety and liveness in a system, where f is the total number of faulty replicas contained within said system; however, at least one known BFT implementation [6] is able to reduce this to

$$2f + 1 \quad (3)$$

required replicas.

C. Delegated Byzantine Fault Tolerance (dBFT)

As the name implies, Delegated Byzantine Fault Tolerance (dBFT) is a variant of standard BFT. Described in the NEO whitepaper [7], this fault tolerance algorithm splits clients within a P2P 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.

III. High-Profile Consensus Algorithms

As there are currently over 1,500 active cryptocurrencies (that is, actively tradeable on the global market) and it is possible for a new cryptocurrency to be created at any given moment, "high-profile" in this context is determined by a cryptocurrency's market cap. Although cryptocurrency market values are in a state of constant flux, this ranking schema was determined to be the fairest in ordering the currencies (and the algorithms behind them).

TABLE I. Top Ten Cryptocurrencies by Market Cap (in Billions) as of 2018-02-03

Currency Name	Consensus Algorithm	Market Cap
Bitcoin	Proof of Work	$ 157.3 B
Ethereum	Proof of Work[1]	$ 95.7 B
Ripple	Ripple Protocol Consensus Algorithm	$ 37.1 B
Bitcoin Cash	Proof of Work	$ 21.4 B
Cardano	Proof of Stake	$ 11.8 B
Stellar	Stellar Consensus Protocol	$ 8.3 B
NEO[2]	Delegated Byzantine Fault Tolerance	$ 8.2 B
Litecoin	Proof of Work	$ 8.1 B
EOS	Delegated Proof of Stake	$ 6.5 B
NEM	Proof of Importance	$ 5.7 B

1. Planned switch to Proof of Stake sometime in 2018
2. Formerly known as Antshares

A. Proof of Work (PoW)

According to the Bitcoin whitepaper [2], the PoW system works by scanning for a value that, when hashed, has a hash starting with a number of zero bits. This is accomplished by adding a nonce (putting in work) to the original value until the resultant hash starts with the requisite number of zero bits. Once this nonce has been found and the proof of work has been satisfied, the block cannot be changed without redoing the work for that specific block and all blocks that come after it.

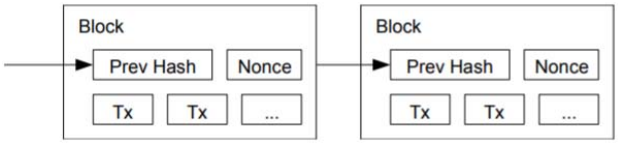

Figure 1. Visualization of two blocks within a PoW blockchain [1]

As can be seen in fig. 1, 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.

B. Ripple Protocol Consensus Algorithm (RPCA)

As the name implies, Ripple Protocol Consensus Algorithm (RPCA) [8] is a consensus algorithm used exclusively by the Ripple cryptocurrency and was developed specifically to address latency issues present within other algorithms. As defined in the whitepaper, RPCA functions as follows:

- Each server takes all valid transactions it has seen prior to a new consensus round and puts them into a public list called the "candidate set."
- Each server combines all candidate sets found on its "Unique Node List," which is a set of other Ripple servers that the server kept reference to.
- Each server votes on the veracity of each transaction in a series of one or multiple rounds.
- All transactions that meet a minimum of 80% "yes" votes in the final round are written to the public ledger and the ledger is closed.

C. Proof of Stake (PoS)

1) Original Proof of Stake

First implemented in 2012 in the form of the cryptocurrency PeerCoin [9], Proof of Stake (PoS) is of a hybrid design, where PoW is used for initial coin minting and PoS is then used for most of the network security. Within a PoS system, each coin's age is taken into consideration in the form of "coin-days." This concept is simply explained through example: holding 10 coins for 10 days equates to 100 coin-days. Upon spending these coins in a transaction, the age of the coins is consumed and reset to zero. Unlike in a PoW system, where the chain with the most work is seen as the main chain, a PoS system uses the chain with the highest consumed coin age.

Under the PoS system, a validator pays himself (thereby consuming his coin age) for the privilege of minting a new block for the network. 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:

$$proofhash < coins \times age \times target \quad (4)$$

As described in the whitepaper of another PoS cryptocurrency called BlackCoin [10], 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 akin to PoW's implementation. The difficulty, in the case of blockchains, being the difficulty of the mathematical problem that validators must solve.

2) Cardano's Ouroboros

The creators of Cardano's Ouroboros protocol [11] took the PoS algorithm described above and added additional security measures to ensure persistence and liveness within their system. 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.

