Technical report - Privacy and Security in Blockchain: Proof-of-Reputation based Consensus

XING Yidi

May 10, 2019

Abstract

The appearance of blockchain technology allows people to build a distributed, decentralized and tamper-resistant ledger through a trust-less based P2P network, its potential prospect for various application scenarios is widely favored. However, most of the existing blockchain system are limited by their consensus mechanism, thus it is difficult to maintain a balance among the performance, the energy efficiency and the decentralization feature of the system.

To fill this gap, we propose a consensus protocol based on reputation mechanism: Proof-of-Reputation(PoR), which could guarantee the reliability and the integrity of transactions results in the system, having at the same time a much higher performance than bitcoin-like blockchain, and it also effectively ensures the robustness of system decentralization properties. In PoR, it's not necessary to issue cryptocurrencies, instead of that, we could use reputation value as incentive for the maintain and the update of the system, thus could be integrated in general application scenarios.

Keywords: blockchain · consensus protocol · reputation system · decentralization

1 Introduction

Blockchain technology was introduced by Nakamoto along with Bitcoin applications in 2009, it combines the utilization of encrypted hash functions, digital signature, Merkle tree, consensus protocol and P2P network, it could be used not only for financial trading systems, but also Scientific research, resource management, political domain, etc. Blockchain system is a distributed database system based on decentralized P2P network, it could record public ledger – sorting group of transactions in chronological order and encryptedly linking transaction blocks, thus enable trust-less based distributed applications to be operated in the system.

Consensus protocol is an important part of the blockchain system. The existing consensus protocols mainly face 4 serious challenges: system performance, energy efficiency, security and decentralization feature. We introduce the reputation system based on the blockchain architecture and use reputation as the incentive, in order to solve the above 4 problems.

Reputation represents the trustness of a node in the network. Within the p2p network, applying reputation system can promote the mutual trust of the nodes, so that the update of the ledger can be carried out in a cooperative and deterministic manner, omitting the "mining" step that consumes a lot of resources, and the convergence speed of the consensus is also much higher than with the probabilistic mode. The ability of a node to influence the consensus is determined by its reputation determined during interaction with other members within a certain time frame, and the reputation needs time to be accumulated, it cannot be exchanged, spent or transferred, so the decentralization properties and security of the consensus protocol could also be highly guaranteed.

Using our protocol, any P2P application can establish a reputation layer to log transactions in a secure, auditable manner, and each participant's reputation can be objectively evaluated without being manipulated by a third party.

The rest of this paper is organized as follows: in the second section, we will introduce the existing work in this field; the third section will mainly explain the threat model that the consensus protocol will face, and explain the security strategies of the protocol in response to these attacks; in section 4, we will describe the reputation system itself and how the PoR consensus protocol actually works; section 5 is about the experimental data and results analysis so far; then finally, we will review and summarize our work, looking to the nearest Future work direction in section 6.

2 Background

In this section, we will introduce the theoretical background of our research, our motivation, and try to answer to 3 questions that the audience probably hold:

- what is a consensus protocol, and what is a consensus algorithm?

- How do we understand their role within a blockchain system?

- And finally, why we are such interested to the consensus protocols, why this notion is worth doing advanced researches on it?

Blockchain network

All existing blockchain systems are based on networks, this kind of network is called a Peer-to-Peer Network(p2p network), where all the nodes in the network possess a equivalent class, and they all play simultaneously the role of a 《server》 and a 《client》 .

Within a such network, the decisions of every kind of transactions are executed, only if all the peers have got a consensus. This sounds like a troublesome process, however, this also indicates that the system does not need to rely on any 《self-proclaimed trustworthy》 third party, neither need to be controlled or supervised by any authoritative organizations.

This idea is called as 《decentralization》 , its purpose is to allow to every networks nodes equally participating into system decisions making, let all the users together maintain the operation of the system, and make every system decision being transparent to every user.

