R2-PBFT: An Improved PBFT Consensus Algorithm Based on Hash Ring and Reputation Model

LeLing Zhou[1,2,a], Kunfeng Wang[1,2,b], JianYing Chen[1,2,*]

{1225612160@qq.com[a], 921658495@qq.com[b], chenjy@swun.edu.cn*}

The Key Laboratory for Computer Systems of State Ethnic Affairs Commission, Southwest Minzu University, Chengdu, Sichuan, China[1]
School of Computer Science and Engineering, Southwest Minzu University, Chengdu, Sichuan, China[2]

Abstract. To address the issue of poor stability in consensus networks caused by arbitrary selection of main nodes in the PBFT consensus algorithm, this paper proposes a improved algorithm named R2-PBFT, which means Random and Reliable PBFT and is improved from PBFT algorithm based on hash ring and reputation model. Firstly, a dynamic reputation evaluation model was constructed to classify nodes with good reputation values into main node group. Then, the nodes within the main node group are hashed and formed a logical hash ring. Meanwhile, when a transaction is comping, it is mapped to the hash ring in the same way. Finally, the first node in the clockwise direction at the hash ring is selected as the main node. Theoretical analysis and experiments have shown that, our algorithm effectively avoids problems such as predicted attacks on the main node and power concentration, and effectively improves the reliability and stability of consensus networks.

Keywords: blockchain; Consensus algorithm; PBFT; Reputation model

1 Introduction

As the cornerstone of Bitcoin[1], blockchain achieves the goal of system decentralization through its unique data structure, P2P network architecture, and clever encryption and decryption design[2]. The consensus algorithm is mainly used to solve the problem of how honest nodes reach consensus in distributed and untrusted network environments[3-4]. The core of consensus is two parts: "selecting the owner" and "accounting", which can be divided into four steps[5]: selecting the owner, building blocks, verifying, and linking up. Currently, mainstream consensus algorithms include PoW[6], PoS[7], DPoS[8], Raft[9], PBFT[10], and so on. PBFT is a Byzantine fault-tolerant consensus algorithm that can ensure the correctness of consensus results in the presence of a certain number of malicious nodes in the network.

The main node selection method for PBFT consensus algorithm has been widely studied by scholars. Bai Shangwang adopts a verifiable random function to randomly select the master node from the consensus nodes, and adds a random number seed to the request sent by the client. After receiving it, each slave node verifies the legitimacy of the master node through the VRF algorithm[11]; Ren Xiuli proposed a multi master node consensus mechanism, which calculates the number of valid consensus rounds based on the historical behavior of nodes, and selects multiple master nodes based on the number of valid consensus rounds[12]; Wang Rihong

ICMEIM 2023, September 08-10, Wuhan, People's Republic of China

DOI 10.4108/eai.8-9-2023.2340194

proposed a consensus algorithm based on Time Weighted Value (TWV), which sets TWV for each node and takes the node with the largest TWV as the main node[13]; Han Zhenyang changed the election method of the main node in PBFT to active detection, and other nodes periodically detect activity towards the main node[14].

Random selection in the above literature [11] is beneficial for fair competition among nodes, but it cannot guarantee that the selected nodes are excellent. Literature [12] [13] [14] basically selects the optimal node based on some evaluation rule. Once a node continues to serve as the main node, it is easy to cause excessive concentration of consensus operation. The improvement ideas for PBFT main node selection mostly aim to randomly select or select more reliable main nodes, lacking a selection mechanism that simultaneously considers the randomness and reliability of main node selection.

In view of this, this article proposes an improved PBFT algorithm based on hash rings and reputation models—R^2-PBFT (Random and Reliable PBFT). Introduce a dynamic reputation evaluation mechanism for nodes, and divide the main node group based on reputation threshold to ensure the reliability of the selection of main nodes; Drawing on the idea of consistent hashing, the hash value of consensus requests is mapped to a logical hash ring composed of a group of master nodes to achieve random selection of master nodes.

2 Design of R2-PBFT consensus algorithm

2.1 Algorithm idea

In order to improve the reliability of selecting master nodes, the R^2-PBFT algorithm adds reputation value attributes to all nodes, constructs a dynamic reputation evaluation model, dynamically updates the reputation value of each node, and filters out master node groups based on reputation thresholds. Each master node comes from a master node group with good reputation values.

