Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-022-01408-2

Practical Byzantine fault tolerance consensus based on comprehensive reputation

Jiamou Qi[1] · **Yepeng Guan**[1,2]

Received: 22 December 2021 / Accepted: 20 October 2022

Abstract
Consensus protocol is challenging due to the poor node reliability, low efficiency and decentralization. A comprehensive reputation based Practical Byzantine Fault Tolerance consensus method (CRPBFT) has been proposed. Comprehensive reputation model has been developed to evaluate the credibility of each node from service behavior and consensus process at first. The nodes with higher reputation are selected to participate in the consensus process, which helps to reduce the probability of consensus failure caused by the existence of malicious nodes. A consensus communication structure is optimized by replacing the whole network broadcast structure in the commit phase with a star one. It can be applied to degrade the network communication overhead and improve consensus efficiency. A rotation mechanism for replacing the consensus nodes regularly has been proposed to increase the degree of decentralization and enhance the robustness and dynamic of the consensus network. Some experimental results demonstrate that the developed method has excellent performance by comparisons with some state-of-the-arts.

Keywords Blockchain · Byzantine fault tolerance · Consensus protocol · Reputation model · Distributed network

1 Introduction

Blockchain is a recently trending technology in distributed network [1]. Essentially, it is a distributed ledger, which is maintained by all nodes through peer-to-peer networks. It has some characteristics of transparency, information traceability and security [2]. According to the degree of decentralization, blockchain is generally divided into public chains, consortium chains and private chains [3]. The public chains are completely decentralized and open to all nodes. Each node can send transactions and participate in the consensus process, such as Bitcoin and Ethereum [4]. The consortium chains are managed by several institution. Any node that wants to join the consensus process must be agreed by them, which means that the consortium chains are partially decentralized. The private chains are only used by individuals or private organizations, generally not open to other nodes. So it has a large degree of centralization. Nowadays, application scenarios of blockchains have been spread from cryptocurrency [5] and digital asset [6] to non-financial applications, such as smart city [7], medical information management [8] and supply chain [9]. With the development of blockchains and Web services, blockchain technology has been applied to solve security problems and trust crisis of central service platform.

Consensus protocol is the core of blockchain with important guarantee for the security of blockchain system [10]. The order of transactions received by nodes in the network is inconsistent due to network delay. Some consensus protocols have been proposed to make all nodes in the network reach a consensus within a certain time. They can be divided into Proof-of-X (PoX) and Byzantine Fault Tolerant (BFT) [11]. The PoX consensus are widely suitable for public blockchain, which mainly includes Proof-of-work (PoW), Proof of Stake (PoS) and Proof of Reputation (PoR), respectively. These consensus protocols are designed with excellent node scalability through node competition. However, they are greatly energy-consuming and have long transaction confirmation delay. The BFT consensus are used in consortium blockchain, which mainly include PBFT [12], Scalable BFT [13] and HoneyBadgerBFT [14]. These protocols are

✉ Yepeng Guan
ypguan@shu.edu.cn

[1] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

[2] Key Laboratory of Advanced Display and System Application, Ministry of Education, Shanghai 200072, China

Published online: 06 December 2022

energy-saving and can achieve higher throughput than PoX consensus protocols. However, most of them have the shortage of poor scalability.

We consider PBFT [12] consensus as an example. Since PBFT [12] doesn't consume much energy and can achieve much higher throughput, it is widely adopted in private and consortium blockchains. However, the node scalability which reflects the capacity of network to process node growth is a bottleneck for PBFT [12]. It can only scale to a few tens of nodes due to the high communication complexity. When the number of nodes in the network exceeds this threshold, the transaction confirmation delay of PBFT would be greatly increased, and the throughput would be greatly reduced [15]. In addition, when the master node is a malicious one or failure one, the view change process would be triggered. With the rise of large-scale blockchain system, PBFT consensus hinders its further application due to poor scalability and low efficiency.

Many improved consensus protocols have been developed based on PBFT by adjusting its communication structure. HoneyBadgerBFT [14] has been proposed, which can reach a consensus in asynchronous network. However, it causes a large communication complexity. HotStuff organizes process in a star-topology to reduce message complexity [16]. And it introduces the concept of pipelining optimization that can overlap different rounds of different consensus instances. Unfortunately, the network is sensitive to reliability of the leader. A scalable Byzantine fault tolerant method (SBFT) [17] has been proposed to reduce communication in collectors and threshold signatures [18]. A scalable multi-layer PBFT consensus was introduced in [19] by grouping nodes into different layers and limiting the communication within the group to degrade the communication complexity. An optimized consensus was introduced in [20], which designs a hierarchical structure and combines PBFT and RAFT [21] together to increase scalability. The innovation of hierarchical structure [20, 21] can be applied to improve the scalability of blockchain. However, the problem is that the more layers there are, the less fault-tolerant it is for Byzantine nodes. The safety performance would decrease to some extent.

Some methods have been proposed by introducing the knowledge of cryptography to ensure both security and scalability of blockchain. PBFT with short-lived signature was presented in [22], which utilizes short-length cryptographic key to verify messages in consensus process. Extensible-PBFT [23] uses verifiable random function to elect consensus nodes for reducing communication overhead. A random selection was developed to cooperate with the voting mechanism in [24], which can be applied to reduce the number of nodes participating in the consensus process. This scheme is characterized by unpredictability and impartiality. A non-interactive cryptographic aggregation scheme was introduced in [25]. The leader can verify and aggregate the result from other nodes to reduce the frequency of communication. A high fault tolerance consensus called as NBFT was proposed in [26]. The original PBFT, where all nodes broadcast to each other, is changed to the consensus within the group, which is very similar to the hierarchical structure. In order to increase the security, the consistent hash strategy has been introduced to select and group nodes. Each node has uncertainty about the consensus role in advance, achieving the purpose of restraining the evil nodes. However, these methods as mentioned above [22–26] take time for calculation due to the addition of encryption algorithms.

