Trust-PBFT: a PeerTrust-based Practical Byzantine Consensus Algorithm

Wei Tong, Xuewen Dong and Jiawei Zheng

September 25, 2019

Trust-PBFT: a PeerTrust-based Practical Byzantine Consensus Algorithm

Wei Tong[12], Xuewen Dong[13], Jiawei Zheng[13]
[1]Shaanxi Key Laboratory of Network and System Security, Xidian University, Xi'an, China
[2]School of Cyber Engineering, Xidian University, Xi'an, China
[3]School of Computer Science and Technology, Xidian University, Xi'an, China
Email: wtong@stu.xidian.edu.cn, xwdong@xidian.edu.cn, jwzheng@stu.xidian.edu.cn

***Abstract*—Blockchain is becoming increasingly popular, while the performance problems have confused its implementation, especially the limited size of the distributed network. In this paper, we propose Trust-PBFT, a combination of PeerTrust P2P trust calculation model and PBFT consensus algorithm. We first introduce PeerTrust to evaluate the trustworthy of the nodes that qualify as participants of PBFT, which replaces the original that all nodes participate in, so that the size of the distributed network can be expanded arbitrarily. Besides, in order to simplify the structure of block storage, particularly the storage of the feedbacks of transaction used to calculate trust value of nodes, we design a tri-chain architecture to store accounts, transactions and feedbacks in three blockchains, respectively. Moreover, due to the introduction of PeerTrust, a weak benefit is to improve the fault tolerance performance efficiently. Finally, we design simulation experiments to evaluate the fault tolerance performance and scalibility of Trust-PBFT.**

***Index Terms*—Consensus Algorithm, P2P Trust Calculation Model, Blockchain, PBFT, PeerTrust**

I. Introduction

Blockchain has become one of the research hotspots, and the number of commercial platforms and academic literatures [1]–[4] about it is increasing, such as Bitcoin [1], Ethereum [2], Hyperledger Fabric [3] and so on. Due to these features, blockchain is widely used for financial transactions, copyright protection, product traceablity and access control. A blockchain, in its essence, is a traceablity and tamper-proof ledger, periodically recording the common consensus reached by multiple distributed nodes. Blockchain is also a technology set, including Cryptography, Distributed Storage, P2P Communication, Consensus Algorithm and Smart Contract.

As a core technology for blockchain, consensus algorithm is used to order received transactions, simulate the execution of these transactions and finally reach a common state within a distributed network of consensus participant nodes, locally. There are various kinds of consensus algorithms [1], [4]–[8] in existing blockchain platforms, like PoW (Proof of Work) [1] for Bitcoin and Ehtereum 1.0, PoS (Proof of Stake) [4] for Ethereum 2.0, PBFT (Practical Byzantine Fault Tolerance) [5] for Fabric v0.6 and KAFKA [6] for Fabric v1.x, etc. However, these algorithms have many problems to restric the further development and application of blockchain, for example PoW consumes lots of computing power and PBFT limits the size of the distributed network.

The basic technology in blockchain network layer is P2P communication which leads to untrust between distributed nodes that requires consensus algorithm to solve, while traditional PBFT consensus algorithm limits the size of the distributed network. Emerging in the first decade of the 21st century, P2P trust calculation model [9]–[12] initially was proposed to deal with the above untrust problem and pick up high-trust nodes to communicate directly. Nowadays, because of a good deal of prior recorded transactions for trust value calculation for each node, P2P trust calculation model can be seen as a catalyst to address scale constraints for blockchain with consensus algorithms.

In this paper, we propose Trust-PBFT, a PeerTrust-based practical byzantine consensus algorithm which can elect the right number of participants from a large-scale blockchain distributed network to execute the traditional PBFT consensus algorithm. Combined with the characteristics of the recordings on the blockchain, we design a tri-chain blockchain architecture to record accounts transactions, general transactions and feedback value transactions, respectively. Besides, we also modify the original PeerTrust P2P trust calculation model in order to adapt to the format of the transactions in the blockchain. Finally, we conduct simulation experiments to evaluate the fault tolerance performance of our consensus algorithm and the experiment results show that the fault tolerance performance with our algorithm is more efficient than the traditional one and another benefit, in addition, is the increased size of the blockchain distributed network discretionarily.

