Comparative Analysis of Blockchain Consensus Algorithms

Manjula M[1], Mallikarjun Pujeri[2], Naresh BabuR[3]
[1, 2, 3] Atria Institute of Technology

Abstract- *Cryptocurrenices 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," withinthe 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
MIPRO 2018/SP 1791

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.

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

IV. COMPARISONS

Table II shows a basic comparison between various algorithms as provided by. 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.

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

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

V. 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 blockchain (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 cryptocurrenices implementations of consensus algorithms as opposed to directly pitting algorithms against themselves.

VI. 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: 04Feb-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: