Abstract

The appearance of blockchain technology allows to build a distributed, decentralized and tamper-resistant ledger through trustless-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 effectively ensure the robustness of system decentralization properties. In PoR, it's not necessary to issue cryptocurrencies, but use reputation value as incentive of the maintain and the update of system, thus could be integrated in general application scenarios.

Keywords: blockchain · consensus protocol · reputation system · decentralization

I. 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 trustless-based distributed applications to operate in the system.

Consensus protocl 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 [1]. 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 [2], 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.

II. Relative Works – Consensus algorithms

1) PoW(Proof-of-Work)

The PoW - as a consensus algorithm - was firstly proposed by Nakamoto in his first article of Bitcoin, in order to resolve "double spending" problem and thus establish a distributed trustless-based consensus. PoW is not a Nakamoto's invention, but he associated technologies such as encrypted hash function, digital signature, Merkle tree/chain and P2P network with it, constructed a usable distributed consensus system.

Within the PoW protocol, nodes that participate to the update and the maintenance of system ledger are called <<miners>>, they need to compete with each other, trying to solve a mathematical problem called <<hash puzzle>> to get the block generation right for current round, and thus receive reward tokens such as cryptocurrencies. To guarantee this benefit, every rational miner would have the tendency to invest its computing resources on the longest chain, and make this chain becoming the fastest growing and most trustworthy one – this phenomenon caused by PoW's incentive mechanism guaranteed system security.

2) PoS(Proof-of-Stake)

The PoS is an alternative algorithm proposed after PoW, which aims on resisting the impact of scale economies on distributed consensus system. The competition for the right of block generation(the maintenance of system's services) still exists, but depends on node's asset within the system, not depends on their computing resources. For example, if a node possesses 10% of equity(cryptocurrency) in the system, then the probability that it wins the right of block generation is 10% at each round.

Skip the <<hash puzzle>> step makes <<nodes would easily work simultaneously on multiple blockchain forks>>, and in different projects, research team propose different additional mechanisms to compensate for this problem in the PoS protocol.

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

The idea of dPoW is to construct a blockchain system based on an existing, secured blockchain which uses PoW; the operation and the service provision of former depends on the latter's security – guaranteed by its owned hash computing power.

Through a serie of selected nodes – called <<notaries>> - the depent system commit its pending transactions to the base system, so that the older one takes charge of the validation of new one's all transactions, and keeps the depent system in light level.

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

dPoS implements a selection mechanism. The nodes compete according to the equity they own, but the winners do not directly obtain the qualification of the current round's block generation right, they enter a witness committee instead. The members of this committee take charge of the generation of blocks in a polling manner.

Members of the committee can vote to decide whether to deprive other members of the qualifications. When there is a vacancy in the committee, ordinary nodes will be able to fill the gap by competing, according to the PoS.

5) Algorand

The Algorand protocol was proposed by MIT's research team and consists of 2 key technologies: BA* (Ultra Fast Byzantine Protocol) and cryptographic sortition; the former's role is to randomly generate a committee in each round, the members, while signing this round's block, they reveal their membership identity and restore to a common node. The purpose of this design is to enable the system to maintain the decentralization level as much as possible while verifying transactions and providing services in an efficient cooperation mode (deterministic mode). The role of the cryptographic sortition is to make at each round the committee secret: only the members themselves know their identity, thereby improving the security of the system.

Algorand's random committee membership selection is based on the number of equity owned by the nodes, it is actually a variant of PoS.

6) PoA(Proof-of-Authority)

In PoA, the system will pre-appoint some accounts and name them "validators". These nodes will be responsible for the verification of transactions, the generation of blocks, and the provision of the service in a polling manner. The key step of this protocol is the establishment of a list of validators during the establishment of the system. The holders of the nodes need to provide sufficiently reliable real-world identity certificates. PoA is therefore actually designed to serve non-public chain systems.

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

The PoET protocol was introduced by Intel, its most important innovation was the resolution of the "random leader election issue". If a node wants to participate in the update and maintenance of the system ledger to obtain revenue, it must first sleep a randomly generated length of time. A fortune node that has been dormant for a short period of time could wake up the first then sign and publish its own block.