The remaining of this paper is organized as follows. Section II outlines the related work including blockchain, PBFT and PeerTrust. In Section III, we introduce the tri-chain architecture of blockchain and the feedback-chain-based trust model. We present our consensus Trust-PBFT from two perspectives: function modules of each node and workflow in Section IV, and design simulation experiments to evaluate the fault tolerance performance and scalibility of our consensus algorithm in Section V. Section VI concludes this paper and discusses future works.

II. Related Work

A. Blockchain

With the rise of Bitcoin in 2008, blockchain has entered the field of vision of researchers and gradually become a research hotspot in the industry. The number of cryptocurrencies [1], [2], [13], [14] based on blockchain technology is increasingly growing, such as BTC [1], ETH [2], EOS [13] and HT [14], etc. From the perspective of access mechanism, the above blockchain products all belong to permissionless blockchain, that is, any node can freely produce, verify and order transactions in the blockchain network. The other is permissioned blockchain, which allows only certain nodes to join the network and participate in transaction operations.

Overall, the final record of a transaction on the blockchain is roughly divided into three steps: transaction produced and broadcast by a node, transaction verified, ordered and simulated executed by consensus algorithm participants node locally, and a batch of transaction packaged and recorded on the blockchain. Therefore, consensus algorithm in the second step is the key technology for blockchain, which is the focus of our paper.

B. Consensus Algorithm

At the end of the last century, a variety of consensus algorithms had been proposed in the literature. After 2008, in order to address the common consensus problem within the blockchain distributed network of nodes, various kinds of consensus algorithms was applied and optimized. For example, Bitcoin, the most mature application, and Ehtereum 1.0 apply PoW, whereas PoW consumes lots of computing power to mine; Ehtereum 2.0, the most popular permissionless blockchain platform, applies the combination of PoW and PoS which also have to consume computing power to solve difficulties. Therefore, the consensus algorithms for permissionless blockchains costs time and computing power to resist Sybil Attack [15], while that for permissioned blockchains have an account management module that can apply fault-tolerant-based algorithms, such as PBFT and KAFKA. For instance, Fabric v0.6, the earliest open source permissioned blockchain platform, applies PBFT and KAFKA for production environment, while Fabric v1.x apply only KAFKA.

PBFT consensus algorithm is proposed to deal with byzantine generals problem in an entirely untrust environment and the original PBFT consists of three steps: pre-prepare, prepare and commit, as is shown in Fig. 1. Analyzing the workflow in Fig. 1, we can draw the following conclusion:

- PBFT allows only $1/3$ consensus algorithm participants do evils, otherwise, the common consensus cannot be reached.
- PBFT requires all honest consensus algorithm participants participate communication, which puts a lot of pressure on network and leads to limited network size, generally less 20 nodes [16].

C. P2P Trust Calculation Model

In the first decade of this century, a variety of P2P trust calculation models had been proposed in the literature, a portion of which was based on the trust value calculation method. Yu et al. proposed a trust calculation model based on Evidence Theory, and the evaluation that a node is given to the target node is expressed as an evidence of its support [9]. Wang et al. proposed a Bayesian Network-based trust calculation model that computes nodes' trust value by statistically updating the beta probability density function [10]. Yamamoto et al. proposed a distributed trust calculation model PageRank, which calculates the trust value of nodes using PageRank algorithm in a distributed manner [11].

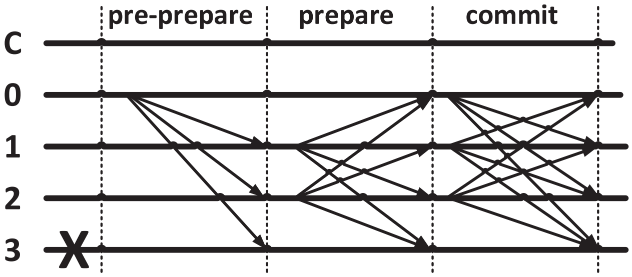

Fig. 1: Workflow of PBFT consensus algorithm.

PeerTrust, proposed by Xiong et al., utilizes multiple parameters to adjust automatically the trust value of nodes with time to elect finally the high-trust node to communicate [12]. PeerTrust defines the trust value of node u by $T(u)$ in (1).

$$T(u) = \alpha * \sum_{i=1}^{I(u)} S(u,i) * Cr(p(u,i)) * TF(u,i) + \beta * CF(u), \tag{1}$$

where α and β denote the weight factors and $S(u,i)$, $Cr(p(u,i))$, $TF(u,i)$ and $CF(u)$ denote the multiple parameters about a transaction and the network environment.