D. Stellar Consensus Protocol (SCP)

Stellar Consensus Protocol (SCP), defined in [12], is a decentralized consensus protocol wherein nodes within the network do not need to trust the entirety of the network but, rather, have the ability to choose which nodes they trust. This group of nodes which trust each other is referred to as a "quorum slice," a concept first introduced by this protocol. A "quorum" is a set of nodes sufficient to reach an agreement, whereas a quorum slice is a subset of a quorum which convinces one particular node of agreement.

SCP starts with a "nomination protocol" by proposing new, candidate values for agreement. Each node which receives these values will vote for a single value among these, which eventually results in one value winning the majority vote. After the nomination protocol has been successfully executed, the "ballot protocol" is deployed. During this phase, nodes initiate voting on whether or not to commit or abort the values that were selected during the previous phase. In the event that a set of nodes cannot reach consensus, the value is moved to a higher valued ballot to be voted on again [13].

E. Delegated Proof of Stake (dPOS)

Within a Delegated Proof of Stake (dPOS) system [14], stakeholders vote to elect any number of witnesses to generate blocks. During each maintenance interval, the roster of witnesses is shuffled, and each witness is given a turn to produce a block at the fixed schedule of one block per n number of seconds (where n depends on the implementation). Witnesses are paid for each block produced; however, should a witness fail to produce a block after being elected, they may be voted out in future elections.

Specific to the EOS blockchain [15], blocks are produced every three seconds by authorized producers and, every 21 blocks the list of said producers is shuffled. If a producer has not produced any block within the last 24 hours, they are removed from consideration until they notify the blockchain of their intention to start producing blocks again.

F. Proof of Importance (PoI)

Proof of Importance (PoI) is used within the NEM network [16], which uses an underlying cryptocurrency called XEM. Each account within the NEM network has a XEM balance that is split into two parts: vested and unvested. Whenever an account receives XEM, this new XEM is added to the account's unvested balance. One tenth of every account's unvested balance is moved into the vested part every 1440 blocks. In addition, when an account sends XEM, XEM is taken from both the vested and unvested balances in order to retain the same vested to unvested ratio.

For an account to be eligible for an "Importance Calculation" it must hold at least 10,000 vested XEM. Given eligibility, Importance is calculated based on the amount of vested XEM held, the rank of the account within the network (found using the NCDawareRank algorithm), a weighting factor based on the topological location of the account (as in, whether or not the account is an outlier or part of a cluster of nodes), and two suitable constants determined by the NEM network.

IV. COMPARISONS

Table II shows a basic comparison between various algorithms as provided by [17]. Note that *energy saving* is only given a vague yes-no-partial answer, as it is impossible to provide precise numbers into how much energy each implementation uses due to confounding factors such as processor efficiency and type. Table III, shows information for algorithms not included in [17].

TABLE II. CONSENSUS ALGORITHM CHARACTERISTICS, PART I, BASED ON [17]

	Algorithm Name				
Property	***PoW***	***PoS***	***PBFT***[1]	***DPoS***	***Ripple***
Energy Saving	No	Partial	Yes	Partial	Yes
Tolerated power of adversary	< 25% computing power	<51% stake	< 33.3% replicas	< 51% validators	<20% faulty nodes

1. Practical Byzantine Fault Tolerance

TABLE III. CONSENSUS ALGORITHM CHARACTERISTICS, PART II

	Algorithm Name		
Property	***DBFT***	***SCP***	***PoI***
Energy Saving	Yes	Yes	Yes
Tolerated power of adversary	< 33.3% replicas	Variable	<50% importance

A. Security

In Tables II and III, *tolerated power of adversary* refers to how much control an attacker would need to have over the network in order to successfully attack it. For example, the PoW algorithm would hypothetically require that an attacker control at least 25% of the computing power within the system in order to forge transactions. With the PoS algorithm, an attacker would need to control at least 51% of the stake (as in, total currency) in the network to forge transactions.

The Stellar Consensus Protocol is an outlier when comparing attacker potential. Due to the use of quorum slices in SCP, a client can choose which other clients it wishes to trust. This means that, hypothetically, an attacker could control a large portion of the network but still not be able to manipulate the blockchain within small, trusted groups. Conversely, this means that an attacker could manipulate the quorum slices within the network rather than the entire quorum, itself. There is no easy solution to either of these hypothetical scenarios, as admitted by the creators of SCP in [18].

There is also, of course, the problem that not all potential attacks against each protocol have been discovered. PoW, for example, was originally thought to be safe against any attacker with less than 51% of the total computing power of the network; however, it was later discovered in [19] that an attacker could theoretically achieve the same results with only half of that number.