To solve the problems of predicted attacks on the main node and excessive concentration of packaging power when continuously serving as the main node, the R^2-PBFT algorithm utilizes the idea of consistent hashing to hash the attributes of each node, forming a logical hash ring for the nodes in the main node group. During consensus trading, transactions are also hashed, mapped to the hash ring, and the first node is searched clockwise as the current round of main node. Due to the unpredictability of trading tasks, the selection of the main node is truly random.

2.2 Network structure design

The schematic diagram of the R^2-PBFT network structure is shown in Figure 1. R^2-PBFT introduces computing nodes, which are operated by regulatory agencies with administrative influence and do not participate in consensus. They are responsible for summarizing the local behavioral reputation values and self recommended reputation values of each node in the CollectCredit stage, calculating the global reputation values and reputation thresholds of each node in the BroadcastCreditG stage, and broadcasting them to all nodes.

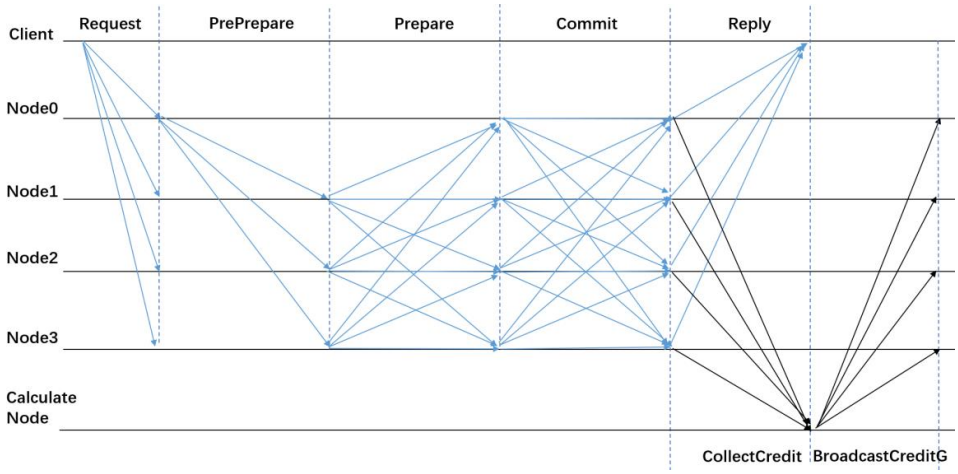

Fig. 1. R2-PBFT network structure diagram

2.3 Dynamic reputation evaluation mechanism

In order to reduce the probability of malicious nodes serving as the main node, R^2-PBFT introduces the reputation value attribute for nodes and establishes a dynamic reputation evaluation model for nodes. The reputation value of nodes is closely related to their own performance and historical behavior. The defined node behavior is shown in Table 1.

Table 1. Node behavior description

type	illustrate
Normal behavior	Able to correctly complete message verification, provide signatures, assemble data, send messages, maintain data, and other operations within a limited time frame
Malicious behavior	Malicious behavior such as tampering and replaying of messages can be considered as Byzantine nodes
Silent behavior	Not responding to messages that should be responded to, there are two situations: malicious denial of service and passive denial of service

Definition 1: Local behavior reputation value. The local behavior reputation value is the evaluation of a node on the evaluated node, mainly considering the interaction situation of the current round of consensus. For example $\mathrm{Credit\_P(X)}_{i\to j}$, it represents the local behavior evaluation of node i on node j. The local behavior reputation value of node jis determined by node i based on the interaction behavior, behavioral roles, consensus efficiency, and participation of node j. It is expressed in formula 1.

$$Credit\_P(x)_{i\to j} = \alpha \times A_{i\to j} + \beta \times B_{i\to j} + \gamma \times C_{i\to j} \tag{1}$$

Among them,$A_{i\to j}$ represents the current behavior factor that node i evaluates node j, which is determined by the current role coefficient and behavior content of node j from the perspective of node i (Definition 2);$B_{i\to j}$ represents the consensus efficiency factor of node j when node i

interacts with node j (Definition 3); $C_{i\to j}$ indicates the participation of node j in the consensus of node i in this round (definition 4);α, β,γ corresponding coefficients.