III. Architecture and Model

In this section, we introduce the architecture of the tri-chain blockchain and the functions of each chain and identify four important parameters for evaluating the trust value of a node in our trust model.

A. Tri-chain Blockchain Architecture

The common blockchain maintain only one blockchain within a distributed network of nodes and record all kinds of information on it, such as transactions, accounts, feedbacks, smart contracts and so on, which results in extremely complexity. On the contrary, we propose a tri-chain blockchain architecture that records account information, transaction information and feedbacks information of transactions, repectively, as is shown in Fig. 2.

- **Transaction Blockchain (TBC)**: TBC increasingly records transaction information, such as transfer money, reserve a technological achievement and subscribe to a service, etc, not account information and feedbacks information of transactions. Another function of this chain is that the format of transaction information is encrypted by Hash256 and Merkle Tree, which ensures that the realistic information is not widely available by the nodes in blockchain distributed network. Moverover, the

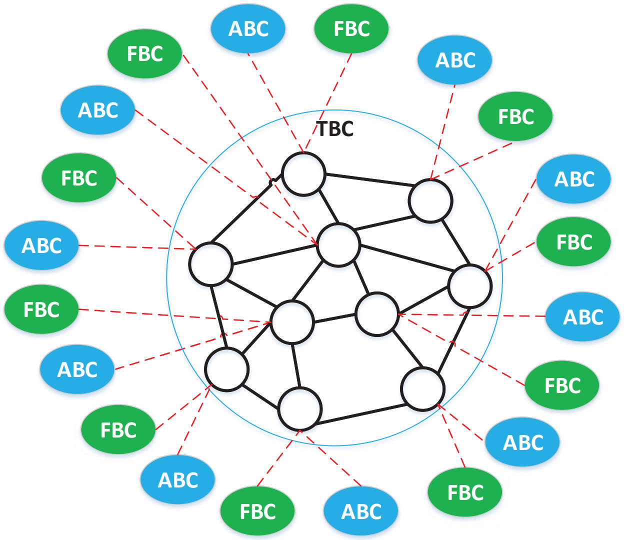

Fig. 2: The architecture of tri-chain blockchain.

way to store transactions on a separate chain is simplify query complecity and facilitate multiple chains extension. Actually, a general transaction consists of various value, such as the publisher's account address, subscriber's account address, publisher's signature, transaction amount, transaction fee, smart contract's address, timestamp and so on.

- **Account Blockchain (ABC)**: ABC records only account information. When a new node join into the blockchain network, especially the permissioned blockchain, Certification Authority and Membership Service Provider generally assign a certification, a key pair and an account address to the node. The key pair is used to sign details as producing a transaction or a feedback and verify the signature of the details as receiving a transaction or a feedback. And the account address represents the real-world nodes that transact in blockchain network to ensure anonymity and privacy. In addition, this chain is also responsible for establishing the link of cross-chain communication. Actually, an account transaction contains basic information, e.g., the account public key, account address and organization the account belongs to, etc.
- **Feedback Blockchain (FBC)**: FBC records feedback information of transactions used for providing a portion of evidence for trust value calculation of nodes and the trust value of nodes calculated by the equation in the III-B according to the important parameters introduced in the III-B. Unlike the transaction information, feedback information entirely depends on historical transactions, that is, feedback can only be received after the closing of the transaction. Besides, feedback information often comes in numercal form and cannot be changed once issued, which makes this chain simple and reliable. Actually, a feedback transaction includes various information, like the publisher's account address, publisher's signature, transaction hash, feedback value and timestamp.

Overall, tri-chain blockchain architecture not only simplifies the complex recordings, but also provides the necessary support for Trust-PBFT.

B. *Feedback-chain-based Trust Model*

In Trust-PBFT, a node's trust value is dedined by feedbacks of the nodes that it receive in publishing transactions to other nodes in the past. In our trust model, there are four important parameters we identify for such evaluation:

1) the feedback score a node receives from other nodes,
2) the number of feedback during the recent block-time,
3) the trust value of the node who issued a certain feedback, and
4) the transaction inner factor, such as value, fee and other differences.

We explain these parameters in the tech services scenario.