However, each individual node could only locally take decisions, then communicate at most with all its connected nodes, so how to reach consensus in the scale of the whole network, make this process efficient, and guarantee a good consistency(low probability of divergence appeared): we need a suitable consensus protocol to deal with these exigencies.

Consensus protocol

The consensus protocol is the rules of a blockchain system, which are about the proposals that influence the system status, and about how the consensus could be reached within the system network. The consensus algorithm refers to the mechanism that implements the corresponding rules.

The specific role of the consensus protocol

In a blockchain system, the consensus protocol concretely defines the rules for the below issues:

a) the selection of blockchain forks;

b) the generation and the validation of the blocks;

c) the incentive mechanism, and the possible distribution of the reward tokens;

d) the verification and the completion of transactions;

e) the gossip mechanism of the blockchain network;

f) the interaction way of the nodes(users);

g) the way of data transmission among the nodes; etc.

Main challenges faced by the consensus protocols nowadays

a) Performance 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.

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.

b) 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.

c) 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.

d) 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.

e) Centralization issue: As for 2017, 80% of all blocks generated in Bitcoin network are mined by large mining companies in Iceland and in China, 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.

3 Related Works – Consensus algorithms

Presentation of 16 consensus protocols

In order to let the audience get a better understanding about the evolution and the state of the art of the blockchain consensus protocols, we list and explain 16 different protocols below. The content of the explanation includes a summary introduction, their concrete internal mechanism, and an analysis about their advantages and defects.

1) proof-of-Work(PoW)

PoW is the first consensus protocol applied to the blockchain system, as a protocol, it mainly answered to 4 questions below:
• Who package transaction blocks and then update the ledger(maintain the system operation)?
• Why users would have the motivation to take care of the update of the ledger?
• How the rewards of maintaining the system operation are distributed?
• How do we locally determine the main chain while forking occurs?

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.

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

Defects 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".

2) Proof-of-Stake(PoS)

Proof-of-Stake is a variant of PoW. Its idea is to replace the notion of "work(or, computing power)" by the notion of "interests(or assets, stakes)"

The process mechanism of PoS is basically the same as PoW, only differs at the block generation method: The "resolving hash puzzle" step is canceled, instead of that, in order to update the ledger then gain the system rewards, nodes need to 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

Advantages of PoS:
• Attacking a PoS system is very harmful for the attackes, because they are also interests holder of this system.
• PoS is resistant to the "scale economy": as in PoW, 10000 miners that pay 1 dollar electricity fee per minute, they have actually a pretty low computing power, although 1 miner who pays 10000 dollars electricity fee per minute, get generally a very high computing power. While in PoS, the interest that can be brought in 1 dollar is constant.

Defects of PoS:
• "Nothing-at-the-stake attack": seeing the fact that mining in a PoS system is barely free, 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 system inflation, then serious depreciation of system assets.

3) delayed-Proof-of-Work(dPoW)

The idea of dPoW is – based on an existing blockchain which uses PoW or PoS protocol – constructing a new blockchain system.

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, then rely on the security of the original one to verify these transactions. The selected nodes return the results of transaction processing to the new system at last, and here comes the notion "delay" in the title of this protocol.

Advantages 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

Defects 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.

4) PoET(Proof-of-Elapsed-Time)

The PoET protocol was introduced by Intel research team, 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".

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.

Advantages 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.

Defects 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.

5) dPoS(delegated-Proof-of-Stake)

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: 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.

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.

Advantages 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.

Defects 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.

6) Algorand

The algorand protocol was proposed by MIT's research team in 2017. It's a protocol based on PoS, PBFT 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)".

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.

Advantages 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.

Defects of Algorand:
• Despite its complex process, there is no direct results showing that Algo-

rand has a better performance than other election mechanism based protocol such as dPoS.

7) 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.

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.

8) PoBurn

The PoBurn protocol is a variant of PoW, 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.

9) PoA(Proof-of-Authority)

PoA protocol runs based ona pre-determined committee of nodes called signers; 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.

10) PoHistory

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

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.

11) 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".

