RESEARCH

A Study of Blockchain Consensus Mechanisms with Emphasis on Proof-of-Reputation

Yidi XING[1*]

, Omar HASAN[1]

, Sonia BEN MOKHTAR[1]

, Tarek AWWAD[1]

, Lionel BRUNIE[1]

and Harald KOSCH[2]

Correspondence:
idi.xing@insa-lyon.fr
LIRIS Laboratory, National
nstitute of Applied Sciences of
yon, 20 avenue Albert Einstein,
9100 Villeurbanne, FR
ull list of author information is
vailable at the end of the article

Abstract

- The emergence of blockchain technology enables people to build a distributed, decentralized and tamper-proof account book through a trust free P2P network. This technology has broad application prospects in the fields of digital assets, remittances, online payment and other financial services. Sytems based on blockchain technologies combined the application of P2P network, public key cryptography, hash pointer and cryptographic hash function to ensure the decentralization, persistence, tamper resistance, forgery resistance and auditability of the system.

- Users, as distrustful parties, can agree on the existence, value and transaction history of each other's accounts by maintaining consistency on the global blockchain network. This feature of blockchain network makes it possible to greatly save transaction costs, especially financial transaction costs, and improve transaction processing efficiency. It also allows financial services without the support of any banks or intermediaries.

- In the area of blockchains, consensus algorithms are the key elements in each blockchain P2P network, because they are responsible for maintaining the integrity and security of these distributed systems and ensuring that the system can operate on a trust-free basis. Consensus algorithms can be defined as a mechanism to achieve agreement in blockchain networks. Blockchain systems have decentralized attributes and are constructed as distributed systems. Since they do not rely on a central authority, decentralized nodes need to agree on the validity of transactions, which is the function of consensus algorithms. Consensus algorithm ensures that all nodes comply with the rules defined by the system designer and that all transactions are conducted in a reliable manner.For example, in the field of cryptocurrency, each token coin used for trading can only be spent once.

Abstract

- While trying to balance security with functionality and scalability, each consensus protocol shows its own advantages and disadvantages. In this paper, we will focus on the analysis and comparison of different types of consensus protocols. In the second section, we first present the general design model of the hierarchical block chain system we envisage. We will further reveal the importance of the consensus layer by showing its importance, utility and potential interaction with other layers. Then in sections III and IV, we analyze and compare fourteen different consensus protocols. In the fifth, sixth and seventh sections, we will focus on an innovative concept of consensus protocols: proof-of-reputation protocols(PoR). PoR introduces the concept of reputation into the consensus process. We first introduce the general design model of PoR. Then we enumerate five existing por projects, compare and analyze their ideas, advantages and disadvantages, and try to provide possible trends for the future development of proof-of-reputation protocols.

Keywords: blockchain; consensus protocol; proof-of-reputation; decentralization

Declaration

Availability of data and materials

The blockchain systems data that support the findings of this study are available from “bitcointalk.org”, “www.coingecko.com/fr/pièces/”, “www.feixiaohao.com”, “coincheckup.com”, “blocktivity.info”, “bitinfocharts.com”, “www.reedit.com/r/Vechain/comments/97zmoy”.

Also, the next reported blockchain systems data were used to support this study and are available at “Practical Byzantine fault tolerance”, “Bitcoin: A peer-to-peer electronic cash system”, “https://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf”, “DBFT: Efficient byzantine consensus with a weak coordinator and its application to consortium blockchains”, “The ripple protocol consensus algorithm”, “On security analysis of proof-of-elapsed-time (poet)”, “Slimcoin: A peer-to-peer crypto-currency with proof-of-burn”, “Proofs of space”, “Delegated proof-of-stake (dpos)", "Komodo: An Advanced Blockchain Technology, Focused on Freedom", "Komodo: An Advanced Blockchain Technology, Focused on Freedom", "Solana: A new architecture for a high performance blockchain v0.8.13", "Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain", "Algorand: Scaling byzantine agreements for cryptocurrencies", "gochain.io/assets/gochain-whitepaper-v2.1.2.pdf", "Blockchain: The State of the Art and Future Trends". These prior studies (and datasets) are cited at relevant places within the text as references [8-11, 13-23].

Competing interests statement

The authors declare that they have no competing financial interests.

Fundings

//TO DO

Authors' contributions

Y has drafted the work. Y was the major contributor in writing the manuscript and also substantively revised it. O and SB ahve made substantial contriubtions to the conception and the design of the work. O and SB have also substantively revised the manuscript. L and H have drafted the work, and have made important contributions to the conception of the work. T have made important contributions on the substantive amendments. All authors read and approved the final manuscript thus the submitted version.

I Introduction

Blockchain technology was first implemented by Nakamoto with Bitcoin applications in 2009[9]. It combines the application of encrypted hash functions, digital signature, Merkle tree, consensus protocol and peer-to-peer(P2P) network, so as to build a distributed and decentralized system based on trust-free P2P network. It could be used not only for financial trading systems[1],[2], but also Scientific research, resource management[3],[4], political domain[6],[7], etc. Using blockchain technologies, we can build a distributed database system based on distributed P2P network. The system could record a public account book, or called a "public ledger"

– this ledger sorts groups of transactions in chronological order and uses encrypted hash function such as SHA256 to encryptedly link each group of transactions. Those sets of transactions in the record are stored in a specific data structure, which we call a data block. As new transactions continue to be completed, they are packaged into data blocks, which are submitted to the end of the list of data blocks on the public ledger. That's also why we call this technology blockchain.

The information contained in the ledger shows transaction history up to the current time through block chains. These transaction records prove the existence and value of each account. Therefore, in a decentralized block chain system, every update of the ledger must be authenticated by each account holder in the network. Of course, this means that there is a need for consensus among participants. In the real world, we may not be able to find application examples with the same limitations. For example, when an entity (bank or country) decides to issue legitimate digital currency, it does not need to establish a public ledger that must be confirmed in real time by each currency holder, because the entity, as the central agency, is responsible for the verification needed to use such digital currency for transactions and ensures the security of transactions. In blockchain networks, this is not the case: nodes operate independently. In order to reach consensus, it is essential and necessary for nodes to communicate with each other through the network.

It can be imagined that in such a distributed system, there will be many kinds of errors in the process of sending messages between nodes. We can generally divide them into two types: the first is the error including node crash, data packet loss and network failure. The characteristics of these errors are that the nodes themselves are not malicious to the system. We call them "non-Byzantine errors". The second type of errors refers to the arbitrary actions of the nodes and deliberate violations of the rules of action formulated by the system designers. At this point, the wrong node may itself be malicious. The behaviors include sending messages with different contents at the same time to different nodes, delaying or rejecting messages in networks, deliberate attempts to submit illegal transaction records, and so on. Such errors are called "Byzantine errors''. In serious cases, there may be collaboration between malicious nodes, making Byzantine errors a serious problem.

Consensus protocols are thus a critical part of the blockchain system. There are a lot of practices: Bitcoin which made a great success on marketing, uses this Proof-of-Work protocol where users profit from computing proofs to randomly find the node determining the next block[9]; or PoS protocol[10], which is used by Peercoin, where users profit there locked stake within the blockchain system prove that they are trustworthy, and to compete to win the right of generating subsequent blocks; or as PBFT protocols, all nodes identity should be known under this configuration, all nodes have equivalent voting rights, and they consumes numerous rounds of communications to reach consensus[8].

The existing consensus protocols mainly face 4 serious challenges: system performance, energy efficiency, security and decentralization feature[22].

The rest of this paper is organized as follows. Section II introduces the general design model for blockchain system. Section III shows the state-of-art of fourteen different consensus protocols. Section IV summarizes the precedent one by giving tables and explanations showing the analysis results of those protocols, with a detailed explanation for these table and figures. Section V introduces the idea of proof-of-reputation, explains its idea, its operation principles, its general model, advantages and disadvantages. Section VI is an another state-of-art section where we list and present four different existing por blockchain projects. Section VII concludes.

II Background

In this section, we will introduce a general introduce the theoretical background of our research. We are going to explain a blockchain system under our envisionment, through a basic model constructed by 5 layers: a data layer, a network layer, a consensus layer, an incentive layer, a contract layer and an application layer[23].

The data layer defines the representation of data within a blockchain system. The network layer determines the data transmission method. The consensus layer focuses on reaching a consensus at the systemic level, namely a consensus of data verification. The existence of incentive scheme is to guarantee honest and legitimate behaviors of users(network nodes), since the data generation, data propagation and data verification depend on their actions and operations.

The data layer, the network layer, the consensus layer and the incentive schemes are mostly related to the implementation of consensus protocol, they construct the underlying architecture that support various contracts and general applications for a blockchain system. In general, a blockchain system consists of a data layer, a network layer, a consensus layer, an incentive layer, a contract layer, and an application layer. The data layer encapsulates the underlying data block and related data encryption and time stamping and other basic data and basic algorithms; the network layer includes a distributed networking mechanism, a data propagation mechanism, and a data verification mechanism; the consensus layer mainly encapsulates the network node. Various types of consensus algorithms; the incentive layer integrates economic factors into the blockchain technology system, mainly including the issuance mechanism and distribution mechanism of economic incentives; the contract layer mainly encapsulates various scripts, algorithms and smart contracts, and is a blockchain. The basis of the programmable features; the application layer encapsulates various application scenarios and cases of the blockchain. In this model, time-stamp-based chain block structure, distributed node consensus mechanism, consensus-based economic incentives, and flexible programmable smart contracts are the most representative innovations of blockchain technology.

II.a Blockchain data

The data layer represents the distributed ledger, which is shared by all the nodes within the decentralized blockchian system. It encapsulates the underlying data block, the related data structure and algorithms of data encryption and time stamping, etc.

Through the presence of data layer, every distributed node can use a specific hash algorithm(determined within the data layer) and the Merkle tree data structure, to encapsulate the transactional data received in a certain time period into a data block and with time stamping on it, then add it to the end of local main blockchain, thus became it the latest block on the blockchain.