- **Feedback in terms of satisfaction.** Trust value-based systems rely on feedback to evaluate a node. Feedback in terms of satisfaction a node received after a transaction over reflects how well this node publish the transaction.
- **Number of feedback.** Because of the features of blockchain, prior feedback transactions are usually recorded into FBC with block-time. Therefore, the number of feedback during the recent block-time affects the results of trust value of nodes updates.
- **Trust value of feedback node.** The feedback node u received from node v after a transaction over is simply a result regarding how satisfied v felt about the quality of the transaction published by u. However, a node may make a false feedback to other nodes' transactions due to malicious motives or a node often make a low score result to all transactions due to personal habits. In this paper, we only consider the false feedback to other nodes' transactions. In Trust-PBFT trust model, we introduce the trust value of feedback node to assign higher weight to the node with higher trust value.
- **Transaction inner factor.** Transaction inner factors is another important parameter when calculating the trust value as transactions may vary widely. Various transaction inner factors, such as the value, timestamp, or fee of the transaction, can be incorporated so that the transaction for higher, more recent and higher transactions can be assigned more weight than those for other transactions.

Now that we have discussed these parameters, we dormalize thesr parameters, present a trust value calculation equation and explain it. Given a recent block-time, the time interval between the two blocks, let $N(u)$ denote the total number of feedback of transactions received by a node u from all other nodes who have done transactions with before, $n(u,i)$ denote the node who participate node u's ith transaction, $Fv(u,i)$ denote the feedback value that node u received from $n(u,i)$ in its ith transaction, $Tv(v)$ denote the trust value of feedback node v, $V(u,i)$ denote the transaction inner factor for node u's ith transaction. The trust value of node u denoted by $T(u)$, is defined in (2).

$$T(u) = \sum_{i=1}^{N(u)} Fv(u,i) * Tv(n(u,i)) * V(u,i), \qquad (2)$$

where $Tv(n(u,i))$ is actually the recent trust value $T(n(u,i))$ of node $n(u,i)$ which recorded in the last block in FBC and the different symbol is used here to distinguish $T(u)$.

We also give an example of transaction inner factor. Analyzing existing attacks targeting transactions, we choose value to reflect the weight for a certain transaction.

IV. Trust-PBFT

In this section, we illustrate the five function modules of each node to support Trust-PBFT and the workflow of Trust-PBFT with three steps, respectively.

A. Node Function Modules

Compared with the original Fabric nodes, the nodes of Trust-PBFT have higher requirements, such as publish feedback and calculate trust value of nodes. Therefore, in order to support the proposed PeerTrust-based practical byzantine consensus algorithm, each node must have the following basic functional modules, as is shown in Fig. 3.

- **Transaction Production Mudule (TPM)**: This module is mainly responsible for producing transactions which is similar to the original blockchain.
- **Consensus Module (CM)**: This module of the nodes who participate consensus algorithm (also called participants) is to verify and order the transactions producing during the recent block-time and reach the common consensus. Besides, this module of the leader node is also responsible for packing multiple transactions into one new block during the recent block-time after executing consensus algorithm.
- **Trust Value Calculation Module (TVCM)**: This module is mainly responsible for calculating the trust value of nodes based on the important parameters and the equation we introduced in the III-B. It is the core functional mudule for the proposed PeerTrust-based practical byzantine consensus algorithm. Sepecific calculation workflow is introduced in the IV-B.
- **Network Module (NM)**: This module is mainly responsible for network operations, including sending various transactions, sending verification and calculation results and sending various blocks, etc. In addition, this module is also in charge to set up routing tables for the whole distributed network, especially among the participants after each participant-time, the time interval between the two groups of participants elected by trust value of nodes.
- **Storage Module (SM)**: This module is mainly responsible for storing both the blocks recording the hash of the general transactions information and account transactions information and the blocks recording the real trust value of feedback transactions information, which is about privacy and open, reapectively.

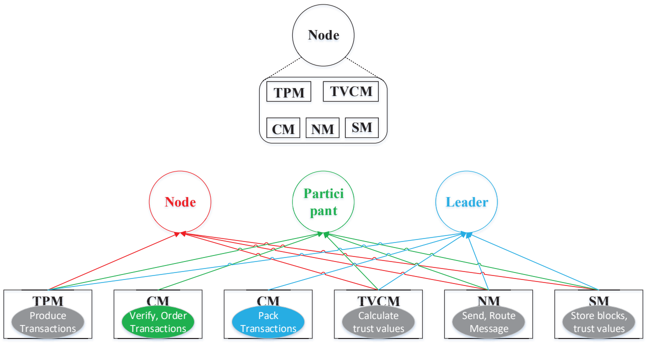

Fig. 3: Illustration of node function modules.