The combination of both reputation model and consensus mechanism may maintain the trust state of nodes in blockchain network [27]. Some reputation-based consensus methods for further improving the performance of PBFT have been developed. An optimized PBFT based on EigenTrust model was proposed in [28]. It evaluates the trust of nodes through transactions between nodes to reduce the number of consensus nodes and communication complexity. A reputation strategy has been proposed in [29] by using an incentive mechanism to influence both consensus and transactional behaviors of nodes. A reputation based blockchain consensus protocol has been introduced in [30], which utilizes reputation information and timestamp to ensure the reliability of nodes and enhance the security of blockchain by a penalty mechanism. An unsupervised online learning scheme has been developed in [31] for reputation calculation to build the consensus committee. A mixed Byzantine fault tolerance was proposed in [32]. It uses a credit mechanism and a random node selection to improve the security and fault tolerance. Credit-delegated Byzantine fault tolerance was proposed in [33]. The protocol adopts a voting rewards and punishments scheme to reduce the participation of abnormal nodes. However, there are centralization tendency and low overall efficiency in these optimization methods [28–33]. A practical Byzantine consensus based on quantitative role (QPBFT) was presented in [34], which first uses analytic hierarchy process (AHP) to quantify the reliability of node. A QoS-aware trust service global evaluation mechanism was incorporated in [35], which implements service reliability ranking by conducting dynamic evaluation according to the real-time performance of services. However, the method in [34, 35] would appear the phenomenon of reputation accumulation of partial nodes with the continuation of consensus process, which would lead to the centralized trend of system. In addition, since the complex communication structure is maintained, the performance improvement of these consensus protocols is limited to some extent.

A comprehensive reputation based PBFT (CRPBFT) method has been proposed. Some reputation values of nodes are periodically updated according to developed reward and punishment model. Some malicious nodes are disabled from destroying consensus to enhance the reliability of nodes in

the network. The static network structure in PBFT is optimized to reduce resource consumption and improve consensus efficiency. The rotation mechanism has been developed to reduce the centralization trend caused by the reputation model. Some main contributions are as follows. Firstly, the credibility of nodes in the network are quantified by comprehensive reputation model. The nodes with higher reputation are selected to establish the blockchain consensus group. Only nodes in the consensus group are eligible to participate in the consensus, which reduces the number of consensus nodes and improves the consensus efficiency. Secondly, the consensus protocol is optimized to replace the whole network broadcast structure in commit phase with a star structure to speed up the process of blockchain consensus consistency. Finally, an important contribution is that the rotation mechanism is designed to enhance the robustness of network by removing malicious nodes from the network in time. Besides, such mechanism can be applied to maintain a higher degree of decentralization by regularly replacing consensus nodes. Comparative experiments with some state-of-the-arts have indicated the developed approach with superior performance.

The rest of the paper is organized as follows. A comprehensive reputation model is described in Sect. 2. Some comprehensive reputation based PBFT details are illustrated in Sect. 3. Performance analysis is discussed in Sect. 4. Experimental results and discussions are described in Sect. 5 and followed by some conclusions in Sect. 6.

2 Comprehensive reputation model

Nodes in a network usually include normal and malicious ones. PBFT [12] does not distinguish nodes before consensus process. To distinguish between normal and malicious nodes, reputation model has been proposed in [28, 30, 35]. Since these works [28, 30, 35] only consider service or consensus, they do not perform well in evaluating node reliability. A new comprehensive reputation model is developed to evaluate the reliability of nodes more accurately. It is composed of both service and consensus reputations. The comprehensive reputation value of nodes is updated dynamically, which helps to reduce the occurrence rate of malicious nodes in the consensus process.

2.1 Service reputation

Nodes in the blockchain network could be regarded as service providers. Quality of Service (QoS) is an indicator to measure the service performance of network. When the service has low response delay and high throughput, the blockchain nodes usually have high performance and reliability.

It is assumed that the response time is taken as the service evaluation index, which ranges from 0 to 1. The service reputation value of the nodes would be evaluation periodically. The average response delay of all services provided by node i is calculated in a cycle. The average response delay is calculated as follows:

$$\overline{T} = \frac{1}{K}\sum_{j=1}^{K}\overline{T}_j \tag{1}$$

where $\overline{T}_j$ denotes the average response time for node i serving node j in a cycle. K represents the total number of services for node i.

Different response times reflect the different performance of the node. The shorter the average response time is, the better the performance and reliability of the node are. Both reward and penalty strategies that are different from previous ones have been developed. Firstly, when the node with high service reputation continue to provide better services, the growth rate of reputation value would be smaller and smaller. It can be applied to avoid the rapid growth of the reputation. Accordingly, when it provides poor service, the service reputation decreases rapidly. Secondly, the service reputation of the node is easy to calculate, which reduces the computing cost.

The node that provides normal services would be rewarded according to the node performance in a cycle as:

$$R_r = sin\frac{(1-\overline{T})\pi}{2} \tag{2}$$

After this cycle, the service reputation value of the node would be updated as following:

$$S^n = S^{n-1} + \alpha R_r(1 - S^{n-1}) \tag{3}$$

where S^n represents the service reputation value of the node in the n^{th} cycle. The initial service reputation value $S^0 = 0.5$. α is a parameter that controls the growth rate of the service reputation, which is set to 0.05 in the experiment.