In order to achieve the functions described above, the data layer mainly relies on six technologies: the data block, the hash pointers, the cryptographic hash function, the Merkle tree, the timestamps and the asymmetric cryptography.

Data block

Also called as "trasaction block" because it stores mostly transactions' information. Each data block contains a Header part and a Body part.

The block header encapsulates current block index, the address of the previous block, the hash value of current block, the Merkle-root of current block and its timestamp.

The block body contains the amount of transactions stored in current block, then the records of all validated transactions encapsulated during the generation of current block. Those transaction records together generate the Merkleroot(through the hashing process of a Merkle tree) saved in the block header.

Hash pointers

The data structure which allows the node to link the latest block to the previous one, thus constructing the chain of data blocks.

Through this technology, all history of data appeared in the blockchain system is locatable and auditable.

Sometimes, a node may have two or even several valid latest blocks that it must make choice among them to adding one of them on their local main blockchain. This is called as "fork selection" as a problem to deal with.

Timestamps

The timestamp is encapsulated in the header part of a data block, during the creation time of the block. It signifies the write-in time of the corresponding block, the purpose is to make it possible to confirm that blocks are arranged in chronological order within the blockchain.

The hash pointers and the timestamps, together they construct the Proof of existence of every data block, thus make the blockchain becoming a tamper-resistant ledger.

Cryptographic Hash function

The raw data of transactions are not recorded in the blockchain, but their hash value. The use of cryptographic hash function gives six properties to the records data:

1. As input, the raw data can be any string of any size.

2. The output is a fixed size.

3. The process to transform raw data to hash value is efficiently computable. Intuitively this means that for a given input string, we can figure out what the output of the hash function is in a reasonable amount of time. More technically, computing the hash of an n-bit string should have a running time that is O(n).

4. Collision-resistant: even if the input differs by only one byte, it will produce significantly different output values. It is infeasible to find same output value with different input.

5. Hiding: there's no feasible way to reverse the input value through the hash output.

6. Puzzle friendliness: if someone wants to target the hash function to come out to some particular output value y, and if there's part of the input that is chosen in a suitably randomized way, it's very difficult to find another value that hits exactly that target.

The use of cryptographic hash functions guarantee the "tamper-resistant", "efficiently computable during the creation" and "auditable" properties of blockchain records. The function that is most generally used is SHA256.

Merkle Tree

The Merkle tree's function is to allow to the efficient summarization and validation of the existence and integrity of the block data.

Asymmetric Cryptography

Asymmetric encryption usually uses two asymmetric ciphers in the encryption and decryption process, called public and private keys. This key pair has two characteristics:

The first is to use one of the keys (public or private). After encrypting the information, only another corresponding key can decrypt it;

Secondly, the public key can be disclosed to others, and the private key is kept secret, and other people cannot calculate the corresponding private key through the public key.

The asymmetric encryption technology is applied in the scenarios of the blockchain's information encryption, digital signature, and login authentication. The information encryption scenario is mainly performed by the sender of the information (denoted as A) using the public key of the receiver (denoted as B) to encrypt the information and then send it to B, B then decrypt the information by using its own private key.

The digital signature scenario is that sender A sent messages with his/her own private key to B, B uses the public key of A to decrypt, and to ensure that the messages are made by A.

As for the login authentication scenario, the client encrypts the login information with the private key and sends it to the server. The latter takes client's public key to decrypt and authenticate the login information.

II.b Blockchain network

The network layer encapsulates the network building mode, the messaging protocol, the data verification mechanism, etc.

Those mentioned factors of network layer should be defined corresponding to the need of real applications based on. Through the network layer, it is possible for every node within the blockchain system to participate to the maintenance(verification of data) and updating of data blocks.

This function is basic for a blockchain system since the system is distributed, we need that all the nodes could synchronize with each other on the updating of distributed ledger.

Network Building Mode

Existing blockchain systems generally take Peer-to-Peer Network(p2p network) as their networking mode, nodes within the network are the users who have the right to participate to do the data verification and ledger's updating.

Within a p2p network, all the nodes possess a equivalent class, they connect and communicate with each other based on a flat topology. There are no special centralized nodes, neither hierarchical structures. Each node will individually take on the network routing, block data verification, block data propagation and new nodes' discovering tasks.

For a blockchain network, nodes are often divided into "full nodes" and "lightweight nodes". The former stores the total records from the gensis block(first instantiated block at the creation of the blockchain system) until the latest one, participates on real-time to the data verification and ledger updating. As for the "lightweight nodes", they record only partially the blockchain, and generally request their required data from connected nodes to accomplish their operation such as data verification,

A general reason that not every user could support a full node is the high space cost of it, as for Bitcoin, a full node means a data set more than 60GB after 2016[23]; Different existing blockcahin projects offer their own strategy for their "lightweight nodes", again as for Bitcoin, they have designed a Simplified Payment Verification method to support.

For a blockchain network, the entire network data is stored on all nodes of the decentralized system. Even if some nodes fail, as long as there is still a functioning node, the blockchain main chain data can be completely recovered without affecting the recording and update for subsequent block data. This decentralization-based concept brings a better data security compare to other centralized or multi-centralized data storage mode such as Cloud.

Messaging Protocol

Since the network is distributed, once upon the generation of a data block, the generator node needs to broadcast its result to other nodes on the global netowrk in order to get their verification for this block.

For a blockchain system, its messaging protocol generally include five steps as shown below:

1. Nodes involved by transactions broadcast their transaction data to the nodes on the global network.

2. Every full node collect their received transactions then package them into a data block.

3. Through the consensus protocol adopted by current system, some of the full nodes will get the right to sign and publish their block packaged - they broadcast the block to the nodes on the global network.

4. Data verification: other nodes only validate the block when all transactions within are legitimate and not stored in the ledger yet.

5. Block acceptation: once the data verificaion has done, nodes could accept this received block and add it in the ledger(on the end of their local blockchain).

Data verification mechanism

This mechanism mainly handles two operations: verification for transaction data, and verification for data blocks.

For the transactions' data received from connected nodes, their validity would be firstly verified. If they are validate data, they will be put into a local transaction pool by chronological order, and be broadcasted at the same time to the subsequent connected nodes; if they are illegitimate transactions, these data will be rejected thus banned from the blockchain network.

The validity of transaction data concerns mostly their data structure, their grammatical normative, their data signature, etc.

As for the data blocks, their validity is also firstly verified. If they are validate, they will be locally accepted into a main chain by current node, and be broadcasted to the subsequent connected nodes; if not, they will be rejected and thus banned from the network.

The validity of data blocks concerns their hash value, their timestamp, their content transactions' validity, etc.

II.c Consensus protocol

How to achieve consensus efficiently in distributed systems is an important research issue in distributed computing field, the utility and the importance of consensus layer is to - in a decentralized system with highly decentralized decision-making power - make each node highly efficiently achieve agreement on block data validity.

Existing consensus protocols are various, some of the representative ones are PoW(Proof-of-Work) and its variants such like PoS(Proof-of-Stakes), dPoS(delegated-Proof-of-Stakes); PBFT(pratical-byzantine-fault-tolerance) and its variants such as FBA(federate-byzantine-agreement), dBFT(delegated-byzantine-fault-tolerance).

The general idea of existing consensus protocol is to - for each round of the system - as much as possible randomly elect a leader(or multiple leaders), so that all nodes could have consensus on the updated content of the ledger after locally completing data verification, and every node has equivalent opportunities to become a leader node. For that purpose, the general design of existing consensus protocols is that nodes must show a proof supported by a certain scarce resource(such as hash computing power with PoW, cryptocurrencie tokens with PoS and dPoS, nodes' votes with dBFT[11], dPOS[17] and FBA, etc) in order to win the right of ledger updating. The scarcity of such resource guarantees the fairness of this "leader election" process, and could be considered as a "security deposit" that winner nodes will honestly and legitimately operate - if they act maliciously then they will lose their invested resource.

The existing consensus mechanisms have their own advantages and disadvantages. The PoW-like consensus mechanism has formed a mature cryptocurrency-mining industry based on its first-mover advantage, for example, Bitcoin and Litecoin projects; while emerging mechanisms such as dPoS, FBA have their relative advantages on safety, environment friendly and/or efficiency. The choice of consensus protocols has become the most difficult problem to reach a consensus for blockchain system researchers.

II.c.1 Main challenges faced by the consensus protocols nowadays

Pormance bottle neck:

Taking Bitcoin and Ethereum – the most successful blockchain projects – as examples: in Bitcoin, the system could process 7 transactions per second in average, and with Ethereum, this number is currently 20, which is much lower than centralized online payment system such like Paypal and Visa, which – in practice - process separately 115 and 2000 transactions per second[9],[23].

Most of the recent consensus protocols aim on the improvement on performance with, however, a trade off between the performance and the scalability, the security and/or the decentralization.

Energy overhead issue:

As of today, 3.5 million US households could be powered with the energy used to run the Bitcoin network, while Ethereum uses the equivalent power of 1 million households. This is an unsustainable overhead. To resolve this problem, there exists 3 convenient ways which are "decreasing the exigency on local computing ability for the individual node", "reducing the complexity of data/messages transmitted on the network", "reducing the complexity of number of rounds needed to reach the consensus" - numerous recent protocols proposed different solution concepts.

Scalability problem:

As for a blockchain system, the scalability represents principally the openness, and the admissible network size of the system. It's considerable that a lot of recent protocols – in order to improve the system performance – sacrificed the scalability, making their system became closed, or the acceptable number of nodes being limited.

Security problem:

The security notion signifies principally the reliability of results of the protocol, the security of transaction operation lanced by every individual node, and the confidentiality of data for every individual node. The classical consensus algorithm of Bitcoin provides – well proved in practice – a very nice security, although for some new protocols which direct the performance and the energy efficiency improvement, a strict proof on their security is lacking, some of them even have a hard-to-solve security hole, thus can not be operated independently.