Definition 2: Current behavior factor. Due to the fact that there are two types of roles in the PBFT algorithm: the master node and the slave node, which have different impacts when different roles commit crimes. Therefore, the role coefficients are set in the current behavior factor to provide reasonable rewards and punishments for the behavior of different roles. In the consensus process, there are three stages that require node interaction: Prepare, Prepare, and Commit. The current behavior factor calculation method is extracted as Formula 2.

$$A_{i\to j} = \frac{r_1 \times Act_1 + r_2 \times Act_2 + r_3 \times Act_3}{3} \tag{2}$$

r_1, r_2, r_3 respectively represent the role coefficients of the three stages (master node role and slave node role),Act_1, Act_2 and Act_3 respectively represent the node behavior content of the three stages (normal behavior, evil behavior, and silent behavior).

Definition 3: Consensus efficiency factor. Consensus efficiency is an important indicator for evaluating the level of participation of nodes in consensus. Based on the three stages of the consensus process, the ratio of the interaction time $time_{(p)i\to j}$ between nodes i and j in the p-th stage to the total time $time_{(p)i}$ spent by node i in the p-th stage is calculated, and the average is obtained as the independent variable input. The dependent variable obtained $B_{i\to j}$ is the consensus efficiency factor, which is expressed in formula 3.

$$B_{i\to j} = exp(-\frac{\sum_{p=1}^{k} \frac{time_{(p)i\to j}}{time_{(p)i}}}{k}) \qquad (k=3) \tag{3}$$

Definition 4: Node consensus participation. The participation index is used to determine the contribution of nodes in the consensus process, and is the basis for selecting persistent silent nodes, expressed in formula 4.

$$C_{i\to j} = \frac{Interaction\_real_{i\to j}}{Interaction\_should_{i\to j}} \tag{4}$$

$Interaction\_real_{i\to j}$ represents the actual number of interactions between node j and node i in this round of consensus, and $Interaction\_should_{i\to j}$ represents the number of interactions that node j should actively have with i.

Definition 5: Self recommended reputation value. Based on the node's own performance configuration and hardware situation, reputation recommendations are made to the node, and the evaluation model formula is Formula 5.

$$Credit\_S(x)_j = \delta \times exp(-\frac{D1_j + D2_j + D3_j}{3}) \tag{5}$$

Credit_S(x)$_j$ represents the recommended reputation value of node j in round x, where, and respectively represent the CPU usage, bandwidth usage, memory usage, etc. of node j, and δ are their coefficients.

Definition 6: Global reputation value. The global reputation value is calculated comprehensively by the local behavior reputation value of each node towards the evaluated node, the recommended reputation value of the evaluated node, the historical reputation value of the evaluated node, and the time offset. The global reputation value of node j in the x-th round is represented as Credit_G(x)$_j$, and the global reputation model formula is Formula 6.

$$Credit\_Gt(x)_j = \frac{\sum_{i=1}^{n} Credit\_P(x)_{i\to j}}{n} + Credit\_S(x)_j$$

$$Credit\_G(x)_j = \begin{cases} Credit\_Gt(x)_j & (x=1) \\ 0.6\times Credit\_Gt(x)_j + 0.4\times Credit\_G(x-1)_j & (x=2) \\ 0.5\times Credit\_Gt(x)_j + 0.3\times Credit\_G(x-1)_j + 0.2\times Credit\_G(x-2)_j & (x=3) \\ \frac{w}{10}\times Credit\_Gt(x)_j + \sum_{m=1}^{w}(\frac{w-m}{10}\times Credit\_G(x-m)_j) & (x\geq 4) \end{cases} \quad (6)$$

w is a sliding evaluation window with a size set to 4 and m is a time offset, $(w-m)/10$ as the historical reputation value coefficient within the window. This makes the global evaluation of node j in the x-th round depend on the historical reputation value during the window period. The larger the window, the greater the time offset. The longer the distance from the current round, the smaller the value, and the smaller the weight of the historical reputation score in the x-m round in the current global score; In addition to the historical reputation value during the window period, the global reputation value of the current round is also determined by the average of the local reputation values of the current round and the recommended reputation value of oneself.