12) PBFT（Pratical Byzantine Fault Tolerance）

PBFT is a state machine replication algorithm. 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.

Defects of PBFT:

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

13) dBFT(delegated Byzantine Fault Tolerance)◦

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

Advantages of dBFT:

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

Defects of dBFT:

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

14) 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.

15) Ripple consensus

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

Defects of Ripple:
- The fault tolerance percentage is only 20% for Ripple system.

16) Stellar consensus

The Stellar is also a variant of FBA protocol. Unlike in Ripple, the Stellar system does not pre-set trusted nodes, or in other words, there is no UNL for the validating nodes. 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.

3.1 Analysis

Various consensus algorithms have different strengths and drawbacks. Table I. brings an assessment around various consensus algorithms, and we use the properties considering following.

Algorithm	Applicable blockchain type*/Node identity	Energy Saving	Tolerated power of adversary	Examples	Multi-center mechanism**	Block generation type
PoW	public/public	No	25% Computing power	Bitcoin, Ethereum, Litecoin, Dogecoin		Competitive & probabilistic
PoS	public/public	Partial	50% Stake	Ethereum (intend), Peercoin, Nxt		Competitive & probabilistic
dPoW	public/public	Yes(side chain)	25% Computing power	Komodo		Competitive & probabilistic
dPoS	public/public	Partial	< 51% Validators	BitShares, Steemit, EOS, Lisk, Ark	Yes	Cooperative& deterministic
Algorand	public/public	Yes	33.3% Byzantine Voting Power	Algorand	Yes	Competitive& deterministic
PoA	permissioned/ public	Yes	50% of online stake	POA.Network, Ethereum Kovan testnet, VeChain	Yes	Cooperative& deterministic
PoET	public/public	Yes	Unknown	HyperLedger Sawtooth		Competitive& deterministic
PoHistory	public/public	No	50% Stake	Solana		Competitive & probabilistic
PoBurn	public/public	No	25% Computing power	Slimcoin, TGCoin		Competitive & probabilistic
dBFT	public/private	Yes	< 33.3% Faulty Replicas	Neo	Yes	Cooperative& deterministic
PBFT	permissioned/ private	Yes	33.3% Faulty Replicas	Hyperledger Fabric, Dispatch		Cooperative& deterministic
FBA	public/public	Yes	33.3% Byzantine Voting Power	Ripple, Stellar	Yes	Cooperative& deterministic
Ripple	public/public	Yes	< 20% Faulty Nodes in UNL	Ripple	Yes	Cooperative& deterministic
Stellar	public/public	Yes	33.3% Byzantine Voting Power	Stellar	Yes	Cooperative& deterministic

1) Applicable 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 4 concepts in terms of type division—
a) the public chain, in which all member nodes can freely join and leave;
b) the private chain, completely private, with strong third party providing node identity assurance and controlling node permissions distribution;
c) the consortium chain, between the completely decentralized public chain and the private chain with a "pseudo-decentralized" image, we have the "partially guaranteed decentralization" consortium chain – also called as "semi-private chain", it is generally operated by an organization that opens the inscription access to qualified users and ensures that the identity of the nodes is audited and documented;
d) the permissioned chain, a concept that is the set of private and consortium chains, which means a blockchain system that nodes cannot be joined without being licensed and identity audited.

In Table 1, the consensus algorithms applicable for the public chain system - unless otherwise specified - also support the operation of the permissioned chain; the algorithms applicable for the permissioned chain system don't support the operation of the public one by default; in addition, unless otherwise specified, the algorithms applicable to the permissioned chain system defaults to be applicable for the private chain and the consortium chain.

As for the node identity, only with PBFT serie algorithm which serve for permissioned chains, node identity could keep staying private.

2) Energy Saving: As for PoW and some of its variants such like PoBurn, PoHistory, the demand on hash computing power make the system environment unfriendly; as for PoS and its variants such like dPoS, dPoW, the hashing related step is removed, but the competition consensus mode – or in other words, a pure decentralized block generation phase – is kept; Regarding PBFT, FBA series protocols, there is no more concept of mining, the block generation phase is somehow centralized and thus saved power tremendously.