In fact, even for the Bitcoin algorithm, the recent research on “selfish mining strategy/attack” also pointed that, the Bitcoin’s security mechanism could only tolerate half of the malicious nodes compare to its intended design.

Centralization issue:

As for 2017, 80% of all blocks generated in Bitcoin network are mined by large mining companies in Iceland and in China[23], the system’s decentralization has been gradually lost. The ensuring of the system decentralization is, in general, the most different part of diverse protocols. In addition, some of recent protocols made concessions on the decentralization degree for the system’s performance and reliability.

II.d Incentive schemes

The nature of the consensus layer is to outsource the ledger updating and maintenance tasks to the glboal nodes. Every rational node is self-interested. The purpose of having incentive schemes is make the individual rational behavior that maximizes the benefits of each node being consistent with the overall goal of the security and effectiveness during the consensus process of the decentralized system.

Issuing mechanism

Currently, the issuing of incentive tokens is mostly based on the augmentation of new data blocks and new transactions, the reason of this situation is that the practical effect of incentive mechanism is to make the use of system services by nodes always profitable for the users.

Taking the Bitcoin as example, each block since the genesis block will issue 50 bitcoins to the bookholders of the block, after which the number of bitcoins issued per block will be reduced by half every 4 years (namely 210,000 blocks in average). The number of Bitcoins will stabilize at the upper limit of 21 million. The bitcoin transaction process will also incur a fee, the current default fee is one ten thousandth of a bitcoin.

Distribution mechanism

The general distribution approach of incentive tokens could be divided into two parts: one part is for the leger updater nodes, they have contribution for the maintenance and updating of the distributed ledger, so they should be rewarded because of their contribution; the another part is for the transaction proposer nodes within the system, their action animates the system, increases system network traffic and creates needs of system service.

II.e Contract layer

The contract layer encapsulates various script codes and algorithms of the blockchain system and the more complex smart contracts generated therefrom. If we take the three levels of data, network and consensus as the data modeling, data propagation and data verification functions for the base system, then the contract layer signifies the business logic and algorithm built on this blockchain virtual machine, which is the basis for flexible programming and operation of the system.

Digital cryptocurrency including Bitcoin mostly use non-turing complete simple script code to program and control the trading process, which is the prototype of smart contract. With the development of technology, other Turing-completed smart contracts can be realized to achieve more complex and flexible smart contracts such like with Ethereum. Those newly created scripting language enables blockchains to support many applications of macrofinancial and social systems.

II.f Application layer

The blockchain system has the characteristics of distributed high-redundancy storage, time-series data ,tamper-resistant and forge-resistant, decentralized credit, intelligent execution of smart contracts, security and privacy protection, which makes blockchain technology not only could be successful in the field of digital cryptocurrency, there are also a wide range of applications in economic, financial and social systems.

III Related Works – Consensus algorithms

Presentation of 16 consensus protocols

In order to let the reader get a better understanding about the evolution and the state of the art of the blockchain consensus protocols, we list and explain sixteen different protocols below. The content of the explanation includes a summary introduction, their mechanism, and an analysis about their strengths and weaknesses.

III.a Proof-of-Work(PoW)

Definition

PoW is the first consensus protocol applied to the blockchain system. As a protocol, it mainly answered to four questions below:

1. Who package transaction blocks and then update the ledger(maintain the system operation)?
2. Why users would have the motivation to take care of the update of the ledger?
3. How the rewards of maintaining the system operation are distributed?
4. How do we locally determine the main chain while forking occurs?

Consensus process

The detailed mechanism of PoW contains 4 phases:

1. In order to commit the transactions(such as, online payment, data/file transmission, etc) to the ledger, the nodes need to broadcast their own transactions in the p2p network.

2. The nodes that are willing to participate in the update of the ledger are called as "miners", they firstly verify the received transactions, then store the validate ones in local, thus form a pre-committed transactions pool.

3. For each round(in Bitcoin, 1 round is 10 minutes, and as in Ethereum, it's 15 seconds), miners need to compete, trying to – in the fastest way – resolve a

mathematical problem called “hash puzzle”. Only the miners who have found a solution are able to package their transactions in the pool into a block, and sign, publish, broadcast this block to the entire p2p network.

When a block is accepted into the main chain, then the signer could get rewards for it - it could be an amount of cryptocurrencies, or in form of other tokens.

4. The block signer needs to put their solution founded into their block’s header, “hash puzzle”’s verification is very simple, so the common nodes can easily check if this signer has the right to publish its own block.

On the other hand, because of the fact that, the earlier a miner publishes its block, the higher probability it will win for this round’s competition, whenever a node received blocks signed by the other miners, it will have the tendency to verify it, accept it then continue to find new solutions. Now it has more chance to be the winner for the next round, but not the other way around; at the same time, the miner nodes have also the tendency to accept a new block preceded by a longer chain, because that means more computing power are invested on this fork, and miners have a higher probability to gain benefits from mining on this fork.

Through the incentive mechanism which allows the mining being a profitable thing, the PoW protocol guaranteed that the selection of forks by the miners is converge. As for the common users, in order to use the various services provided by the system, they will follow the majority of the miners to choose their main chain in local. In this way, a global consensus of the network on the main chain can be achieved.

Strengths of PoW:

• Since 2009 it has been widely tested, and still generally used nowadays, its reliability and security are well known.

Weaknesses of PoW:

• The “Resolving hash puzzle” step is very consummable in term of computing resources and electricity, thus not environment friendly.

• The amount of real money invested can directly affect the nodes' computing ability: the system decentralization and security mechanism are easy to be harmed in front of the "scale economy".

III.b Proof-of-Stake(PoS)

Definition

Proof-of-Stake is a variant of PoW[10]. Its idea is to replace the notion of "work(or, computing power)" by the notion of "interests(or assets, stakes)". Stakes, or cryptocurrency tokens, are themselves a proof of scarce resources, a proof of work, thus it is not necessary to specifically invest hash computing power to make a "proof-of-work".

On the other hand, this design allows us to skip the "hash puzzle resolving" step as in PoW, that means a significant drop in energy overhead.

Consensus process

The process mechanism of PoS is basically the same as PoW, only differs at the method of block generation method:

The "resolving hash puzzle" step is canceled, instead of that, in order to update the ledger then gain the reward tokens, nodes need to firstly lock a portion of the assets held in their own accounts. These locked assets are called "stakes". At each round, the system chooses randomly a stake holder, and attribute the right of signing the next block to it.

The weight of each stake holder is directly associated with their amount of stakes held, for example, if a node possesses 10% of equity(cryptocurrency) in the system, then the probability that it wins is 10%.

Strengths of PoS:

• Attacking a PoS system is very harmful for the attackers, because they are themselves stake holders of the system.

• PoS is resistant to the "scale economy": in PoW, for ten thousands miners that each pays one euro electricity fee per minute, they hold actually a pretty low computing power, although for one miner who pays ten thousands euros electricity fee per minute, it gets a very high computing power. While in PoS, we can guarantee that the interest brought by one euro is constant.

Weaknesses of PoS:

- "Nothing-at-the-stake attack": seeing the fact that mining is barely free for every participant in a PoS system, the rational users will have the tendency to generate blocks on as many as possible forks, in order to gain a maximal benefit. But this behavior can lead to a system inflation, then a serious depreciation of system assets.

III.c delayed-Proof-of-Work(dPoW)

Definition

The idea of dPoW is – based on an existing blockchain which uses PoW or PoS protocol – constructing a new blockchain system[18]. Its mechanism relies on a serie of notarized nodes selected by prior voting. These nodes import the dPoW blockchain into an existent blockchain such as Bitcoin, making the consensus protocol be benefited from the security of the existing powerful blockchain.

Consensus process

Here we take the Komodo as example - the first cryptocurrency where the dPoW is implemented:

By select a group of nodes called "notaries" in the network of the original system, the new one transmits firstly all its pre-committed transactions to these notaries; the selected nodes submit those transactions to the safe and existing PoW blockchain, then return the results of transactions processing back to the new system - here comes the notion "delay" in the title of this protocol.

Strengths of dPoW:

- The dPoW system does not have any necessity on hash computing power, thus is it environment friendly.
- Even without the "hash puzzle resolving" step, the system could also have a good security guaranteed.
- dPoW could give additional value to other system, without need of directly offering cryptocurrencies, neither making any tradings among them

Weaknesses of dPoW:

• The system must rely on a PoW/PoS system.

• With the existing of notaries, the original system must arrange different hash rates for common nodes and notaries nodes, otherwise, the relied system could not actually operate, or the original system's security will be weakened.

III.d PoET(Proof-of-Elapsed-Time)

Definition

The PoET protocol was introduced by Intel research team[14], it's also a variant of PoW. Its idea is to replace the notion of "work(or computing power)" by the notion of "time cost".

Consensus process

The process of PoET is also basically the same to PoW, only differs at the block generation method: in PoET, in order to generate new blocks and get rewards, nodes need to firstly sleep for a randomly generate length of time. Once it's awaken, it could send the awaken time to a pre-committed block for current round. Among all the nodes competing for a same block, the first of them to wake up wins.

Strengths of PoET:

• The PoET system gives an equal chance of winning to a large number of network participants, low resource users are also worthy to join the competition.

• For all the participants, it's very easy to verify that the block generator was delegated in a legal way.

• The cost that every node needs to pay for being delegated, is proportional to the benefit obtained from it.

Weaknesses of PoET:

• Hardware dependencies & Single point of failure: The PoET mechanism has 2 critical exigencies: the waiting(sleeping) time of each node is randomly choosed, and the winner participant has really accomplished the wating. This internal mechanism demands that this part of trusted codes need to be operated in a trusted environment, as for PoET, it relies on some specific Intel hardwares. It also could cause a single point of failure issue, whenever someone hack the Intel hardware, the corresponding node could generate as much blocks as it wants.