Definition 7: Reputation threshold. The reputation threshold for the x-th round of the network Credit_T(x)$_j$ is calculated using formula 7 based on the global reputation values of all nodes in the network.

$$Max\_Credit\_G = MAX\{Credit\_G(x)_i \mid i \in 1,2,...,n\}$$

$$Min\_Credit\_G = MIN\{Credit\_G(x)_j \mid j \in 1,2,...,n\}$$

$$Credit\_T_{(x)} = \frac{n-1}{3n}\times(Max\_Credit\_G - Min\_Credit\_G) + Min\_Credit\_G \quad (7)$$

N is the number of consensus nodes in the network, Max_Credit_G is the node with the highest global reputation value, and Min_Credit_G is the node with the lowest global reputation value. The purpose of setting a reputation threshold is to classify consensus nodes in the network and filter out the main node groups that meet the reputation threshold.

3 Conclusion

A R2-PBFT consensus algorithm based on hash rings and reputation models is proposed to address the issues of insufficient reliability and security in selecting the main node of the PBFT consensus algorithm. A dynamic reputation evaluation mechanism is introduced to construct a global reputation value calculation model that is determined by local behavior reputation evaluation and self recommendation reputation evaluation, and is associated with the node's historical reputation, ensuring the reliability of the main node selection. In the future, we will consider layering the consensus network, dividing it into consensus groups and accounting groups, reducing the number of nodes participating in the consensus network, and conducting research on improving consensus efficiency.

Acknowledgments. This research was funded by the Special Fund for Basic Research Business Expenses of Central Universities of Southwest University for Nationalities (Grant No. 2023NYXXS049) and Sichuan Science and Technology Program (Grant No. 2023YFN0026).

References

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. Decentralized business review, 2008: 21260.

[2] Yuan Yong, Wang Feiyue. Development Status and Prospects of Blockchain Technology [J]. Journal of Automation, 2016,42 (04): 481-494. DOI: 10.16383/j.aas.2016.c160158

[3] Wang H, Qin H, Zhao M, et al. Blockchain-based fair payment smart contract for public cloud storage auditing[J]. Information Sciences, 2020, 519: 348-362.

[4] Liu Yizhong, Liu Jianwei, Zhang Zongyang, et al Overview of Blockchain Consensus Mechanism Research [J] Journal of Cryptography, 2019, 6 (4): 395-432

[5] Yuan Yong, Ni Xiaochun, Zeng Shuai, Wang Feiyue. The Development Status and Prospects of Blockchain Consensus Algorithms [J]. Journal of Automation, 2018,44 (11): 2011-2022. DOI: 10.16383/j.aas.2018.c180268

[6] Jakobsson M, Juels A. Proofs of work and bread pudding protocols[C]//Secure Information Networks:Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999, Leuven, Belgium. Springer US, 1999: 258-272.

[7] King S, Nadal S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake[J]. self-published paper, August, 2012, 19(1).

[8] BitShares. Delegated proof of stake [Online], available: https://how.bitshares.works/en/master/technology/dpos.html

[9] Ongaro D, Ousterhout J. In search of an understandable consensus algorithm[C]//2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14). 2014: 305-319.

[10] Castro M, Liskov B. Practical byzantine fault tolerance[C]//OsDI. 1999, 99(1999): 173-186.

[11] Bai Shangwang, Ma Xiaoqian, Gao Gaimei, Liu Chunxia, Dang Weichao. A Byzantine fault-tolerant consensus algorithm based on verifiable random functions and BLS signatures [J]. Journal of Guangxi Normal University (Natural Science Edition), 2022,40 (03): 194-2011. DOI: 10.16088/j.issn.1001-6600.2021071002

[12] Ren Xiuli, Zhang Lei. An improved multi master node consensus mechanism based on practical Byzantine fault tolerance [J]. Computer Applications, 2022,42 (05): 1500-1507

[13] Wang Rihong, Yuan Shanshan, Xu Quanqing, An Liangyu. Research on consensus algorithms based on time weight values [J]. Computer Application Research, 2021,38 (11): 3243-3248. DOI: 10.19734/j.issn.1001-3695.2021-03.0097
[14] Han Zhenyang, Gong Ningsheng, Ren Jiamin. Improvement of a practical Byzantine fault-tolerant algorithm for blockchain [J]. Computer Application and Software, 2020,37 (02): 226-233+294