If the node fails temporarily or provides poor service multiple times in a cycle, which may lead to a long average response time. In this case, the node would be penalized as following:

$$R_p = sin\frac{(-\overline{T})\pi}{2} \tag{4}$$

After this cycle, the service reputation value of the node would be updated as:

$$S^n = S^{n-1} + \alpha R_p S^{n-1} \tag{5}$$

2.2 Consensus reputation

The reliability of nodes could be guaranteed in the service stage due to existence of service reputation. However, some nodes could not be guaranteed to be absolutely reliable in the consensus stage when they are selected as consensus nodes to participate in consensus. The consensus reputation is proposed to measure the contribution of consensus nodes in the consensus stage. The consensus reputation value of the nodes would be settled periodically when the consensus cycle is defined, as shown in (7).

$$C_{cur} = \frac{1}{1 + e^{-\beta(\sum_{k=1}^{m} T_k - \sum_{k=1}^{m} F_k)}} \tag{6}$$

$$C^n = \gamma C_{cur} + (1 - \gamma) C^{n-1} \tag{7}$$

where C_{cur} is the consensus reputation value of the node i in the current cycle. m is the number of the consensus that needs to be completed in a consensus cycle. $T\,k$ indicates whether node i normally replys in the k^{th} consensus or not. If the node returns the correct information, $T\,k = 1$. Otherwise, $T\,\mathrm{k} = 0$. $F\,k$ indicates whether node i mistakenly replys in the k^{th} consensus. If so, $F\,k = 1$. Otherwise, $F\,k = 0$. β and γ are the weight parameters, which control the growth rates of the consensus reputation value, respectively. C^n indicates the consensus reputation value of the node in the n^{th} cycle. m, β, and γ are set 100, 0.05, and 0.6, respectively in the experiment.

The comprehensive reputation value of nodes can be calculated by combining both service and consensus reputations as following:

$$R = \eta S + (1 - \eta) C \tag{8}$$

where η is a weight. It would be discussed later.

3 Comprehensive reputation based PBFT

3.1 Construction of consensus group

The trust degree of nodes could be quantified by using reputation evaluation model. Those nodes with higher comprehensive trust value are more trustworthy. The consensus group would be constructed by selecting some nodes with higher trust value instead of all nodes. It could enhance the fault-tolerant rate of network by excluding the low credit nodes and reach consensus quickly by narrowing the range of nodes.

It is assumed that the number of all nodes in the network is N and the number of nodes selected into the consensus group is $d*N\,(0 < d \leq 1)$, where * is a multiplication operator. An idea is to selected the top $d*N$ nodes directly according to the size of reputation value. However, in an actual network, all nodes have only service reputation without consensus one in the initial stage. In this situation, it is difficult to ensure that $d*N > 3f + 1$, where f is the number of Byzantine nodes in the consensus group. Furthermore, even if the nodes have a high service reputation value in the initial stage, it is difficult to fully ensure its performance in the consensus process. Therefore, such method should be avoided.

A segmented screening scheme is proposed to solve this problem as mentioned above. In the initial stage, all nodes are sorted in a descending order according to service reputation value. Some top $(d+x)*N$ nodes are selected into temporary consensus group, where $0 < x \leq (1\text{-}d)$. Then it enters the adjustment stage with progress of consensus. Temporary consensus group would eliminate the nodes with poor performance to get consensus group. The comprehensive reputation value of all nodes in the temporary consensus group is recalculated after each consensus cycle. Some D consensus nodes with the worst comprehensive performance are eliminated. In order to get a final consensus group, the number of consensuses to be completed is calculated as:

$$M = \left\lceil \frac{xN}{D} \right\rceil \times m \tag{9}$$

where $\lceil\ \rceil$ is a rounding up calculation operator.

The selection of final consensus group is described as following:

Algorithm 1 Get final consensus group

Input: node set *Nodes*, comprehensive reputation model *R*, *M*, *m*, *d*, *x*, *N*, *D*
Output: final consensus group

```
Sort (Nodes, R)
For each i ∈ Nodes do
    If R_i is in the top (d+x)*N then
        Add node i into Temporary consensus group;
    Else
        Add node i into Candidate group;
    End if
End for
If consensus times < M then
    If consensus times % m == 0 then
        Sort (Temporary consensus group, R);
        Exclude D consensus nodes from Temporary consensus group
    End if
End if
Final consensus group = Temporary consensus group
Return final consensus group
```

3.2 Consensus process

Sine the communication structure of PBFT consensus [12] is complex with low system throughput, it has been optimized by some work [23, 30]. These work increases the data throughput to a certain extent. However, it is difficult to ensure the final consistency when the leader fails.

A comprehensive reputation evaluation model has been proposed as mentioned above to overcome this limit. The node with the highest reputation value is taken as the leader. Both stability and reliability of the leader can be ensured. So it can be applied to improve the network efficiency. Since the master node is highly trusted, the whole network broadcast structure can be replaced with a star one in the commit phase. The communication structure of consensus is optimized to degrade the communication overhead, which is shown in Fig. 1.

Algorithm 2 describes the framework of CRPBFT protocol.

Algorithm 2 CRPBFT protocol

```
1  The client sends a data request;
2  Primary node broadcast Pre-Prepare messages;
3  Replica node broadcasts Prepare message and counts the number of Prepare messages and denotes it as count1;
4  If count1 >= 2f then
5      Replica node sends the commit-1 message to primary node;
6  End if
7  Primary node counts commit-1 messages and denotes it as count2;
8  If count2>=2f+1 then
9      Primary node broadcast commit-2 messages;
10 End if
11 If commit-2 message is legal then
12     Commit Log;
13 End if
14 Reply client;
15 Updates the consensus reputation value according to consensus performance;
```

The detailed processes are as follows:

1. The initiator of a transaction initiates a transaction. It signs the transaction with the private key and then broadcasts it to the whole network.
2. The primary node would determine whether the message is legitimate or not. If so, the master node would generate a block and broadcast it to the whole network. The format of the message is:

$$<< Pre-Prepare, p, h, Sig(p), dig >, block > \quad (10)$$

where p is the primary node number. h is the current block height. $Sig(p)$ is the signature of the primary node p and dig is the summary of block.