3) Tolerated power of adversary: 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; 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, which makes it only maintaining correctness when the proportion of faulty nodes in a unique node list are lower than 20

4)Examples: List of the most representative projects of each corresponding protocol in the real life.

4)Multi-center mechanism: A multi-centered consensus protocol refers to establish a committee, by means of competition or voting; then the committee centralized the maintain of system service offering and ledger update, for a certain next time period.

The nature of this mechanism is allow the centralization appearing when decisions need to be made for the system at some points of time, in order to get benefit from the efficiency and the security that a closed and small system could have.

In some articles, we alos call it as "weak centralization".

5) 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.

Cooperative consensus are all able to generate new blocks deterministicly, as for competitve ones: both probabilisitic and determisitic modes exist. Regarding the huge size that a real world system generally has, the probabilistic mode is however limited in term of scalability.

4 Relative works – existing Proof-of-Reputation approaches

PoR is a new concept of consensus protocol. Its idea is mainly 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 researchers have proposed different designs and/or methods. In this section, we will highlight 3 existing designs that are named PoR protocols.

4.1 PoR p2p

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.

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.

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".

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.

4.2 Aigents

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

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, only the 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. 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.

4.3 Gochain

The third model is the PoR protocol proposed by the GoChain team in their own altercoin project GoChain (currency code GOC). Their protocol is basically based on the Clique algorithm which belongs to the serie of Proof of Authority(PoA) algorithms, created by the Ethereum community. Its mode of operation is that among all nodes within the network, only some of them called authoritative nodes(super nodes) could play the role of "miners", and have the right to sign and publish - in a polling manner - the transaction blocks.

The improvement of the Clique algorithm by GoChain's development team is mainly reflected in two aspects. Firstly, they reiterated the fact that the companies' reputation and organizational resources far exceed personal credit and personal resources, thus decided they not to allow individual nodes to become authoritative ones: only 50 listed companies with sufficient reputation and assets can enter the initial system's authoritative nodes committee. Secondly, 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 the Clique algorithm, the authoritative committee operates in such a way that the miners are responsible for the assembly and signing of subsequent blocks in a polling manner: so there is a concept of "miners on duty", block published by the "on duty miner" enjoys a higher weight, thus reducing the risk of chain fork; the concept of "rounds" is preserved, any miner node can only propose 1 block in the same round, and then need to wait for a long enough 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.

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.

4.4 Bitconch

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.

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.

• 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.

- 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.

Index	Blockchain system type	Type of consensus process	Block generation process	Broadcaster of the submitted transactions
PoR P2P	Permissioned	Cooperative	Deterministic	Block generator node
Aigents	Public	Competitive	Probabilistic	Block generator node
Gochain	Public	Cooperative	Deterministic	“On-duty” authoritative node
Bitconch	Public	Cooperative	Deterministic	Leader node(that is, primary node, in PBFT algorithms)

Index	Temporary centralization during the block generation phase	Type of leader election process	How does the reputation involve the consensus process as weight	Factors for reputation quantification
PoR P2P	Yes	Deterministic	The node with highest reputation, involved by a pending submission transaction block, win the block generation right.	**Ratings** made by the other party in every transaction; **coin age**
Aigents	No	Nonexistent	The nodes with higher reputation get higher probability to win, as an imitation of “mining action” in PoW	**Ratings** made from other nodes in the community; **activity**
Gochain	Yes	Deterministic	Nodes with good reputation could be chosen to become authoritative nodes, and could vote to invite/exclude the others from the validator list.	**Subjective judgements** made by Gochain development team and all other validator nodes
Bitconch	Yes	Random	The nodes with 5% highest reputation value become systematically validator nodes.	**Activity**; **coin age; contribution**