B. Scalability

The theoretical maximum number of transactions per second (TPS) that a cryptocurrency can achieve is listed in this section. As shown in Table IV, while the algorithm that underlies a cryptocurrency ultimately dictates the maximum TPS that can be achieved, there is still some variance between networks that use the same protocol.

The TPS numbers in table IV are primarily theoretical and often never tested. Where possible, TPS data was sourced from other scientific papers [17], whitepapers [15, 16], or other documents that originated directly from the creators of the cryptocurrency [20, 21]. However, where not possible, TPS data was sourced from community discussions between members of each cryptocurrency's respective community or third-party sites.

TABLE IV. TRANSACTIONS PER SECOND FOR SELECTED CRYPTOCURRENCIES

Cryptocurrency Name	Protocol	TPS
Bitcoin	PoW	7
Ethereum	PoW	15
Ripple	RPCA	1500
Bitcoin Cash	PoW	60
Cardano	PoS	7
Stellar	SCP	1000
NEO	DBFT	10000
Litecoin	PoW	56
EOS	DPoS	~millions
NEM	PoI	4000

Transactions within a PoW system appear to be primarily bottlenecked by block sizes, with Bitcoin, for example, suffering slow TPS numbers due to a hard-coded one-megabyte size limit on block size.

Furthermore, as not all sources in Table IV are readily verifiable (or perhaps even trustworthy, in the case of numbers sourced from marketing materials), it is difficult to provide a truly accurate comparison of TPS for each currency's protocol.

A comparison that can be made, however, is to Visa and PayPal, two well-known payment processors. Visa, as of 2018, is processing roughly 24,000 transactions per second, whereas PayPal is processing roughly 193 transactions per second [22]. These numbers bring into doubt EOS's claim that millions of transactions could be handled on a per second basis as such an increase would be multiple orders of magnitude greater than the current processing capabilities of Visa's systems.

C. Power Consumption

Most cryptocurrencies do not have an accurate estimate of how much power is being consumed on their networks, with the notable exceptions of Bitcoin and Ethereum. As such, in this section, only these two currencies will be compared. As both Bitcoin and Ethereum are currently using the PoW protocol, it can be safely assumed that the other currencies included in this study that are not using PoW will consume less power per transaction.

Fig. 2 shows the current power usage of the Bitcoin and Ethereum networks compared to selected countries. At the time of writing this paper, the entire Bitcoin network consumes slightly more power than Singapore and Portugal, and less than Uzbekistan and Romania; however, it is more than three times more than entire energy consumption in Croatia. At the same time, the entire Ethereum network consumes slightly 3.6 times less than Bitcoin network, i.e. slightly more power than Myanmar and Tajikistan, and less than Slovenia and Mozambique; and in this case it is 15% less than to Croatia.

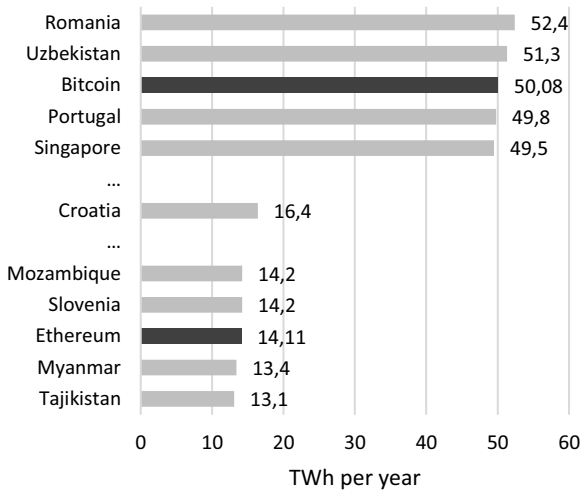

Figure 2. Comparison of energy consumption between Bitcoin and Ethereum networks and selected countries in TWh/year, based on [23] and [24], as of 2018-02-19

Fig. 3 further highlights the relatively massive amounts of power the Bitcoin network consumes. To put the above numbers into greater perspective, the US Energy Information Administration estimated that the average annual electrical consumption per US household was 10,766kWh [23]. Using a conservative estimate of 3.5 transactions per second (half of the theoretical maximum), the Bitcoin network consumes the equivalent of one US household's annual energy needs every four seconds.

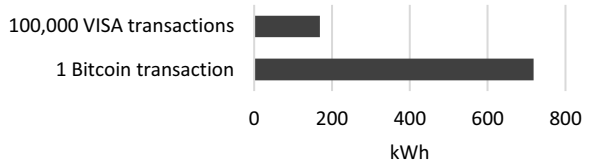

Figure 3. Power consumption (in kWh) for average Bitcoin transaction compared to 100,000 average VISA transactions, based on [23]

Comparing to Bitcoin's largest competitor Ethereum and its blockchain which currently implements the PoW algorithm, fig. 4 shows the difference in energy consumption between Bitcoin and Ethereum.