III.e dPoS(delegated-Proof-of-Stake)

Definition

dPoS is a variant of the PoS protocol. With dPoS, it's still important for the nodes to hold an amount of equity within the system, but they no more need to partially block their assets as tokens, and they do not compete to gain a "stake holder" identity[17]: different from PoS, the nodes do not compete to win the right of block generation, their right is to elect leaders(called as "witness"). The witnesses form a committee, then take charge of the generation of blocks in a cooperative way. In dPoS, the system actually centralized the block generation step.

Consensus process

Here's a concrete process of dPoS protocol:

1. During each period of "ledger maintaining", nodes could vote for other nodes as "witnesses of current period". Most of the dPoS systems use "affirmative votes" mechanism, which means they could only vote in favor, thus the nodes who get the highest accumulated weight can be elected: the weight of votes of every node depends directly on their holding stakes, more specifically, it depends on the proportion of their holding stakes to the total stake of the system.
2. Once the election completed - some of the dPoS systems will also elect a list of alternative witnesses, who will replace some of the actual witnesses if they acted maliciously or if they couldnt't work normally - a committee of witnesses is actually established, the witnesses collect the pre-submitted transactions, then package them into transaction blocks by a polling manner.

Without changing the solutions proposed in PoW of "why the nodes have the motivation to maintain the ledger" and "the distribution of incentive tokens", the dPoS made innovations on the solutions of "the generation of new blocks" and "the selection of blockchain forks": the former is taken over by a delegated committee, the latter's answer is that every on duty witness signs and publishes deterministicly their block.

Strengths of dPoS:

• High energy efficiency compare to PoW and PoS. The existing of the elected committe reduces the complexity of messages and rounds needed to reach the consensus, the skip of "hash puzzle" step saves also a lot of computing power.

• High performance. The reduced messages and rounds complexity also improve the protocol performance.

Weaknesses of dPoS:

• The centralization in "blocks generation" step make the system being possibly controlled by a grouop of high equity nodes.

• As a supplement to the above point: in order to get the incentive tokens, high stake holder nodes will always have a tendency to vote for themselves - and they have high voting weight by themselves - which make the elect process also becoming centralized.

III.f Algorand

Definition

The algorand protocol was proposed by MIT's research team in 2017[21]. It's a protocol based on PoS, PBFT[8] and elect mechanism, the research team focused on the "random leader election problem", or in other words, "the distribution of the right of blocks generation". For that purpose, the Algorand protocol mainly answered to 3 questions: "how to build a randomness generator", "how to guarantee that elected leaders could prove themselves whtiout revealing their identity(avoiding leader-targeted attack)", and finally, "how to deal with off-line nodes(appeared in the election process)".

Consensus process

The concrete process of Algorand consists of 2 basic phases:

1. Proposer election. The proposers have the right to generate blocks in the current period. The election process is an imitation to PoS, the weight of being selected of a node depends on its holding equity.
2. Using BA*(Byzantine Agreement*) algorithm to reach the consensus.

The Algorand protocol uses a cryptographic sortition algorithm, such that every proposer learns in a secret situation that is was selected.

Each proposer firstly broadccasts the highest priority block that it considers, afterward broadcasts its known highest priority block, these 2 steps are achieving by using PBFT process.

The consensus is firstly made among the proposers, thus would be inserted in local for all other normal nodes.

Strengths of Algorand:

• It combines the using of PBFT algorithm and the idea of public blockchain: the Algorand system is freely for nodes to join or leave, and benefits from the fault tolerance featue of PBFT consensus protocol.

Weaknesses of Algorand:

• Despite its complex process, there is no direct results showing that Algorand has a better performance than other election mechanism based protocol such as dPoS.

III.g PoC(Proof-of-Space)

PoSpace, also called as PoC(Proof-of-capacity), is a variant of PoW protocol, instead of hash computing power, the tokens that nodes need to invest into the competition is a certain amount of memory or disk space[16].

The concrete process of PoC is very similar to the PoW, only using a different and special hash function called MHF(Memory Hard Function): the function feature is, its computing cost depends on the memory size that this function can call.

The “hash puzzle” step in PoC could prove that the node - which have found a solution - saved or say “invested” enough memory space for the competition. The verification step shoudl stay efficient, one possible solution is by asking the competitors to generate Pebbling figures, and verifiers just simply needs to check several random spaces in the figure.

Advantages of PoC:

• It is more environment friendly compare to PoW, because the storage space is a more generic resource than the hash computing power, and occupy also lesser energy.

Defects of PoC:

• The capacity based competition could lead to an another centralization situation.

• The fact that hard disk space become valuable could encourage hackers to develop malicious software, and attack people's hard disk.

III.h PoBurn

The PoBurn protocol is a variant of PoW[15], instead of investing on hash computing power, the miners need to send their cryptocurrencies(tokens) to a unretrievable address and thus "burn" their tokens, in order to win the right of mining new blocks.

Basically the same as PoW, the only change that PoBurn has made in its consensus process is that the protocol will randomly generate some addresses which do not have a private key, thus the coins stored in there could not be spent, and the protocol also creates a book to track these coins.

Advantages of PoBurn:

• Users who tend to hold cryptocurrencies for long-term gains would have more chance to be benefited from a such system.

Defects of PoBurn:

• Still wasting resources insignificantly.

• Nodes that don't care the waste of their coins would have more possibility to generate blocks, which means, the high resource nodes could still control the system service, just like in PoW now.

• The fact that "coins have been burnt" is not easy to be verified, this could either cause security issue, either lead to delay in transaction processing.

III.i PoA(Proof-of-Authority)

PoA protocol runs based ona pre-determined committee of nodes called signers[20]; the signers take charge of blocks generation; signers could vote for invite new members; signers work in a polling manner, and each signer must wait for a fixed period to have the chance to generate a block again.

Here's the concrete process of PoA Protocol:

1. A list of initiate signers are determined in the genesis block.

2. The signers take charge of the blocks generation in a polling manner, which means, the "IN-TURN" signer could publishe its block with a higher priority, and the other "OFF-TURN" could also propose their own block - but with an inferior priority - in order to deal with the situtation that the "IN-TURN" one was offline.

3. The signers could potentially make a proposal of "invite new signer join in the list" or "exile an original signer" by broadcast it as a transaction.

Advantages of PoA:

- The consensus has high energy efficiency compare to PoW.
- The consensus has high performance.

Defects of PoA:

- The system is actually centralized, or more specifically, "multi-center", thus more adoptable for a system where all the nodes identity are verified before joining.

III.j PoHistory

PoH protocol aims on making transactions processing independent from the consensus process. This protocol is a variant based PoS algorithm[19].

With PoH, we form a "hash chain" by continuously running the hash function. This chain includes the number of times the function runs, the function state, the output value, and the block index. Each record on this hash chain is stored inside a transaction block, which is equivalent to, coding a trusted clock into the blockchain—the research team's assumption here is that the timestamps of transactions received by the system are not necessarily trusted.

The significance of PoH is that the nodes do not need to witness, neither to communicate with each other, every node can verify locally the time and sequence of event occurrences. Thus the PoH system does not demand to all the nodes to achieve a consensus, but only asks everyone to agree that event A occurred before event B.

The hash chain generated by PoH is a part of blockchain, as for the generation of blocks, the PoH protocol relies on PoS algorithm.

Advantages of PoH:

• High Performance, especially high throughput, because of reduction on message exchanging complexity.

• The consensus has high performance.

Defects of PoH:

• The PoH project in the real world is still in early days, lack of information.

• Experiments about the system's reliability are not begun yet.

III.k BFT(Byzantine Fault Tolerance)

The BFT is the description of the reliability of a fault-tolerant computer system facing Byzantine failures: the Byzantine failure is a crash(or fail-stop) where the failure nodes could have any arbitrary behaviors. While happening Byzantine failures, if the node behaviors include malicious responses and information forged, we call this situation as "Byzantine faults", and these nodes as "Byzantine nodes".

III.l PBFT（Pratical Byzantine Fault Tolerance）

PBFT is a state machine replication algorithm[8]. The service is modeled as the state machines, the state is replicated in different nodes of the distributed system. PBFT is adopted for closed system and demands communications among every pair of 2 nodes.

The concrete consensus process of PBFT is:

1. The client send requests to primary nodes.
2. The primary nodes broadcast the received requests to backup nodes.
3. The backup nodes verify the primary identity.
4. The backup nodes commit the received transaction/request.
5. The backup nodes reply to the primary one.

Advantages of PBFT:

• High Performance: high throughput and high bandwidth.

• High Security: It has a relative security since all members joining the network are being validated. However, this situation could be considered as "insecure" for small users who don't belong to any of those center organizations.

Defects of PBFT:

- Only adopted for closed and non-large scale system.
- The system is centralized, or at least "multi-center".

III.m dBFT(delegated Byzantine Fault Tolerance)

With dBFT protocol, the global nodes select some agents nodes by voting; then those agents run the PBFT algorithm[8] between them to decisively complete the block generation mission. Voting in the network is real-time and asynchronous[11].

Advantages of dBFT:

- High Performance.
- High scalability for large scale system.

Defects of dBFT:

- The system is centralized, or at least "multi-center".

III.n FBA(Federated Byzantine Agreement)

The main difference between FBA and PBFT is that, the nodes no more need to get consensus with other nodes on the entire network, but with "a certain quorum of nodes", or with a "subnet representing a sufficient number of nodes".

As for the concrete process, FBA works basically the same as PBFT, the only difference is that the system could have - at the same moment - a list of primary nodes, each primary node takes care of its own main chain, then in chronical order make consensus among them to get an agreement of the global view.

Advantages of FBA:

- Tremendeous throughput.
- Low transaction processing delay.
- Good system scalability.