Index	Reward Token	Reward token receiver nodes	Whether the reputation value as a scarce resource will eventually tend to be centralized
PoR P2P	Reputation value	Block generator – node with highest reputation in current transaction block	Not evident
Aigents	Reputation value	Block generator – node that win the “mining”	Yes

		competition	
Gochain	Nonexistent	Block generators – nodes in the authoritative nodes committee	At least partially
Bitconch	Reputation value	The m elected nodes – nodes that have at least 5% highest reputation	Yes

Index	Language	Test network	Performance results
PoR P2P	Python	Proper Prototype	Tps<100 (participants varies in [100, 500], transactions/block varies in [200, 1000])
Aigents	Python	Steemit, Google+, Ethereum	Unknown
Gochain	Go	Gochain blockchain network	1,300 tps
Bitconch	Go	Bitconch testnetwork beta1.0	100,000 tps

5 The description of our approaches

Reputation model

1) Nodes begin with equal local reputation values Rl when they are created.

2) The reputation value is divided into the standard reputation Rst and the service reputation Rsr. The former is global, representing the trustness of the node in the eyes of the system, and determines the ability of the node to influence the consensus; the latter exists between any pair of nodes i and j, representing the impression and evaluation of i on j.

2.1) The standard reputation of any node i is calculated based on its local reputation and the service reputation from all other nodes connected to i.

3) The renewal of the reputation value is periodic, and this cycle is named "Epoch". The epoch is directly related to the increment of blocks. In our conception, for every 100 blocks produced, the number of epoch increases by 1.

Change in reputation

1) We divide the nodes involved in a transaction into two parties, the "providers" of the service and the "requesters". After each trading completed, the requester will evaluate each provider of the transaction by giving them separatly a rating, which will cause the service providers' reputation to change. In order to rule out the influence of human subjective factors, the way of evaluation is limited to giving a real number rating between -1 and 1 for this service. For actual applications in the future, in order to facilitate the user's understanding, the interval of the rating can be transferred to 0 to 10, or 0 to 100 and the like, then, the rating records will be stored locally on both sides.

2) Whenever node i participates in a round to the update of ledger, the system will give a rating to node i as the "requester" of the service, and the system's rating can range much larger than the average node. This will also cause changes to the reputation of i. This rating will be directly applied to the recalculation of the local reputation Rl of node i, the result should be stored locally in all nodes participating in the maintenance of the ledger at the same round.

Standard reputation and service reputation calculation

1) By averaging the ratings in the record, the evaluation of node i for any other node j is recorded as the service reputation score Rsr of i versus j. Therefore, there is no way to get any additional benefit between the two nodes by scoring each other very many times.

2) By applying the Eigen Trust algorithm, based on the service reputation score

Rsr from all connected nodes of node i and the local reputation Rl of i, the standard reputation Rst of i can be calculated. The standard reputation of i is stored in i and all its connected nodes.

Block generation and incentive distribution

At the beginning of each epoch, all nodes can participate in the competition to become an "public trustee" node. The algorithm used in the competition is PoS. Here, we use the reputation value of the node as PoS token. The result of the algorithm running is to generate a list of public trustees, this list is broadcast across the network.

The public trustees are responsible for the collection of transactions and the assembly of transaction blocks in a polling manner. At the end of an epoch, all nodes on the list will be rewarded with a reputation value increment, and then the list will be emptied, waiting for the competition at the beginning of the next epoch.

The trustees which cheat while producing a block can be reported by other nodes, the revealer will receive a reputation value reward, and the cheat node will suffer a huge loss of reputation value.

Blockchain fork selection

Because the generation of the block is cooperative and deterministic, there is no fork in the case of a correct operation of the protocol.

6 Threat Model and Security Analysis

a. Bad-Mouthing attack

Malicious nodes could harm system service by consciously give unfair and low rating to other nodes that they have interacted.

Mitigation methods:
1) The nodes - that provide services through the system application - receive the reputation scores from a large number of different nodes which requested services, and the impact of the malicious node's bad rating attacks is actually limited.
2) The nodes providing the services are obligated – for their own benefit – to pay attention to those nodes that make unreasonable bad ratings and no longer provide service to them.