B. Workflow of Trust-PBFT

The core idea of the proposed Trust-PBFT is to allow a small portion of nodes with high trust value to participate in the process of executing PBFT consensus algorithm, i.e., becoming the participants. The proposed consensus algorithm is executed by the distributed network of all nodes and the workflow of Trust-PBFT includes the following three steps, which are repeated epoch by participant-time, as shown in Fig. 4.

1) **Step 1. Calculate trust values of nodes**: In this step, each node calculates the trust value of all nodes including himself locally and finally generate the common consensus block linking to FBC. At the beginning of this step, the recent participants read feedback that just linked on FBC due to these feedback has already verified by them. After that, they calculate the trust values of all nodes based on these feedback values by the proposed equation in III-B locally. Finally, they exchange their calculation reusltsand generate a new block to link it on FBC if and only if they reach a common consensus based on PBFT.
2) **Step 2. Select Participants from nodes**: This step is to select a portion of participants from all nodes to participate the process of reaching the common consensus. After linking the trust value recordings block on FBC, the trust values of all nodes can be read by all nodes in the blockchain distributed network and therefore the nodes with high trust value based on the newest block in FBC are elected as the participants during the recent participant-time, as well as the node with the highest trust value is elected as the leader during the recent participant-time.
3) **Step 3. Reach consensus among the participants**: There are two function in this step: one is to execute PBFT consensus algorithm among the selected participants; the other is to judge the end of the recent participant-time. After the selection of the participants, they reach the common consensus of the system states, i.e., generate T blocks of transactions.

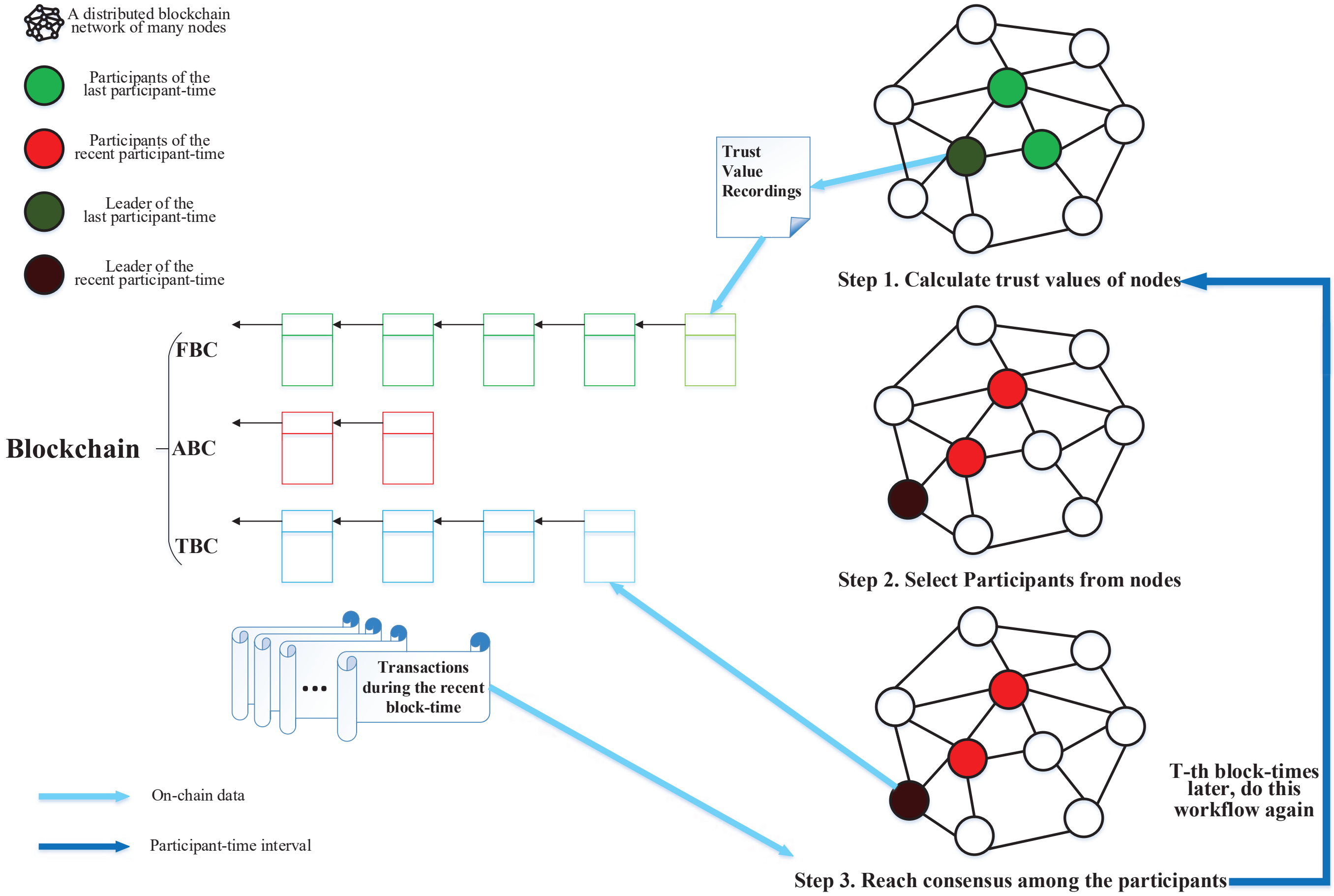
A distributed blockchain network of many nodes
Participants of the last participant-time
Participants of the recent participant-time
Leader of the last participant-time
Leader of the recent participant-time
Trust Value Recordings
Step 1. Calculate trust values of nodes
Blockchain
FBC
ABC
TBC
Step 2. Select Participants from nodes
Transactions during the recent block-time
T-th block-times later, do this workflow again
On-chain data
Participant-time interval
Step 3. Reach consensus among the participants