3. After receiving the pre-prepare message sent by the primary node, each replica node would check the message signature and block height. If verification fails, the replica node discards the proposal message. Otherwise, the replica node would broadcast the prepare message to other consensus nodes. The message format is:

$$<< Prepare, i, h, Sig(i), dig >, block > \quad (11)$$

where $Sig(i)$ is the signature of the replica node i.

4. If the consensus node receives $2f+1$ prepare messages, the consensus would enter the commit stage. The leader node has a high probability of honesty. The replica node only needs to communicate with the leader node. The replica node sends the commit-1 message to primary node instead of all consensus nodes as:

$$<< Commit-1, i, h, Sig(i), dig >, block, true/false > \quad (12)$$

where *true/false* represents the identity of node i to determine whether the message is valid or not.

The primary node would collect the commit-1 message from different replicas and verify the legitimacy of the message. If there are $2f+1$ identical and legitimacy commit-1 message, a commit-2 messages with digital signature of the leader node would be generated and broadcasted to all replica nodes. The message format is:

$$<< Commit-2, p, h, Sig(p), dig >, block, Cert(B^h) > \quad (13)$$

where $Cert(B^h)$ is the block certificate, which contains at least $2f+1$ affirmative commit-1 messages on the block B^h.

It can be used to verify the validity of the block B^h. The replica node receives the commit-2 message before the timeout and has the opportunity to write the proposed block into the log. The block height is increased by one to enter the consensus of a new height. If the replica node does not receive the proof from the leader node during the consensus timeout, the block height remains unchanged. The next round of consensus on the proposal would continue.

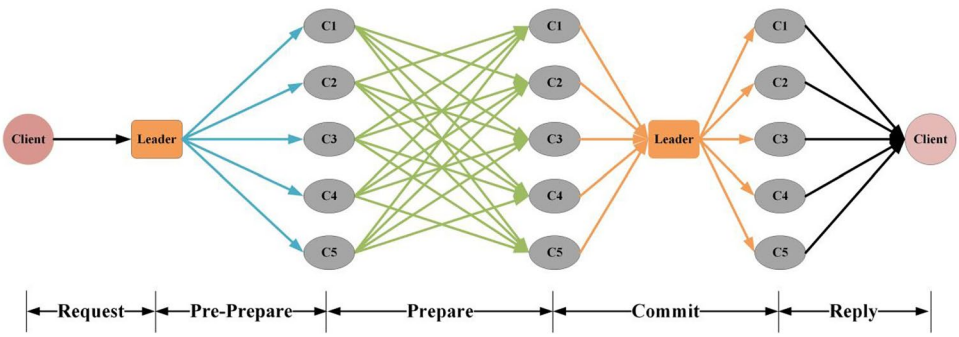

Fig. 1 An overall flowchart of CRPBFT

5. When the consensus node receives the published block, it considers this round of consensus be complete. The node updates the consensus reputation value according to consensus performance, and updates the comprehensive reputation value of the node.

3.3 Rotation mechanism

The rotation mechanism designed in CRPBFT is necessary. On the one hand, it could replace the nodes with problems in time to improve fault tolerance and efficiency of network. On the other hand, this mechanism can help to overcome the risk of network centralization by regularly replacing the consensus nodes.

The rotation mechanism is triggered by r consensus cycles. The parameter can be freely set by the user according to the actual situation. In the experiment, r is set to 10. The comprehensive reputation value of all nodes would be calculated when this mechanism is triggered. Both consensus nodes and candidate nodes are sorted according to their comprehensive reputation value, respectively. Some consensus nodes with the worst comprehensive performances would be eliminated from the consensus group. Some nodes with excellent performances in the candidate group would be promoted to the consensus group for participating in the next round of consensus. The rotation process is shown in Fig. 2.

On the premise that the comprehensive reputation evaluation model remains well, malicious nodes are less likely to be selected. However, this does not guarantee that the consensus nodes always be honest. When there is malicious behavior in the consensus stage, the credibility of malicious nodes would decrease significantly based on the reputation evaluation model. The comprehensive reputation value of the node has a high probability of ranking last among all consensus nodes. In this way, malicious nodes can be effectively identified and cleaned up in time, which can greatly increase the probability of successful consensus. The long-term and stable operation of the blockchain network can be guaranteed by this mechanism.

If the consensus nodes perform their obligations normally in the consensus process, the network has the problem of reputation accumulation without this mechanism. It would lead to the risk of network centralization. The adoption of rotation mechanism helps to overcome this disadvantage. A few consensus nodes would be replaced regularly with candidate nodes, which makes some candidate nodes have the opportunity to participate in the consensus. It has to maintain excellent transaction behavior to improve its trust when a candidate node wants to join the consensus. It can be eligible to participate in the consensus when its reputation value is high. In this case, it is helpful to encourage nodes to maintain excellent behavior habits.

The pseudocode for the rotation mechanism is described in Algorithm 3.

Algorithm 3 Process of rotation mechanism

Input: consensus cycles *r*, number of consensus nodes to be rotated *n*

```
If r == 10 then
    Sort (consensus nodes, R);
    For each i ∈ consensus nodes do
        If node i ∈ the last n of the consensus nodes then
            Exclude node i from consensus group;
            Add node i into candidate group;
        End if
    End for
    Sort (candidate nodes, R);
    For each j ∈ candidate nodes do
        If node j ∈ the top n of the candidate nodes then
            Exclude node j from candidate group;
            Add node j into consensus group;
        End if
    End for
End if
```