Defects of FBA:

- It relies on the trustworthy of the subnetwork chosen by each node.

III.o Ripple consensus

Ripple protocol is a variant of FBA protocol. It's nowadays an opensource online payment protocol[13].

In Ripple's network, the transactions are initiated by the clients (applications). Then the transactions are broadcasted to the entire network via the tracking nodes or the validating node.

Ripple's consensus is achieved between the validating nodes. Each validating node is pre-configured with a list of trusted nodes called UNL (Unique Node List). The nodes on the list should vote on the transaction deal. Once the approved votes reach a threshold, the current validating node will send these deals to other validating nodes: this transmission will continue, until the transaction reaches the fourth time the threshold - which is, 80% of approved vote. Afterward this deal/transaction could be recorded in the ledger.

Advantages of Ripple:

• High performance, low transaction processing delay.

• High Security: It has a relative security since all members joining the network are being validated. However, this situation could be considered as "insecure" for small users who don't belong to any of those center organizations.

Defects of Ripple:

• The fault tolerance percentage is only 20% for Ripple system.

III.p Stellar consensus

The Stellar is also a variant of FBA protocol[12]. Unlike in Ripple, the Stellar system does not pre-set trusted nodes, or in other words, there is no UNL for the validating nodes[13]. In Stellar, the nodes themselves decide the subnet they trust.

Advantages of Stellar:

• High performance and good scalability.

Defects of Stellar:

• Configure a list of trustble nodes is costly for every user; and a bad configuration could cause forks or other Byzantine faults.

IV Analysis

Consensus algorithms comparison

Various consensus algorithms have different strengths and drawbacks.Table I to Table IV bring an assessment around various consensus algorithms, and we use the properties considering following[24],[26],[27],[28],[29],[30].

Protocols/Example	Blockchain Type /Node Identity	Perfomance	Energy Efficiency
PoW/Ethereum	public (public blockchain protocols are also suitable for consortium and private blockchain systems)/public	15tps(transactions per second)	no
PoS/Peercoin	public/public	97tps	partial - Hash computing(mining process) still exists
dPoW/Komodo	public/public	100tps, potential 45.000 tps	partial - Hash computing(mining process) still exists
dPoS/ Bitshares	public/public	100.000tps claimed, daily proven 3400tps	partial - Hash computing(mining process) still exists
Algorand / Algorand	public/public	>1000tps claimed	partial
PoC/Burstcoin	public/public	80tps	partial-using hardware memory instead of hash computing power, however the energy-consuming mining process still exists

Table I-1. Comparison of consensus protocols for blockchain type, performance and energy saving level.

1) Blockchain type and Node identity: it's useful to understand if a protocol could serve for a public system, or only for a closed system. Nowadays, the blockchain systems generally include 3 concepts in terms of type division—

a) the public chain, in which all member nodes can freely join and leave; in Ethereum, Bitcoin, Peercoin, Bitshares, their purpose for a decentralized network made them choosing public chain.

b) the private chain, completely private, with strong third party providing node identity assurance and controlling node permissions distribution; these systems are often controlled by a single organization or company.

c) the consortium chain, "partially guaranteed decentralization" – also called as "semi-private chain". It is generally operated by specific organization groups that opens the inscription access to qualified users and ensures that the identity of the nodes is audited and documented. In practice, many financial and commercial institutions are building their own "circle of friends" based on block chain technology with consortium chain, especially like Lawtooth Lake Hyperledger, Hyperledger Fabric, etc.

Protocols/Example	Blockchain Type /Node Identity	Perfomance	Energy Efficiency
PoA/Vechain	consortium (consortium blockchain protocols are also suitable for private blockchain)/permissioned	10,000tps claimed, 500tps proven in history[25]	yes
PoET / Sawtooth Lake	consortium/public	1300tps claimed	yes - timer certificate instead of consumption of electricity
PoHistory/ Solana	public/public	50.000tps claimed	yes
PoBurn/ Slimcoin	public/public	up to 1000tps claimed	partial - Hash computing(mining process) still exists
PBFT/Hyperledger	consortium/permissioned	1000tps	yes - pbft process excluded hashing procedure. So do the following four pbft-like algorithms
dBFT/Neo	public/public	1000tps, potential 100.000 tps	yes
FBA/Bravo (BVO)	public/public	1500tps claimed	yes
Ripple/Ripple	consortium/public	1500tps	yes
Stellar/Stellar	public/public	1000tps	yes

Table I-2. Comparison of consensus protocols for blockchain type, performance and energy saving level.

2) Performance: Blockchain performance is generally measured by transactino processing delay and network throughput. These two factors could be indicated by “transactions (processed) by second”.

We could see that dpos and Ripple have most extraordinary performance. We could also notice that it’s hard to prove the maximum performance claimed by a lot of protocols.

3) Energy Saving: As for PoW and some of its variants such like PoBurn[15], PoHistory, the demand on hash computing power make the system environment unfriendly; as for PoS and its variants such like dPoS, dPoW, the competition of hash computing power is removed, but the mining process is stille kept[10],[17],[18]; Regarding PBFT, FBA series protocols, there is no more concept of mining, the block generation phase is somehow centralized and thus saved power tremendously.

Protocols/Example	Adversary Tolerance Ability	Scalability(Openess and Expandability)	Decentralization
PoW/Ethereum	<25% computing power	Open Lack of expandability due to low performance	Relative centralization: decentralization gradually lost with pow
PoS/Peercoin	<51% stake	Open and Expandable	Relative centralization: first mover advantage with pos
dPoW/Komodo	<25% computing power	Open Lack of expandability due to dependence on pow protocols	Relative centralization: dependency on pow and pos protocols
dPoS/ Bitshares	<51% validators	Open and Expandable	Relative centralization: voting results can be highly involved by top users
Algorand / Algorand	<33.3% byzantine voting power	Open and Expandable	Decentralization guaranteed
PoC/Burstcoin	<25% computing power	Open and Expandable	Decentralization guaranteed
PoA/Vechain	<51% validators	Open and Expandable	Relative centralization: authority validators mechanism is too centralized

Table II-1. Comparison of consensus protocols for attacker tolerance, scalability and decentralization level.

4) Adversary tolerance ability: Considering the recent research on "selfish mining strategy", once the controlled hash computing power of one miner party exceed 25%, the PoW security guarantee ,thus influence dPoW[18]; the PoS security threshold is commonly known as 50%, same limitation for the variants of PoS; PBFT and FBA

series algorithms are manufactured to manage up to 33.34 defective nodes; as for Ripple, it has a more restrict reliability setting[13], which makes it only maintaining correctness when the proportion of faulty nodes in a unique node list are lower than 20%.

Protocols/Example	Adversary Tolerance Ability	Scalability(Openess and Expandability)	Decentralization
PoET / Sawtooth Lake	potential single point failure risk - highly dependent on Intel hardware enclave technologies	Restricted open(dependency on Intel hardware with SGX) and Expandable	Decentralization guaranteed
PoHistory/Solana	Unknown	Open and Unknown expandability	Unknown
PoBurn/ Slimcoin	<25% computing power	Open and Lack of expandability due to mining process and "coins burning process"	Relative centralization
PBFT/Hyperledger Fabric	<33.3% byzantine faulty replicas	Closed	Relative centralization
dBFT/Neo	<51% validators	Open and Expandable	Decentralization guaranteed
FBA/Bravo (BVO)	Unknown	Open and Expandable	Unknown
Ripple/Ripple	<20% UNL faulty nodes	Closed but expandable	Relative centralization: The company holds a large amount of money and controls many validation servers.
Stellar/Stellar	Unable to conclude(because of the Quorum algorithm and "qurom intersection property")	Open and Expandable	the top 100 accounts hold 95% of the total supply

Table II-2. Comparison of consensus protocols for attacker tolerance, scalability and decentralization level.

5) Scalability: This factor involves two factors: the openess, whether nodes could freely join and leave the system; and the expandability, when tens of thousands, hundreds of thousands of users are online, whether the system could support with its performance.

Consortium chains are generally closed system; however, PoET(Sawtooth Lake) and Ripple are expandable because of its nice performance, where Fabric and Ripple is not. PBFT is not scalable with large scale network.

6) Decentralization: PoW will gradually losting its decentralization because of the fact that hash computing power could easily be centralized, so do dPoW, PoB, etc. As for PoS, "The poorer the poor, the richer the rich" is predictable, because the protocol supports "First Mover advantage", so does dPoS. Consortium chains generally operate under a "multi-center mechanism": they are also relatively centralized.

Protocols/Example	Consensus process	Block generation method	Reward token distribution method
PoW/Ethereum	probabilistic(numerous forks could exist at the same time within the network)	Competitive - a. All nodes have the right to generate blocks b. Nodes compete to win the insertion on the blockchain	Coins - Emitted in proportion to amount of network activity
PoS/Peercoin	probabilistic	Competitive	Coins - Emitted in proportion to amount of network activity
dPoW/Komodo	probabilistic	Competitive	Coins - Emitted in proportion to amount of network activity
dPoS/ Bitshares	deterministic(Only one or a very few forks could exist at the same time within the network)	Cooperative - a. Only a selected nodes have blocks generation right b. Selected nodes principally take turns in blocks generation	Coins - Emitted in proportion to amount of network activity
Algorand / Algorand	deterministic	Cooperative	No new tokens created
PoC/Burstcoin	probabilistic	Open and Expandable	No new tokens created
PoA/Vechain	deterministic	Cooperative	No new tokens created

Table III-1. Comparison of consensus process, block generation method and reward token distribution method.

7) Consensus process: This column describes in which way corresponding protocol reaches the global consensus view. With deterministic process, normal nodes almost don't need to update local chain because of fork problem. As for probabilistic process, forking occurs quite frequently. Naturally, deterministic process could save a lot of communication messages and communications rounds.