Summary: Service providers that interact with enough users will not be significantly affected by bad ratings. Avoiding interaction with malicious nodes is something that service providers need to pay attention to by themselves. In the current design, our consensus protocol does not provide protection for "small providers that are being attacked by a large group of malicious nodes".

b. Sybil attack(saturated)

A large number of malicious nodes join the network during a short period, trying to harm system services and even forging ledger records (creating forgery).

Mitigation methods:
1) The newly created nodes are not eligible to be a "public trustee" for a certain period of time (for example, 1 week).
2) The new nodes begin with a low reputation value that can be accumulated through their online duration. Once the reputation value reaches a threshold (named "new_node_threshold"), the new nodes are now regarded as a normal node, and their reputation value no longer accumulate over time.
3) Set an upper limit on the number of nodes allowed to weekly join the network.

Summary: The above measures make it necessary for an attacker to control and run a large number of accounts in order to affect the operation of the "Public Trustee Committee", and the attacker needs to run its botnet for several weeks; simultaneously, the number of newly created nodes would be close to the upper limit for several weeks. The phenomenon is also easy to notice.

c. Sybil attack(lie and wait)

The attacker only begins malicious behavior after having controlled more than two-thirds of the number of nodes in the Public Trustee Committee. This strategy is more subtle and can cause even greater damage.

Mitigation methods: There is no direct mitigation.

Summary: Under our protocol rules, cultivating high-reputation nodes requires high time cost. At the same time, the randomness of each epoch committee membership requires the attacker to control a large number of high-reputation nodes and keep them online (always active). Therefore, it is very difficult to launch a covert sybil attack on our system.

d. Attacks against public trustees

Because for each epoch the public trustee list is broadcast throughout the network, dos attacks against public trustee nodes can compromise system services and significantly reduce protocol performance.

Mitigation methods:
1) When some trustee nodes continuously miss their own rounds of the value, the remaining trustees will still compete for generate blocks in order to obtain the reputation value reward, the next normal working and "on duty" trustee will determine the convergence of the main chain.
2) Nodes with reputation values exceeding a threshold named "trustworthy_threshold" can be considered as default "alternate trustees". When a large number of trustee nodes cannot be properly popped, the remaining normal working trustees can vote for a qualified alternate trustee to join the committee. In this case, the nodes that did not participate in the voting will be regarded as fault nodes and randomly chosen to be removed from the current committee.

Summary: The impact of this attack on the success rate of the protocol and the possibility of forging a transaction block requires experimentation and data acquisition in a test network.

e. Bot-net attack

There are 3 main types of botnet attacks: ddos attacks that cause system applications to crash by consuming resources; credential stuffing attacks that capture the ownership of some public key addresses in the system, where personal information and personal assets under these addresses are stolen, and these accounts may even be used for other purposes (for example, untrue reputation rating or further transactions); attacking web applications customized by the application server, using its "zero-day vulnerability" to try to take over

the server, or simply "slow attack" through a page or interface that consumes resources.

Summary: The effectiveness of the attack on the botnet depends on the reputation of the nodes being taken over. The impact of this attack on the success rate of the protocol and the possibility of forging a transaction block requires experimentation and data acquisition in a test network.

7 Conclusion and Future Work

The future work mainly includes two mutually influential parts. One is to have a profounder understanding of different consensus protocols, especially those with delegated mechanisms, or those which also trying to skip the "hash puzzle" step, etc; The main goal of this part of work is to be able to stratify and classify these protocols to summarize the current design ideas for the main existence of consensus algorithms in this research field.

The second part is the further improvement of the PoR protocol proposed by this article, especially the "threat model and security mechanism for dealing with various types of attacks", and "the pseudo codes of various methods that need to be invoked in the algorithm flow". The main goal of this second part is to use go-language to build our own PoR blockchain system and blockchain network based on existing open source platforms such as Neo or Ethereum.