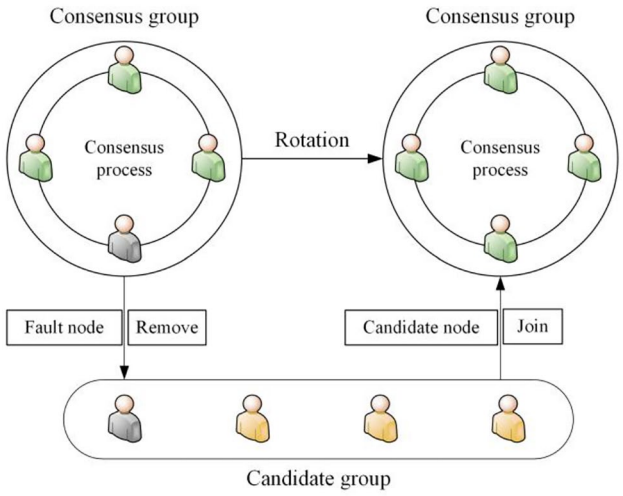

Fig. 2 The process of rotation mechanism

4 Performance analysis

4.1 Safety

This section discusses whether the developed consensus strategy can maintain the security of PBFT protocol or not. Since our improvement is at the commit stage, we mainly analyze this stage.

If the replica node is a malicious node, then it may make an incorrect response to the master node in the commit phase. Assuming that the maximum number of malicious nodes in the consensus group is f, and the normal replica node makes the correct judgment. In general case, the master node would still receive $2f+1$ correct response from different nodes and eventually reach consensus. After the consensus is completed, the behavior of malicious node is recorded due to the wrong response and replaced under the rotation mechanism.

The reputation evaluation model ensures the high reliability of the master node to a great extent. The credibility of the master node is further guaranteed after completing both pre-prepare and prepare phases. However, the primary node may have a temporary failure. If the replica node does not receive the commit-2 message from the primary node before timeout. The replica node would initiate view change and replace the primary node, which is less likely to happen. Whatever the leader node or replica one is malicious, the developed consensus strategy can be applied to ensure the security of PBFT protocol at the whole blockchain.

4.2 Fault tolerance

Assume that the total number of nodes in the blockchain network is N. Since all nodes in the network participate in consensus process, the Byzantine fault tolerance rate of PBFT [12] is (N-1)/3.

If the developed CRPBFT is applied to the network, the Byzantine fault tolerance rate would be promoted to some extent. Since the nodes in candidate group have no effect on the consensus, the fault tolerance rete can easily be deduced as (d*N-1)/3 in the consensus group when the nodes with top d*N ($0 < d \leq 1$) reputation value are selected to participate in the consensus process. In the worst case, namely all of them are Byzantine nodes. The maximum number of the Byzantine nodes in the network is:

$$(dN-1)/3 + (1-d)N = (1-\frac{2}{3}d)N - \frac{1}{3} \tag{14}$$

One can find from (14) that the fault tolerance rate of CRPBFT is superior to that of PBFT [12]. The fault tolerance rate of PBFT [12] can be considered as a special case of our developed CRPBFT as $d = 1$ in (14).

4.3 Efficiency

All nodes need to participate in the consensus process in PBFT [12]. The total communication times after reaching a consensus are:

$$C_{PBFT} = N - 1 + (N-1)(N-1) + N(N-1) = 2N^2 - 2N \tag{15}$$

The consensus process is limited within the bounds of consensus group rather than all the nodes in our developed CRPBFT. Assume that the number of consensus group is d*N. The total communication times of a consensus is as follow:

$$\begin{aligned} C_{CRPBFT} &= (dN-1) + (dN-1)(dN-1) + 2(dN-1) \\ &= d^2N^2 + dN - 2 \end{aligned} \tag{16}$$

Considering the existence of segmented screening scheme, the average communication times of a single consensus in the network is as follow:

$$\begin{aligned} \overline{C_{CRPBFT}} &= (1 - \frac{m\left\lfloor \frac{xN}{D} \right\rfloor}{h})(d^2N^2 + dN - 2) \\ &+ \frac{\sum_{k=0}^{\lfloor \frac{xN}{D} \rfloor} [(d+x)N - Dk]^2 + [(d+x)N - Dk] - 2}{h} \end{aligned} \tag{17}$$

With the increase of block height h, the average communication times of a consensus tend to be close to d^2*$N^2 + d$*N-2. It can be seen that the communication times is reduced on the whole.

5 Experimental results and discussions

5.1 Dataset and implements

To assess the performance of comprehensive reputation model in practical application, we use a publicly-available QoS dataset of real-world Web services, namely QWS Dataset [36]. It records the QoS values of 2,507 web services. Each web service has nine QoS indicators. The service response time is selected as the service reputation evaluation index of nodes in blockchain network. The abnormal values in the dataset are removed to make the service response time value range from 0 to 1. The blockchain simulation platform runs on a Linux machine equipped with an eight-core Intel i7-9200U 3.6 GHz CPU and 32 GB RAM.

5.2 Parameter analysis

The comprehensive reputation is an important index to measure the reliability of nodes, which is composed of both service and consensus reputations. The higher the comprehensive reputation value of a node is, the higher its reliability is. Under this selection mechanism, the probability of malicious nodes being selected into the consensus group is smaller. There are 50, 100, 150, 200, 250, and 300 nodes selected to participate in the consensus, of which 10% nodes were set as malicious nodes. After a cycle, the service reputation and consensus reputation of these nodes can be calculated, respectively. Some nodes with the top 90% comprehensive reputation values are selected, and the proportion of honest nodes selected can be determined. To get an optimal comprehensive reputation value, the η in (8) is changed from 0.1 to 0.9 at an interval of 0.2. Some