However, to make a reliable deterministic consensus protocol, the messages for communicating before the block generation are often heavy. So there's this trade-off.

8) Block generation type: The way of block generation is one of the most fundamental difference about how different protocols reach consensus. As for competitive consensus: a dencentralized competition exists for the generation of block of every round, it protects the fairness for all the system users(nodes), but also costly in terms of time and energy; a cooperative consensus generally centralizes the block generation phase, in order to have a better performance and energy efficiency.

Protocols/Example	Consensus process	Block generation method	Reward token distribution method
PoET / Sawtooth Lake	probabilistic	Competitive	No new tokens created
PoHistory / Solana	probabilistic	Competitive	Unknown
PoBurn / Slimcoin	probabilistic	Competitive	Unknown
PBFT/Hyperledger Fabric	deterministic	Cooperative	No new tokens created
dBFT/Neo	deterministic	Cooperative	No new tokens created
FBA/Bravo (BVO)	probabilistic	Cooperative	No new tokens created
Ripple/Ripple	probabilistic	Cooperative	No new tokens created
Stellar/Stellar	probabilistic	Cooperative	No new tokens created

Table III-2. Comparison of consensus process, block generation method and reward token distribution method.

9) Reward token distribution method: there are two series of protocols in general: in pow-like protocols such as pos, dpos, we distribute incentive tokens(such as cryptocurrencies) to block generator nodes[10],[17]. This method serves mostly for public systems.

In PBFT-like protocols such as Algorand[21], Ripple[13], dBFT, we do not give incentive tokens to encourage block generators, but to network managers. Which means, by cancelling block reward, these protocols keep the transactions fees as the reward of collecting and validating transactions. This method serves mostly for consortium blockchains, as for these systems, in most of the time only a selected nodes have the right to generate block. But these super nodes are still worthy being rewarded beacause of maintain the network.

Protocols/Example	Algorithm used within consensus (incentive) protocol	Language	Github release version & last commit
PoW/Ethereum	Ethash	Golang, C++, Solidity, Serpent, LLL	v1.9.3 (2019-09-03); 2019-09-03
PoS/Peercoin	SHA-256	Michaleson	v0.8.3ppc (2019-08-27); 2019-07-30
dPoW / Komodo	Equihash	C++, Golang, Python	2019-8-30
dPoS/ Bitshares	DPoS	Python, C++	BitShares Core 3.3.0; 2019-09-02
Algorand / Algorand	Algorand(VRF & BA*)	Golang, Java, Python, Javascript	Unknown
PoC / Burstcoin	Shabal256	Golang, C++, Solidity, Serpent, LLL	Burstcoin Reference Software 2.4.2; 2019-09-04
PoA/Vechain	SHA-256	Golang, Java	v1.1.4; 2019-09-04

Table IV-1. Comparison of mathematical algorithms, coding language and last version&commit.

10)Algorithm used within consensus protocol: these are the encryption algorithms, or some more complicated and original algorithms, operating within the protocol on mathematical layer.

11)Language: The coding language for these fourteen representative projects. We could notice that C++, Python and Golang are the most usefule and also most used languages to developing blockchain projects.

12)Github release version & last commit: This columns records the version of the data of each project that we've listed here.

Protocols/Example	Algorithm used within consensus (incentive) protocol	Language	Github release version & last commit
PoET / Sawtooth Lake	cannot summarize	Python	v1.2.2; 2019-9-04
PoHistory / Solana	Unknown	Rust, C++	Mavericks v0.18.0; 2019-9-04
PoBurn/ Slimcoin	Dcrypt	Python, C++, Shell	Slimcoin 0.6; 2019-5-26
PBFT/Hyperledger	cannot summarize	Golang, Java	v1.4.3; 2019-08-30
dBFT/Neo	SHA-256	C#	v2.10.3; 2019-9-02
FBA/Bravo (BVO)	Unknown	Javascrpit, C++	Bravo 0.23.0 Release; 2019-5-28
Ripple/Ripple	Opencoin	Java, Go, C++	rippled Version 1.3.1; 2019-8-23
Stellar/Stellar	Opencoin	Java, Go, C++	v11.4.0; 2019-9-04

Table IV-2. Comparison of mathematical algorithms, coding language and last version&commit.

V Proof-of-Reputation

V.1 Design Overview

The PoR is a new concept about consensus protocol in p2p network environment for blockchain system. Its core idea is to introduce the notion of reputation of each node - which represents their individual trustworthiness within the system - into the consensus process. By considering the reputation as an overall state of node after multiple transactions, the system will assign a different weight to every node in consensus process depending on their own "reputation value".

The weight represents the capacity that nodes could influence the consensus decision making process, especially 1) the leader election process. At each round, we determine the nodes that have right to update the ledger by generating new blocks;

2) the block acceptation phase. At each round, nodes need to get synchronization about their choice on local main chain if they have multiple forks as choices.

V.2 Principles

A consensus protocol generally deals with 3 problems: 1) the block acceptance, namely the fork selection problem; 2) the block generation, namely a random leader election problem; 3) the problem of the issue and distribution of incentive tokens. Facing these issues, the PoR brings improvements based on exsiting consensus protocols such as PoW, PoS, PBFT, dBFT, etc.

Fork selection

While nodes received multiple new blocks propagated from block generator nodes, they need to choose one of them to add to the end of their ledger in local, or even modify some previous blocks. This is what we call the "fork selection" problem.

As the lastest consensus protocol, the PoR could treat this problem with two different design models: the first, is to imitate PoW-like protocols, that nodes accept the longest chain(or the "most weighted" chain) and every block generator could propagate their prepared block of current round. In the global view, the convergence of fork selection of all nodes is probabilistic; the second way is, all nodes know that there is one and only one block generated and propagated for current round, so that the convergence of fork selection is deterministic: no more forking problem if all nodes act honestly.

The influence of the choice among these two methods on system security and performance depends on the concrete implementation, in the exisiting PoR projects, both options have been selected.

Block generation

Within a blockchain system, we update the ledger through generate new data blocks, so it's critical that all nodes should have agreement about the identity of block generator nodes for each round.

The Proof-of-Reputation protocols could also treat this problem with two different design models: the first, is to imitate PoW-like and PoS-like protocols, that every node could compete for the right of generate current round's block by investing a certain scarce token(such as hash computing power for PoW, cryptocurrency shares in PoS), the block generation is competitive seen that the generation and propagation is a competition under this mechanism; the second way is that the system builds a committee among all nodes for each round's block generation, the member nodes of the committee takes charge of block generation in a polling manner generally. The block generation is then cooperative seen that we centralize the block generation right to a limited group of qualified nodes, the generation and propagation of new blocks don't process in the form of a competition, but the members of the committee take turns in charge of cooperation.

The influence of the choice among these two methods on system security and performance depends on the concrete implementation, in the exisiting PoR projects, both options have been selected.

Incentive tokens' issue and distriubtion

The incentive schemes is a strategy largely accepted by existing consensus protocols, of which the purpose is to make the nodes' self-interested behavior consistent with the maintenance of the system. All rational nodes would act honestly and legitimately while participating to the update and the maintenance of the ledger, because they could get reward for it from the system.

With PoR, a common choice as reward token is nodes' reputation value. And, like in almost all other kinds of protocols, the issue and distribution of reward tokens of PoR are through new block generation("block reward") and new transaction completion("transaction fee").

V.3 Advantages Analysis

As mentioned above, while operating a consensus protocol, it's necessary that the participant nodes could prove for themselves that they will obey the protocol rules, be reliable(no malicious acts).

A common practice for consensus protocols is that, the participant nodes need to invest in some certains scarce resources as a "security deposit ": in PoW, we take the hash computing power invested as the "deposit", in PoS, the stakes held by the nodes become an alternative solution. While in PoR, we talk about the reputation of a node.

This design model can bring advantages to a blockchain system on numerous aspects: the performance, the energy efficiency, the decentralization level, the fairness and the security.

Energy Efficiency

Since the “security deposit” used in PoR is - instead of the hash computing power - the nodes reputation, PoR could save a lot of electricity power and computers computing power compare to the PoW-like protocols(PoW in Bitcoin, PoW in Ethereum, dPoW, etc), thus the PoR is more environment-friendly.

Performance

The PoR protocol can improve the efficiency of consensus achievement in 2 ways:

Firstly, using the hash computing power as “security deposit” is not only costly in terms of energy consumption, but also in terms of time overhead. PoR brings improvements on the system performance by skipping the “hash puzzle resolving” step just like in PoS(using stakes as tokens for security deposits[10]), or in PoBurn(using “burned” cryptocurrency as tokens for deposits), etc.

Secondly, the nodes reputations are quantified and could be consulted within the system - which is not the case in Pow, the system couldn’t offer any information about the hash computing power held by any nodes. This advantage allows the “temporal centrazlition during block generation phase” being realizable, which means during the step of generation of subsequent blocks, the system can - based on the ranking of nodes reputation - to distribute at each time the participation rights to a limited number of nodes. This brings advantages in terms of the complexity of number of messages transmitted, and the complexity of number of rounds needed to achieve consensus during block generation step, just like in dPoS(using the ranking of stakes to form the temporal centralized committee) and in dBFT(using the ranking of votes from all the nodes[11]).

Fairness

In the case when we define the reputation as an non-consumable and non-transportable attribute, the Proof-of-reputation could offer a better environement in terms of fairness:

Node's reputation should only be accumulated through every completed transactions of it, thus its reputation takes time to augment, it makes reputation being equivalent to the time and activity that nodes have contributed or invested into the system; time and activities are the fairest investment, because users with high or low resources(in terms of assets, etc) in the real world are all equivalent in term of their input capacity on time and activities. There could a difference in the size of the business for high and low resource nodes, although as long as the influence of the size of the transaction is controlled about the change in reputation value by protocol design, the fairness of the reputation model for all nodes can be guaranteed.