Unlike PoW, the price a node needs to pay for win the competition is not computing power, but time, which gives a large number of possible network participants an equal chance of winning. However, in order to ensure that the length of sleep is randomly generated and that the node does pay for latency, the PoET protocol relies on the use of Intel's hardware resources to perform trusted code in a protected environment.

8) PoC(Proof-of-Space)

PoC is an alternative protocol to PoW, it’s even like a version of PoW that choose MHF(Memory Hard Function) as target hash function for the <<hash puzzle>> step. MHF is a kind of hash function where the difficulty level is directly associated with the memory space that has been invested to solve it.

Within PoC, we use simply memory space to replace computing power as our tokens for the competition phase of the protocol: the more storage space that nodes invest for the competition, the more chance they could solve the current round’s <<hash puzzle>>.

9) PoHistory

The concept of PoH is to form a so-called "hash chain" by continuously running a 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 there is no need for direct witness, neither need for nodes to communicate with each other, any node can verify the time and sequence of event occurrences locally. PoH itself is not a complete consensus protocol, but it can be modularized and combined with other consensus protocols such as PoS.

10) PoBurn

PoBurn is an alternative protocol to PoW, the tokens that nodes invest into the competition are not computing power, but their owned cryptocurrencies; and the way of investment is to <<send their coins to an unretrievable address>>, the more coins that nodes decide to abandon(or, to “burn”), the more chance they could participate to the update of system ledger and thus get rewards tokens.

However, 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.

11) BFT(Byzantine Fault Tolerance)

The BFT is not a name of any consensus protocol, but a description of the dependability of a fault-tolerant computer system facing Byzantine failures.

A Byzantine failure is the loss of system service due to any system faults that present different symptoms to different observers(we call them <<Byzantine fault>>), and that require the system reach consensus for them.

12）PBFT(Pratical Byzantine Fault Tolerance)

The main point of the protocol is that P2P consensus synchronization is required between every 2 nodes. Therefore, the risk of fork of the main chain is low, and in the case of fewer nodes, the system can have good performance(better “throughput per second” than with Proof-of-Work), and the increase in the number of nodes leads to a rapid decline in performance.

Due to the above characteristics, PBFT is mainly applied to the permission chain (or the consortium chain). It could also be applied on the public chain by combining with the election rules such as DPOS.

13) dBFT(delegated Byzantine Fault Tolerance)

The nodes select the agent nodes by voting—for example, in the <<NEO coin>> project, the agents are called “bookkeeper”—and then the agents run the BFT algorithm between them to decisively complete the block generation mission. Voting in the network is real-time and asynchronous.

14) FBA(Federated Byzantine Agreement)

The main feature of the protocol is that the requirement for each node to complete the consensus is no longer to synchronize with other nodes of the entire network, but to reach a consensus with “a certain quorum” or “a subnet representing a sufficient number of nodes”.

Each node at each time, does not need to rely on the same participants, but trustfully decides to trust a subnet to complete the consensus.

15) Ripple consensus

In Ripple's network, transactions are initiated by the client (application), and the transactions will be broadcast to the entire network via a tracking node or a validating node. Ripple's consensus is achieved between the validation/tracking nodes. Each validating node is pre-configured with a list of trusted nodes called UNL (Unique Node List). The nodes on the list can vote on the deal. It is actually an alternative version of FBA protocol.

16) Stellar consensus

In FBA, each participant knows other people it thinks they are important, it waits for most other people to agree on the deal about a transation. In turn, those important participants disagreed with the transaction until they thought the important participants agreed, and so on.

Unlike in Ripple protocol, the Stellar system does not pre-set trusted nodes, or representatives of trusted subnets. In Stellar, the nodes themselves decide which subnet they trust. Stellar Consensus Protocl (SCP) is also a variant of FBA.

III. 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.

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 permission 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.

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 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 here 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 major consecutive status.

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 sub-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.

IV. The Proof of Reputation Protocol

Reputation model

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

2) The reputation value is divided into the standard reputation R_{st} and the service reputation R_{sr}. 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 R_l 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 R_{sr} 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 R_{sr} from all connected nodes of node i and the local reputation R_l of i, the standard reputation R_{st} 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.

V. 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(thus avoid).

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.

VI. 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.