Fig. 4: Workflow of the PeerTrust-based Practical Byzantine Consensus Algorithm.

Overall, no matter how many nodes there are in the blockchain distributed network, there are only a finite number of nodes with high trust value can become participants and execute PBFT consensus algorithm.

V. Simulation Experiments

We perform initial simulation experiments to evaluate Trust-PBFT consensus algprithm and show the fault tolerance performance and scalibility.

A. Simulation Setup

We implemented a simulator in MABLAB 2018a and the simulation setup is shown in TABLE I, including . We consider a larger distributed network with 40 nodes and all nodes have the same initial trust value at 1. We set the number of participants in each block-time is 10 and if the same trust value is encountered beyond the top ten, select the node with more feedback as the participant. A proportion M of nodes are considerd malicious and they must do evils, such as publishing unsatisfied service transactions, providing fault feedback values, ect. Most experiments have 10 general transactions and 10 feedback transactions during each block-time and each result is the average of 20 simulation experiments.

B. Fault Tolerance Performance

One weak objective of this set of experiments is to evaluate the fault tolerance performance. As we all know in II-B, the fault tolerance for PBFT is $1/3$ and PBFT fails once the fault tolerance over $1/3$. We set M_a represent the changes of the proportion of malicious nodes acting malicious behaviors and the values of M_a are 20%, 40%, 60% and 80%. In Fig. 5, it is obvious that the fault tolerance performance of Trust-PBFT for most the values of M_a is better than that of PBFT for the M_a at 100%, while the fault tolerance performance of Trust-PBFT is similar to that of PBFT with only M_a at 80%, and the lower M_a the better fault tolerance performance.

C. Scalibility

The main aim of this paper is to scale up the size of a blockchain distributed network. This experiments start as nodes publish random transactions with each other. In Fig. 6, as the number of nodes increases, the throughput of executing two kinds of consensus algorithms present a different trend. For PBFT, the throughput is an downward trend and getting faster and faster until it dropos to 0 at 20 nodes in the distributed network. While the throughput is a different trend for Trust-PBFT, which remains roughly stable at nearly 800TPS

TABLE I: Simulation Parameters

Parameter	Description	Default
N	the number of nodes in the blockchain distributed network	40
N_p	the number of participants in each block-time	10
M	the proportion of malicious nodes in the blockchain distributed network	25%
M_a	the proportion of malicious nodes acting malicious behaviors	100%
GT	the number of general transactions during each block-time	10
FT	the number of feedback transactions during each block-time	10
FT	the number of experiments over which results are averaged	20