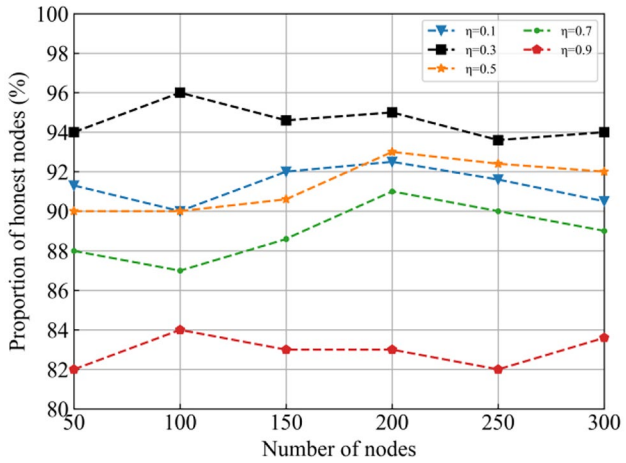

Fig. 3 The proportion of honest nodes with different η in (8)

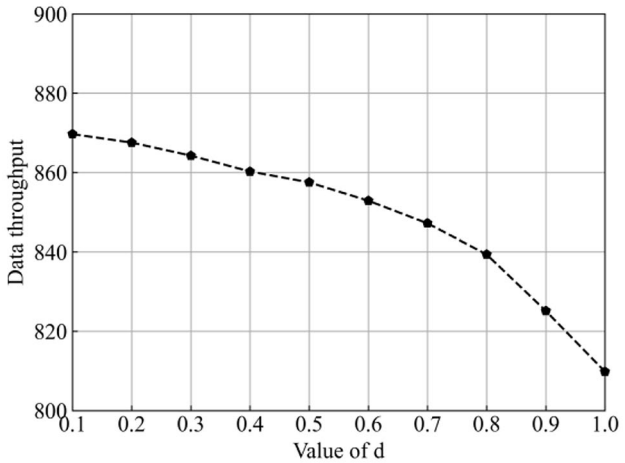

Fig. 5 Data throughput at different d

results are shown in Fig. 3. One can find from Fig. 3 that the proportion of honest nodes is the highest when η is 0.3. In the subsequent experiments, η is set as 0.3 and keep the same.

In order to get a reasonable d, the value of d is changed from 0.1 to 1 at an interval of 0.1. In the experiment, the total number of nodes was set 50, and 50*d nodes were selected to participate in the consensus process. One can find from Fig. 4 that the number of Byzantine nodes can be tolerated when the consensus group increases gradually with the increase of d. It makes the network difficult to be crashed. Meanwhile, the degree of network decentralization is higher. A higher degree of decentralization means it is more difficult to tamper with the information recorded on the block.

The number of consensus nodes would increase with the increase of d, which causes the decrease of data throughput as shown in Fig. 5. The data throughput initially decreases slowly. When d is greater than 0.8, the data throughput decreases sharply. We set d as 0.8, and kept it the same in the experiment.

5.3 Rotation mechanism analysis

To test the developed rotation mechanism, we set the number of nodes participating in consensus process as 20. There are 14 honest nodes, and 6 offline nodes, respectively. In this experiment, in order to make the comparison more intuitive, the consensus nodes participating in rotation process are set to 2. When it is set to other values, the comparison result is not obvious with the same final experimental effect. We take CRPBFT without the rotation mechanism as CRPBFT-N. Some results with the rotation mechanism or not are given in Fig. 6.

One can find from Fig. 6 that the throughput of CRPBFT-N decreases with the increase of consensus times due to