Reputation is non-consumable, non-transportable, individual for each node, only could be accumulated trhough node's invested time and completed transactions, these facts make the reputation not only an attribute bound to the node itself, but also a resource that can not be obtained by or converted from any type of out-of-system resources. Rich nodes aren't able to get reputation easier than the poor ones, and node groups controlling reputation resources are difficult to formed because they cannot share their own reputation with other one, neither provide (other) resources to help allies gain reputation.

It can be seen that the design of PoR not only guarantees the fairness of the reputation model, but also ensures sufficient robust decentralization of the system based on this "fairness" feature.

Security

Reputation is non-consumable, so that we don't have double-spending issue with PoR; reputation needs time to be accumulated, so that naturely PoR is resistant to Sybil attack.

As for service denied attack and system taken over(by attackers) risk, it depends on the concrete implementation of PoR in considered projects.

V.4 General Prototype

A blockchain system which applies a PoR protocol would typically contain two parts:

A reputation system, which defines how the "reputation value" of each node should be quantified - depening on which factors the reputation is calculated, following which kind of formulations, and how it would change along with nodes interaction and/or system operation.

A blockchain based consensus protocol that - through all nodes' reputation value - make them having agreement about block generator nodes' identity and about the lastest blockchain status, thus having agreement on records and data verification for the ledger.

Based on this design, we could fromalize the problem of designing a prototype of a PoR consensus protocol for public or controlled blockchain system as follows. Assume N_{max} the size of maximal possible joiners for the network, N the current number of users - registered or not, depending on whether the blockchain is controlled. An individual participant could be represented by n_i, i $\in$ N, where n means "node". Each node stores all other peers' public key in local, it's allows every node to complete data verification tasks(for transactions and for blocks). Transactions proposed from n_i to n_j is denoted as Sig(x_i^j): where x_i^j $\in$ R - a real number representing considered transaction's index - signed by n_i's private key.

VI State of the Art of the Proof-of-Reputation

As mentioned in the last sectino, the PoR is a new concept of consensus protocol. Its idea is to introduce the reputation—or the trustworthiness of a node in the network—as the weight that this node influences the consensus. However, how to calculate reputation, how to make the reputation of the node affect the consensus process - block generation, chain fork selection, choice on incentive mode, and so on, different researcher groups have proposed different designs and/or methods. In this section, we will highlight 4 different designs of existing PoR based projects.

VI.1 PoR p2p

Background

The first model is from “Proof of Reputation: A Reputation-Based Consensus Protocol for Peer-to-Peer Network”, published in 2018 by National University of Defense Technology in China.

Design Overview

The consensus protocol in this paper is designed for the permissioned blockchain: before joining the network, the identity of the node needs to be verified and recorded by the system.

Design for consensus layer

The block generation and the fork selection are decisive in this system: nodes can collect transactions broadcast on the Internet into their own pool of pre-committed transactions. When the number of transactions in the pool exceeds the threshold, they can be assembled into one transaction block. However, the node can sign and publish this block only if it has the highest reputation value among the nodes involved by the transactions within this block.

Design for reputation model

In the reputation model designed by the research team, the reputation of the node cannot be costed and transferred, and it can accumulate as the node participates in the network transactions (there may be negative growth). The numerical value of reputation is mainly used as an incentive for nodes to maintain and update system ledgers.

The change in reputation is mainly due to the system rewards obtained by participating in the ledger update, as well as the rating scores obtained from other nodes in ordinary transactions. In order to exclude the influence of human subjective evaluation, the rating score only includes two cases: positive evaluation or negative evaluation. In this case, only 1 bit needs to be used to store the scores that affects node’s reputation value. The research team calls it the “single-bit reputation system”.

VI.2 Aigents

Background

The second model is from “A Reputation System for Artificial Societies”, published in 2018 by Aigents Group in Russia and SingularityNET Foundation in Netherlands.

Design Overview

The Aigents team wants to - through a reputation value model - introduce the concept of ”liquid democracy” into their blockchain network: when a node gets good reviews from other nodes, it’s equivalent to the latter giving the former the positive impact of their own reputation. Therefore the former gains a higherweight in the process of cosensus(and other potential operations). This is like a democratic voting process that, in some systems, voters may not vote directly, but delegate their voting rights to other delegates, while retaining the right to withdraw their authorization.

Design for consensus layer

The PoR designed by the research team is a variant of PoW. The nodes still compete with each other to win the opportunity to participate in the ledger maintenance and accept the token rewards, the only diffrence is that tokens placed in the competition are the reputation value of the node, the rewards are also the reputation value.

The research team tried to adopt their protocol for the general public systems, especially social networks. For this reason, the storage and confirmation of reputation status is very important. They proposed a gossip agreement to solve this problem: during the operation of the system, set a special reputation calculation cycle. All nodes broadcast the reputation data status of themselves and their own connected nodes in the network; for the reputation value of a certain node i, if node j receives enough consecutive and consistent data states, it regards it as valid. If an inconsistency (controversy) occurs, node j needs to warn the system’s monitoring service and declare the source of the dispute, and validate the most important consecutive status.

Design for reputation model

The Aigents team considered five factors and four roles to construct a node's puretation. These roles are: a.“follower”. When node i follow node j, it means that ratings from j to its connected nodes directly affect rating from i to the same nodes; b.“peer”. Two nodes lacking the ability to influence each other's reputation and given ratings. c.“Opinion ledaers”. Nodes that are followed by a large number of nodes. Their ratings affect greatly the reputation of nodes being evaluated. d.'connecter'. Nodes that can connect two peer groups that are not connected.

The mentioned roles play an important role in five factors, these factors are:
a. The direct rating from node i to node j. This will affect the reputation value data of j in front of followers of i and i.
b. The indirect rating from node i to node j. This rating could be viewed publicly. For example, after the node generates a block, involved transactions participants could give a rating to this block; or the node leaves work like articles on the blockchain, nodes could evaluate its work. These ratings affect the reputation value of node j in public.
c. Implicit indirect evaluations. For example, in social networks such as forums, nodes' post could receive comments. These comments are not direct ratings, but also contain positive or negative emotions.
d. Implicit direct evaluation. For example, in social networks, node i quotes and/or excerpts from the comments or articles of node j.
e. The financial status of the node itself. Holding stakes, conducting transaction activities can be regarded as a positive evaluation, while canceling transactions or returning goods can cause a decline in reputation.

VI.3 Gochain

Background

This model is a PoR protocol proposed by its business team in 2018. The Gochain blockchain project is developed based on Ethereum platform, dapps and smart contracts running on Ethreum could be transformed on GoChain without any obstacles.

The Gochain team aims on 1300tps; as for energy saving, their goal is to save 100 times more energy than Bitcoin or Ethereum. Maintaining decentralized features and enabling more flexible intelligent contracts are also part of their work plans.

Design Overview

This protocol is based on the Clique algorithm which belongs to the serie of Proof of Authority(PoA) algorithms[20], created by the Ethereum community. Its mode of operation is that among all nodes within the network, only a selected set called authoritative nodes(or super nodes) could play the role of "miners", they have the right to sign and publish - in a polling manner - the transaction blocks.

Design for consensus layer

Firstly, the Gochain team noted the fact that corporate reputation and organizational resources far exceed personal credit and personal resources, thus they decided they not to allow individual users to become authoritative nodes: only 50 listed companies with sufficient reputation and assets can enter the initial system's authoritative nodes committee. Besides, unlike the blockchain that uses the Clique algorithm which is currently a side chain of Ethereum, the Gochain team has built its own blockchain system and network.

In Gochain's PoR protocol, the authoritative nodes are responsible for the assembly and signing of subsequent blocks in a polling manner, so there is a concept of "node on duty": block published by the "on duty node" enjoys a higher weight, thus reducing the risk of chain fork.

The concept of "rounds" is preserved. Which means, any miner nodes can only propose one block in the same round, and then they need to wait for an enough long interval to propose an another block in a certain subsequent round, this design could curb the ability of the malicious miner node to use the authority to destroy the system service.

Design for reputation model

The renewal of the authoritative node relies on the binary voting from the membership of the committee. When a miner receives enough negative votes, it will be removed from the committee; when there is a vacancy in the committee seat, and a normal node receives enough affirmative votes, it can enter the committee. The agreement proposes the concept of "epoch" as a cycle of updating the list of committee members.

Since the concept of reputation is only once used to determine the initial auhoritative nodes list, in Gochain protocol, we didn't implement any mathematical models for reputation values.

VI.4 Bitconch

Background

This model was proposed by a business project "Bitconch", on October 3, 2018, the research team of Bitconch released their newest test results, showing that with their public and distributed blockchain network configured in 5 different countries, they have achieved a peak speed up to 120,000 TPS, which is one of the fastest blockchain under the same operating conditions at present.

Design Overview

The design of this model consists of 2 parts: a Proof-of-reputation consensus protocol and a corresponding reputation system called "Bit-R". Their PoR protocol is a combination of a "dPoS-like or dBFT-like leader election mechanism" and "classical PBFT algorithm". It's the basic protocol of Bitconch's blockchain system; as for the Bit-R system, it uses the quantified results of users' trustworthiness, activity and contribution, to build the portraits of users' individual behavior, thus provide a reference to the weight of each user for the election phase of their protocol.

Design for consensus layer

- Here's a concrete description about how Bitconch's PoR protocol works:

a. The nodes that have the the 5% highest reputation value form a candidates pool, each node among them is possible to be chosen to become the leader node. The membership of this pool updates quartly.

The size of the candidates pool varies from 50 to 300, depending on the scale of the Bitconch blockchain network.

b. With a priorly determined random number generation algorithm and the candidates pool, the system conducts the election phase by selecting 1 node to become the leader, then (M-1) other candidates - at the same time - to become voter nodes.