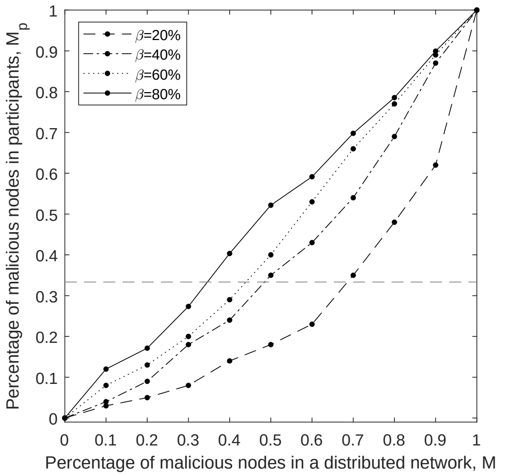

Fig. 5: Fault tolerance performance of Trust-PBFT.

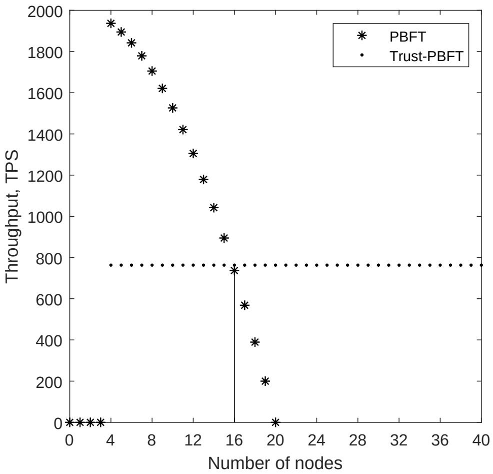

Fig. 6: Scalibility of Trust-PBFT.

(transactions per second). Fig. 6 also describes that the larger the scale, the better the scalibility performance of Trust-PBFT with more than 16 nodes in the distributed network.

VI. Conclusion

We propose Trust-PBFT, a PeerTrust-based practical byzantine consensus algorithm which can elect the right number of participants from a large-scale blockchain distributed network to execute the traditional PBFT consensus algorithm. Besides, we design a tri-chain blockchain architecture and modify the original PeerTrust P2P trust calculation model. Finally, the experiment results show that our consensus algorithm can scale up the size of the blockchain distributed network and soar dramatically the fault tolerance.

In the future, we are going to work for optimizing the transaction consensus throughput performance utilizing sharding, on-chain sacling and off-chain payment channel technologies.

Acknowledgment

This research was supported in part by the National Key Research and Development Program of China No. 2017YFB1400700, the National Natural Science Foundation of China under Grant U1536202, U1736216, 61571352 and Shaanxi Science and Technology Coordination and Innovation Project 2016KTZDGY05-07-01.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[3] "Hyperledger fabric," [Online]. Available: https://www.hyperledger.org/projects/fabric.

[4] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," *self-published paper, August*, vol. 19, 2012.

[5] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, 1999, pp. 173–186.

[6] "Fabric v1.4.0," [Online]. Available: https://github.com/hyperledger/fabric.

[7] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake," *IACR Cryptology ePrint Archive*, vol. 2014, p. 452, 2014.

[8] D. Larimer, "Delegated proof-of-stake white paper," 2014.

[9] B. Yu and M. P. Singh, "An evidential model of distributed reputation management," *adaptive agents and multi-agents systems*, pp. 294–301, 2002.

[10] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," in *Peer-to-Peer computing, 2003.(P2P 2003). Proceedings. Third international conference on*. IEEE, 2003, pp. 150–157.

[11] A. Yamamoto, D. Asahara, T. Itao, S. Tanaka, and T. Suda, "Distributed pagerank: a distributed reputation model for open peer-to-peer network," in *Applications and the Internet Workshops, 2004. SAINT 2004 Workshops. 2004 International Symposium on*. IEEE, 2004, pp. 389–394.

[12] L. Xiong and L. Liu, "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[13] "Eos," [Online]. Available: https://coinmarketcap.com/zh/currencies/eos/.

[14] "Ht," [Online]. Available: https://coinmarketcap.com/zh/currencies/huobi-token/.

[15] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.

[16] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, vol. 310, 2016.