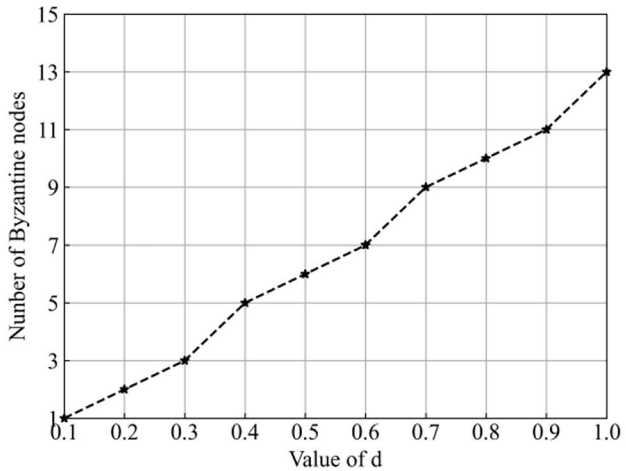

Fig. 4 Byzantine nodes number at different d

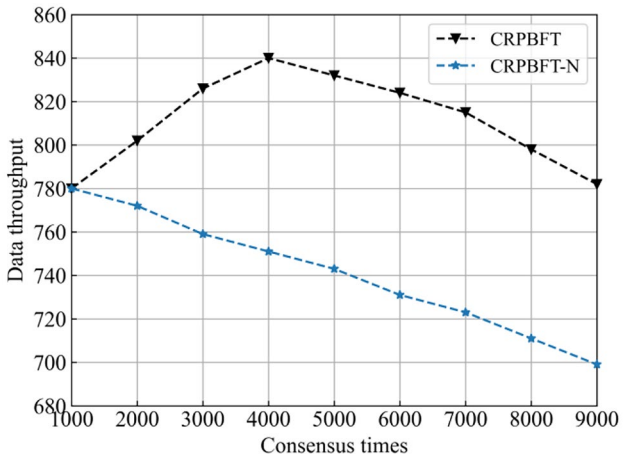

Fig. 6 Data throughput with the rotation mechanism or not

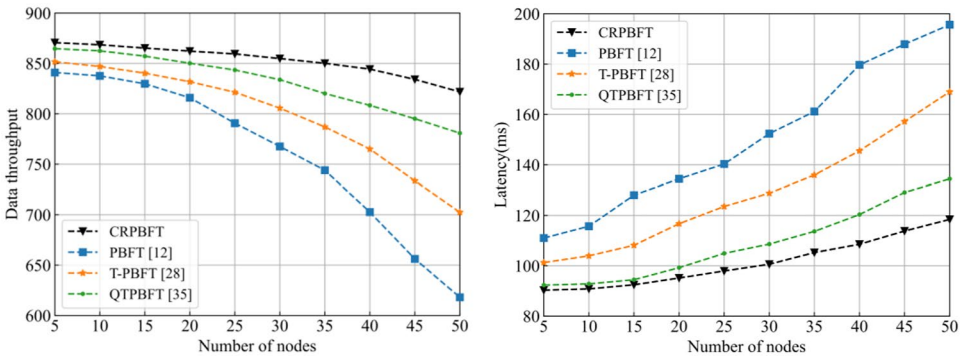

Fig. 7 Comparisons of both throughput and latency

limited computing power. The throughput of CRPBFT increases with the increase of consensus times at the early stage. The reason is that the offline nodes in the network are replaced by some honest nodes one after another when the rotation mechanism is triggered. With fewer and fewer fault nodes in the network, the throughput of CRPBFT has been improved accordingly. The experiment results demonstrate that the proposed rotation mechanism can be used to effectively improve the throughput performance.

5.4 Comparisons with some consensus protocols

Data throughput is an important indicator to measure the consensus performance. The higher the throughput is, the greater the ability to process transaction is. Transaction latency is another important indicator to measure the consensus. It represents the time difference from transaction submission to transaction confirmation, which is the benchmark for calculating the communication efficiency.

In order to further evaluate developed CRPBFT performance, some methods are selected in terms of both data throughput and transaction latency including PBFT [12], T-PBFT [28] and QTPBFT [35]. To get a fair comparison, we set the number of the nodes in the primary group as 1 for T-PBFT [28]. Some results are shown in Fig. 7.

One can find from Fig. 7 that the developed CRPBFT has the best performance among the investigated methods at both data throughput and transaction latency. Some reasons are as follows. In PBFT [12], since all nodes participate in consensus and complex broadcast communication structure, the throughput decreases greatly when the number of nodes increases. Although T-PBFT [28] narrows the consensus consortium nodes down a group of node with higher trust value, the communication structure has not changed, so the improvement effect is not obvious. QTPBFT [35] incorporates a QoS-aware trust service global evaluation mechanism that implements service reliability ranking, which ensures the reliability of consensus nodes. Then some nodes with high value would be selected to participate in the consensus process, which can effectively improve the performance of the network. However, the consensus nodes still need to communicate with each other twice in the consensus process, which is quite time-consuming. Our developed CRPBFT utilizes comprehensive reputation model to select some reliable nodes to participate in consensus process, which could effectively increase the data throughput. And the structure of network communication is simpler than previous work, which helps to reduce transaction latency. The experiment results demonstrate that the proposed CRPBFT consensus algorithms have excellent performance.

In order to get fairer results, more indicators are taken into account. Some comparison results with other consensus protocols are shown in Table 1.

It can be found from Table 1 that the proposed CRPBFT has the best performance among the investigated methods

Table 1 Comparisons with different consensus protocols

Consensus	Communication complexity	Byzantine fault tolerance	Scalability	Degree of decentralization
PBFT [12]	$O(N^2)$	$\frac{N-1}{3}$	Low	High
NBFT [26]	$O([(N\text{-}1)/m]^2)$	$\geq \frac{N-1}{3}$	High	Medium
T-PBFT [28]	$O(N^2)$	$(1-\frac{2}{3}d)N-\frac{1}{3}$	High	Low
QPBFT [34]	$O(N^2)$	$\geq \frac{N-1}{3}$	High	Medium
QTPBFT [35]	$O(d*N^2)$	$\frac{dN-1}{3}$	High	High
CRPBFT	$\mathbf{O(d^2*N^2)}$	$\mathbf{(1-\frac{2}{3}d)N-\frac{1}{3}}$	**High**	**High**

[12, 26, 28, 34, 35]. Some reasons are as follows. PBFT [12] is inefficient in the network composed of large-scale nodes due to the communication complexity of $O(N^2)$. NBFT [26] improves the performance of scalability and communication complexity though, the degree of decentralization has decreased. T-PBFT [28] reduces the degree of decentralization to some extent. QPBFT [34] selects a novel method (AHP) to quantify the reliability of nodes, which ensures the reliability of consensus nodes. Unfortunately, it still maintains high communication complexity. QTPBFT [35] has high scalability and decentralization. However, since its original complex communication structure is maintained, the communication complexity is still high. Our proposed protocol can reduce the communication complexity with high degree of decentralization. Besides the developed protocol could tolerate more Byzantine nodes.

6 Conclusions

A new consensus protocol has been proposed to optimize PBFT consensus protocol in this paper. A comprehensive reputation model is proposed to evaluate the credibility of each node from service behavior and consensus process. Some nodes with higher trust value are selected to participate in the consensus, which helps to reduce the possibility of consensus failure caused by the existence of malicious nodes. The consensus process is improved to degrade the network communication overhead by optimizing the communication structure. The rotation mechanism is designed to enhance the robustness and dynamic of the consensus network by replacing the consensus nodes regularly. Comparative experiments with state-of-the-arts have indicated the superior performance of the developed CRPBFT.

As future research, we would continue to optimize the communication structure in the consensus protocol. In the *prepare* stage of CRPBFT, nodes in the network still need to communicate with each other, which is time-consuming. A new solution would be go into to further speed up the consensus process of the blockchain network while ensuring the security of the network. At the same time, we intend to explore the possibility of applying blockchain to smart grid, which is conducive to promoting the development of smart grid.