M is a natural number, the election of the M nodes - the leader and the voters - is re-executed for each round within the system.

c. The leader node and the voter nodes make consensus through the PBFT algorithm: the leader takes charge of the broadcast of the uncommitted transactions; the voters validate these transactions(or the opposite) - in Bitconch system we describe this step as a voting action; then the leader synchronizes the voting results and the round number with all the nodes in the network.

If more than 2/3*m nodes returned their voting choice(namely, committed their validation), this round is considered as succeed, the leader and the voters gain benefits in terms of their contribution in Bit-R system.

During a successful round, a transaction that received enough certification votes is validated(confirmed). It will be added into the ledger while the leader synchronizing all the nodes. The nodes involved by a confirmed transaction gain benefits in terms of their activity in Bit-R system.

Design for reputation model

- Here is the description of reputation model within the Bitconch system:

a. Activity: D(E,t)=$\sum_{i=1}^{k} E_i^{log(D_r)}$

E_i represents the asset weight of a transaction i, D_r represents the reputation weight of the other party of transaction i.

Thus the "activity" parameter of an user could be quantified by the transactions that he/she has participated, and the nodes that he/has has interacted with. The logarithm function is used here to avoid potential Sybil attacks - nodes with low reputation weight are hard to influence other one's activity.

b. Coin age: T(s,t)=$\beta + \alpha \log(S_t)$

S_t represents the length of time that current user keeps the Bitconch system tokens. The Bitconch system take the users who hold system rights for long-term are more trustworthy.

The logarithm function is used here to limit the potential Matthew effect(first-mover advantages).

c. Contribution: C(N,t)=$\sum N_{file} + logN_{Rnd}$

The "contribution" parameter reflects the frequency that nodes contribute to the normal operation of the system, especially including files sharing($sumN_{file}$) and ledger updates($logN_{Rnd}$)

d. Summary: Based on 3 above parameters, the Bit-R is able to describe the integrity of each user, thus able to give nodes' integrity as a proof, to allow them to participate to the consensus, to contribute their network resources, and to gain rewards token.

VI.5 Repucoin

Background

Repucoin was proposed in February 2019 by a research team from the University of Luxembourg. The proudest design objective reached by Repucoin is the resistancy to 51% computing power attack. Repucoin system calculates voting rights based on migners' reputation. By builing a model of reputation with stringent mathematcial literacy, the system requires miners to accumulate long-term, continuous and honest mining operations.

A Repucoin blockchain can support more than ten thousands tps, even much higher than Visa which could support around 1700 tps.

Design Overview

Repucoin blockchain system is deterministic: generally, only one node has the right to package and sign the next block at each round.

The generation of blocks is cooporative: not everyone but only a selected set of nodes could be randomly elected to become block generator. This group takes also the validation of new blocks in charge.

The selected group of nodes is called as the "cosensus group", it is constitued by nodes who have the highest reputation scores. A ramdonly chosen leader is elected from the membership at each "epoch" and this leader takes charge of the production of blocks of the whole current "epoch". Epoch is a period of time determined by a chunk of blocks on blockchain.

Blocks in Repucoin system are divided into two types: keyblocks and microblocks. Miners use PoW protocol rules to compete to become the leader(block generator) for next epoch, by resolving Repucoin's original hash puzzle. Microblocks are signed and proposed by the current leader to record the transactions into the blockchain.

Design for consensus layer

The consensus process in Repucoin system could be divided into two parts: a periodical election based on PoW mechanism, then a regular blocks validation process based on PBFT mechanism.

During the election phase - which is also the beginning of each epoch - a consensus group having X members is firstly updated. The size of X is determined by meeting a target percentage in global decision power, and the decision power is directly and only based on nodes' cumulative reputation scores.

Design for reputation model

The reputation scores calculation model is designed as a sigmoid function: for beginners and high scores holders, the changing on their scores is slow or even towards stagnation. As for mature participants, users who joined the system for a while and honestly acted so long, they have the opportunity to enjoy potential high-speed returns.

As the calculate method is a sigmoid function, system designers could control the slope and also inflection point of function by two parameters that can be predetermined. Here's the simplified equation for reputaion score R:

$$R = min(1, H * (Ext + \frac{1}{2} * (1 + \frac{y1 * y2 * L - a}{\lambda + |y1 * y2 * L - a|}))) \tag{1}$$

where λ and a are two parameters pre-defined by the designers to adjust the slope and the inflection point.
H is a boolean value, which is set to 1 for every newly joined user, and could be reset to 0 once if a node has misbehaved(especially as a miner).
Ext is a reputation judgement from external resource.
The meaning of y1 and y2 are slightly more complicated: y1 is calculated by the percentage

VII Conclusion

Blockchains, with their core characteristics of decentralization, anonymity, tamper-resistancy, forge-resistancy and auditability, have shown their potential to transform the traditional business.

In this article, we provide a complete overview of blockchain models and blockchain basic rules(consensus protocols). We first outline blockchain technology, giving a general model of the system itself. Then we discuss the standard consensus protocols used in blockchains. We analyzed and compared these protocols from different perspectives.

In addition, we highlight the concept of proof-of-reputation, explaining its potential advantages to the exsiting ones by listing the potential solution to some challenges and problems by implementing PoR, and summarize some of the existing por blockchain projects for indicate their features and for show how the real PoR protocols look like. At present, the applications based on blockchain are rising, and we plan to do further researches and works on original PoR based blockchain system in the future.

Appendix

List of abbreviations

The following table describes the significance of various abbreviations and acronyms used throughout the thesis. The page on which each one is defined or first used is also given. Nonstandard acronyms that are used in some places to abbreviate the names of certain white matter structures are not in this list.

Abbreviation	Meaning	Page
P2P	Peer to Peer	2
PoW	Proof of Work	9
PoS	Proof of Stake	2
dPoS	delegated Proof of Stake	9
dPoW	delayed Proof of Work	14
PoET	Proof of Elapsed Time	15
PoC	Proof of Capacity	18
PoB	Proof of Burn	18
PBFT	Pratical Byzantin Fault Tolerance	2
dBFT	delegated	9
FBA	Federated Byzantine Agreement	9

Author details

[1]LIRIS Laboratory, National Institute of Applied Sciences of Lyon, 20 avenue Albert Einstein, 69100 Villeurbanne, FR. [2]The University of Passau, Innstraße 41, 94032 Passau, Germany.

References

1. G. W. Peters, E. Panayi, and A. Chapelle, "Trends in crypto-currencies and blockchain technologies: A monetary theory and regulation perspective," 2015. [Online]. Available: http://dx.doi.org/10.2139/ssrn. 2646618
2. G. Foroglou and A.-L. Tsilidou, "Further applications of the blockchain," 2015.
3. A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in Proceedings of IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2016, pp. 839–858.
4. B. W. Akins, J. L. Chapman, and J. M. Gordon, "A whole new world: Income tax considerations of the bitcoin economy," 2013. [Online]. Available: https://ssrn.com/abstract=2394738
5. Y. Zhang and J. Wen, "An iot electric business model based on the protocol of bitcoin," in Proceedings of 18th International Conference on Intelligence in Next Generation Networks (ICIN), Paris, France, 2015, pp. 184–191.
6. M. Sharples and J. Domingue, "The blockchain and kudos: A distributed system for educational record, reputation and reward," in Proceedings of 11th European Conference on Technology Enhanced Learning (EC-TEL 2015), Lyon, France, 2015, pp. 490–496.
7. C. Noyes, "Bitav: Fast anti-malware by distributed blockchain consensus and feedforward scanning," arXiv preprint arXiv:1601.01405, 2016.
8. Castro, Miguel, and Barbara Liskov. "Practical Byzantine fault tolerance." OSDI. Vol. 99. 1999.
9. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. 2008.
10. Vasin P. Blackcoin's proof-of-stake protocol v2[J]. URL: https://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf, 2014, 71.
11. Crain T, Gramoli V, Larrea M, et al. DBFT: Efficient byzantine consensus with a weak coordinator and its application to consortium blockchains[J]. arXiv preprint arXiv:1702.03068, 2017.
12. Mazieres D. The stellar consensus protocol: A federated model for internet-level consensus[J]. Stellar Development Foundation, 2015.
13. Schwartz D, Youngs N, Britto A. The ripple protocol consensus algorithm[J]. Ripple Labs Inc White Paper, 2014, 5.

14. Chen L, Xu L, Shah N, et al. On security analysis of proof-of-elapsed-time (poet)[C]//International Symposium on Stabilization, Safety, and Security of Distributed Systems. Springer, Cham, 2017: 282-297.
15. P4Titan. Slimcoin: A peer-to-peer crypto-currency with proof-of-burn, 2014.
16. Dziembowski S, Faust S, Kolmogorov V, et al. Proofs of space[C]//Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2015: 585-605.
17. Larimer D. Delegated proof-of-stake (dpos)[J]. Bitshare whitepaper, 2014.
18. Komodo: An Advanced Blockchain Technology, Focused on Freedom
19. Solana: A new architecture for a high performance blockchain v0.8.13, 2018
20. De Angelis S, Aniello L, Baldoni R, et al. Pbft vs proof-of-authority: applying the cap theorem to permissioned blockchain[J]. 2018.
21. Gilad Y, Hemo R, Micali S, et al. Algorand: Scaling byzantine agreements for cryptocurrencies[C]//Proceedings of the 26th Symposium on Operating Systems Principles. ACM, 2017: 51-68.
22. gochain.io/assets/gochain-whitepaper-v2.1.2.pdf
23. YUAN Yong, WANG Fei-Yue . Blockchain: The State of the Art and Future Trends[J]. ACTA AUTOMATICA SINICA, 2016, 42(4): 481-494
24. bitcointalk.org/index.php?topic=3026750.0
25. www.reddit.com/r/Vechain/comments/97zmoy/
26. www.coingecko.com/fr/pièces/
27. www.feixiaohao.com
28. coincheckup.com
29. blocktivity.info
30. bitinfocharts.com

.