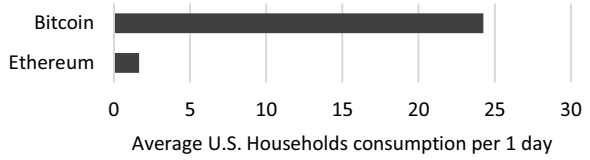

Figure 4. Bitcoin vs Ethereum power consumption per transaction in comparison with average U.S. households, based on [24]

It is obvious, despite the fact that both currencies currently use the same consensus protocol, that Ethereum is many times more efficient than Bitcoin. It is predicted, should Ethereum successfully transition to a PoS system, that the power delta between the two currencies will only grow.

V. Prospective Consensus Algorithms

The power and throughput issues present in PoW systems have not gone unnoticed. There are currently numerous alternative protocols that have not yet seen public release yet. Nevertheless, these provide interesting insights into how the blockchain development community is looking to improve the flaws in current implementations. For the sake of brevity, only two such prospective algorithms are discussed here.

A. Proof of Luck (PoL)

Currently, still a proof of concept, the Proof of Luck whitepaper [25] states that the aim of this novel system is to reduce the large amounts of computational power required by PoW and to increase transactions throughput. Within this system, each block is assigned a "luck" value as it is mined, which is a random number between zero and one: higher numbers being luckier and lower numbers being unluckier. Miners working on verifying transactions within the network will prefer to append their block to the chain with the highest luck value, which is calculated by summing all the luck values contained within each block of the chain starting from the genesis block up to the latest transaction.

Before mining is completed and the block's proof is broadcast to the network, a delay is imposed based upon the random luck value of the block being mined: a higher luck value equating to less delay time. This delay optimizes network communication within the system as, should another miner solve the proof first on a block with a higher luck value, the original miner will not need to broadcast their block to the network.

B. Proof of eXercise (PoX)

Unlike Proof of Luck, which aims to reduce the computational power required by a Proof of Work system, Proof of eXercise aims to direct the computing power towards real-world scientific problems. The PoX whitepaper [26] describes a system where miners will be given matrix-based problems provided by so-called "employers" within the system. The reasoning for using matrices is two-fold: matrices are composable, allowing for easier tuning of network difficulty, and matrices are a principle abstraction for many scientific computational problems. The whitepaper provides the following problems as examples: DNA and RNA sequencing, protein structure analysis, data mining, face detection, and more.

As the required data for these matrix problems must be stored and readily available, at cost, by an employer on the network, a "hostage credit" system is put into place. For a miner to receive a problem to solve, he must first "bid" on this problem by putting forth a deposit that will be refunded upon successfully completing the problem. Likewise, an employer must also deposit a sum that has a greater total cost than the cost of storing the matrix data. Upon completion of the problem, the solution generated by the miner is then sent to verifiers, who will use a probabilistic verification scheme to verify the data in parallel before it is committed to the blockchain.

To avoid potential collusion between miners, employers, and verifiers, solutions the matrix problems are sent through a shuffling service. This is done in two parts: directly after an employer publishes the matrix data, and directly after a miner sends the data for verification. In addition, should multiple miners win a bid to solve the same problem, the coin reward for the solution will be shared between them.

VI. Limitations

The most obvious limitations of this research are the difficulties finding accurate transaction per second numbers for each blockchain network as well as finding energy expenditure figures for the less popular blockchains (i.e. not Bitcoin or Ethereum). TPS numbers often originated from third party sites reporting on the topic or, in the case of the NEM network, numbers originated from marketing materials. These numbers cannot be fully trusted and should only be used to give a general idea of what a network could theoretically be capable of.

In addition, due to the mathematical complexity of the proofs contained within each cryptocurrency's

whitepapers, it was not possible provide an in-depth comparison of each protocol's strengths and weaknesses as the base for a blockchain network. It is due to this complexity that the choice was made to analyze cryptocurrencies' implementations of consensus algorithms as opposed to directly pitting algorithms against themselves.

Related to the above, there was the problem of choosing which cryptocurrencies to include in the comparisons. At the time of writing, the cryptocurrency market is in a state of volatility, causing the top ten list to vary throughout the research period. Furthermore, this method of ranking is the equivalent of a popularity contest rather than being based on a more objective measure.