Acknowledgements This work is supported in part by National Key R&D Program of China (Grant No. 2020YFC1523004)

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Lu Y, Huang X, Dai Y (2019) Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. IEEE Trans Industr Inf 16(6):4177–4186
2. Huang J, He D, Obaidat MS (2021) The application of the blockchain technology in voting systems: A review. ACM Comput Surv 54(3):1–28
3. Li X, Jiang P, Chen T (2020) A survey on the security of blockchain systems. Futur Gener Comput Syst 107(1):841–853
4. Buterin V (2014) A next-generation smart contract and decentralized application platform. White Paper 3(37):1–36
5. Rezaeibagha F, Mu Y (2019) Efficient micropayment of cryptocurrency from blockchains. Comput J 62(4):507–517
6. Harish AR, Liu XL, Zhong RY (2021) Log-flock: a blockchain-enabled platform for digital asset valuation and risk assessment in E-commerce logistics financing. Comput Ind Eng 151(1):1–13
7. Hakak S, Khan WZ, Gilkar GA (2020) Securing smart cities through blockchain technology: architecture, requirements, and challenges. IEEE Network 34(1):8–14
8. Tian H, He J, Ding Y (2019) Medical data management on blockchain with privacy. J Med Syst 43(2):1–6
9. Musamih A, Salah K, Jayaraman R (2021) A blockchain-based approach for drug traceability in healthcare supply chain. IEEE Access 9(1):9728–9743
10. Gramoli V (2020) From blockchain consensus back to Byzantine consensus. Futur Gener Comput Syst 107(2):760–769
11. Zheng Z, Xie S, Dai HN (2018) Blockchain challenges and opportunities: A survey. Int J Web Grid Serv 14(4):352–375
12. Castro M, Liskov B (2002) Practical Byzantine fault tolerance and proactive recovery. ACM Trans Comput Syst 20(4):398–461
13. Behl J, Distler T, Kapitza R (2014) Scalable BFT for multi-cores: actor-based decomposition and consensus-oriented parallelization. Hot Topics in System Dependability, 245–276
14. Miller A, Xia Y, Croman K (2016) The honey badger of BFT protocols. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 31–42
15. Garcia M, Neves N, Bessani A (2016) SieveQ: A layered BFT protection system for critical services. IEEE Trans Dependable Secure Comput 15(3):511–525
16. Yin M, Malkhi D, Reiter MK (2019) HotStuff: BFT consensus with linearity and responsiveness. Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, 347–356
17. Gueta GG, Abraham I, Grossman S (2018) Sbft: A scalable decentralized trust infrastructure for blockchains. Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 1804
18. Gan Y (2021) A fully adaptively secure threshold signature scheme based on dual-form signatures technology. Security and Communication Networks 2021(5):234–267
19. Li W, Feng C, Zhang L (2020) A scalable multi-layer PBFT consensus for blockchain. IEEE Trans Parallel Distrib Syst 32(5):1146–1160
20. Li Y, Qiao L, Lv Z (2021) An optimized byzantine fault tolerance algorithm for consortium blockchain. Peer-to-Peer Networking and Applications 2021(1):1–14
21. Huang D, Ma X, Zhang S (2019) Performance analysis of the raft consensus algorithm for private blockchains. IEEE Trans Syst 50(1):172–181
22. Fan X (2018) Scalable practical byzantine fault tolerance with short-lived signature schemes. Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering, 245–256

23. Li Y, Wang Z, Fan J (2019) An extensible consensus algorithm based on PBFT. 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 17–23
24. Zhan Y, Wang B, Lu R (2021) DRBFT: Delegated randomization Byzantine fault tolerance consensus protocol for blockchains. Inf Sci 559(1):8–21
25. Chander G, Deshpande P, Chakraborty S (2019) A fault resilient consensus protocol for large permissioned blockchain networks. 2019 IEEE International Conference on Blockchain and Cryptocurrency, 33–37
26. Yang J, Jia Z, Su R (2022) Improved fault-tolerant consensus based on the PBFT algorithm. IEEE Access 10(1):30274–30283
27. Li Y, Cheng J, Li H (2022) A survey of consensus mechanism based on reputation model. International Conference on Artificial Intelligence and Security, 208–221
28. Gao S, Yu T, Zhu J (2019) T-PBFT: An EigenTrust-based practical Byzantine fault tolerance consensus algorithm. China Commun 16(12):111–123
29. Sun Y, Xue R, Zhang R (2020) Rtchain: A reputation system with transaction and consensus incentives for e-commerce blockchain. ACM Trans Internet Technol 21(1):1–24
30. Yuan X, Luo F, Haider MZ (2021) Efficient Byzantine consensus mechanism based on reputation in IoT blockchain. Wirel Commun Mob Comput 2021(1):1–14
31. Bugday A, Ozsoy A, Öztaner SM (2019) Creating consensus group using online learning based reputation in blockchain networks. Pervasive Mob Comput 59(1):101–126
32. Du M, Chen Q, Ma X (2020) MBFT: A new consensus algorithm for consortium blockchain. IEEE Access 8(1):87665–87675
33. Wang Y, Cai S, Lin C (2019) Study of blockchains's consensus mechanism based on credit. IEEE Access 7(1):10224–10231
34. Zhang Z, Zhu D, Fan W (2020) Qpbft: Practical byzantine fault tolerance consensus algorithm based on quantified-role. 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, 991–997
35. Liu W, Zhang X, Feng W (2022) Optimization of PBFT algorithm based on QoS-aware trust service evaluation. Sensors 22(12):4590–4598
36. Al-Masri E, Mahmoud QH (2007) Qos-based discovery and ranking of web services[C]//2007 16th international conference on computer communications and networks. IEEE, 529–534

Jiamou Qi is now working toward the M.S. degree in communication and information system with the school of Communication and Information Engineering, Shanghai University, Shanghai, China. His research interests include security and privacy in Blockchain technology.

Yepeng Guan is currently a full professor at the College of Communication and Information Engineering in Shanghai University, China. He received the B.S. and M.S. degrees in physical geography from the Central South University, Changsha, China, in 1990, 1996, respectively, and the Ph.D. degree in geodetection and information technology from the Central South University, Changsha, China, in 2000. His research interests include Machine Learning, Cloud Computing and Blockchain.