VII. Preliminary Conclusions

Based on the preliminary findings within this paper, it can be concluded that the Proof of Work system, which is by far the most popular consensus algorithm in use among cryptocurrencies, will eventually be replaced by newer, more efficient algorithms. Ethereum is the most obvious evidence of this, as the Ethereum blockchain has been planning a transition to Proof of Stake for at least the last year.

Should Ethereum finish the transition to a PoS system, an in-depth comparison of the new system compared to the current one would provide a good avenue for further research. Should the transition not be successful, a more general analysis of RPCA and SCP could be conducted as both protocols aim at providing a global-scale network.

References

[1] M. Iansiti and K. Lakhani, "The Truth About Blockchain", Harvard Business Review, 2018. [Online]. Available: https://hbr.org/2017/01/the-truth-about-blockchain. [Accessed: 04-Feb-2018].

[2] S. Nakamoto, "Bitcoin: A Peer-to-peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed: 05-Feb-2018].

[3] G. Karame, E. Androulaki, *Bitcoin and Blockchain Security*, Norwood, MA: Artech House, 2016.

[4] L. Lamport, R. Shostak and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382-401, 1982.

[5] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," ACM Transactions on Computer Systems, vol. 20, no. 4, pp. 398-461, 2002.

[6] M. Correia, G. Veronese and L. Lung, "Asynchronous Byzantine consensus with 2f+1 processes," Proc. 2010 ACM Symposium on Applied Computing - SAC '10, 2010.

[7] NEO White Paper. (2014). Available: http://docs.neo.org/en-us/. [Accessed: 10-Feb-2018].

[8] S. David, Y. Noah, B. Arthur, *The Ripple Protocol Consensus Algorithm*, Ripple Labs Inc, 2014. [Online]. Available: https://ripple.com/files/ripple_consensus_whitepaper.pdf. [Accessed: 04-Feb-2018]

[9] S. King, S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake,"2012. [Online]. Available: https://peercoin.net/assets/paper/peercoin-paper.pdf. [Accessed: 04-Feb-2018]

[10] P. Vasin, "BlackCoin's Proof-of-Stake Protocol v2," Blackcoin.co, 2016. [Online]. Available: http://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf. [Accessed: 05- Feb- 2018].

[11] Kiayias, A. Russell, B. David and R. Oliynykov, "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol", Advances in Cryptology – CRYPTO 2017, pp. 357-388, 2017.

[12] D. Mazieres, "The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus," draft, Stellar Development Foundation, 2016. [Online]. Available: https://www.stellar.org/papers/stellar-consensus-protocol.pdf. [Accessed: 04-Feb-2018]

[13] L.S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017.

[14] D. Larimer, "DPOS Consensus Algorithm – The Missing Whitepaper," Steemit, 2018. [Online]. Available: https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper. [Accessed: 03-Feb-2018].

[15] EOS.IO Technical White Paper. Github. (2017). [Online]. Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md. [Accessed: 16-Feb-2018].

[16] NEM Technical Reference, Version 1.2. 2018. [Online]. Available: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf. [Accessed: 04-Feb-2018].

[17] Z. Theng, S.Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, , IEEE, pp. 557-564, 2017.

[18] J. Rubin and J. Holliman, "Oor: Stellar Consensus Protocol Implementation", p. 2, 2015.

[19] Eyal and E. Sirer, "Majority Is Not Enough: Bitcoin Mining Is Vulnerable", Financial Cryptography and Data Security, pp. 436-454, 2014.

[20] Stellar Basics. "Ready for Faster, Cheaper Transactions?", Stellar Development Foundation. [Online]. Available: https://www.stellar.org/how-it-works/stellar-basics/. [Accessed: 16-Feb-2018].

[21] Introducing Mijin. [Online]. Available: https://nem.io/enterprise/mijin/. [Accessed: 16-Feb-2018].

[22] Raul. "Transactions Speeds: How Do Cryptocurrencies Stack Up To Visa or Paypal?", howmuch, 2018. [Online]. Available: https://howmuch.net/articles/crypto-transaction-speeds-compared. [Accessed: 13-Feb-2018].

[23] Bitcoin Energy Consumption Index. Digiconomist. (2018). [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption. [Accessed: 17-Feb-2018].

[24] Ethereum Energy Consumption Index (beta). Digiconomist. (2018). [Online]. Available: https://digiconomist.net/ethereum-energy-consumption. [Accessed: 19-Feb-2018].

[25] M. Milutinovic, W. He, H. Wu and M. Kanwal, "Proof of Luck: an Efficient Blockchain Consensus Protocol," Proc. 1st Workshop on System Software for Trusted Execution - SysTEX '16, 2016.

[26] Shoker, "Sustainable blockchain through proof of exercise", 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), 2017.