ISCP: An Improved Blockchain Consensus Protocol

Zhong-Cheng Li, Jian-Hua Huang, Da-Qi Gao, Ya-Hui Jiang and Li Fan
(Corresponding author: Jian-Hua Huang)

School of Information Science and Engineering, East China University of Science and Technology
No. 130, Meilong Road, Xuhui District, ShangHai, China
(Email: y30160704@mail.ecust.edu.cn)
(Received Dec. 21, 2017; Revised and Accepted June 15, 2018; First Online Dec. 10, 2018)

Abstract

SCP is a recent effort that focuses on improving the scalability of blockchains by combining PoW and BFT. However, there exist security problems and performance limitation in SCP. In this paper, an improved blockchain consensus protocol ISCP with higher security and better efficiency is presented. We adopt a decentralized multi-partition consensus model to address the security problem in SCP while keeping the computational-scalable feature of the protocol. We also propose a novel intra-committee consensus algorithm which is more efficient than BFT in intra-committee consensus. We analyze and prove that ISCP has higher security and better efficiency than SCP. Experimental results show that the novel intra-committee consensus protocol can significantly reduce consensus delay and greatly increase throughput of the network.

Keywords: Blockchain; Consensus Protocol; Security; Throughput

1 Introduction

Blockchains that can provide trusted, auditable computing in a decentralized network of peers are the underlying technology of cryptocurrency platforms represented by Bitcoin [7]. They also show broad application prospects in fields such as finance, logistics, healthcare [8], and e-commerce [15]. A blockchain is a kind of state machine based on peer-to-peer networks. Ideally, the state of every peer should keep consistent. A Proof-of-Work(PoW) [9] consensus protocol based on CPU power is utilized in the Bitcoin network to achieve consistency among peers by selecting one of participants called miners to issue a proposal that everyone adopts. The miners collect transactions and compete to solve cryptographic puzzles. This process is also known as mining [13]. The PoW consensus can ensure communication efficiency and security of blockchains. However, there are some limitations to the consensus mechanism, such as consuming too much computing power and spending too long time in each epoch. The Bitcoin blockchain grows steadily at a rate of one block every 10 minutes, with size of 1MB per block [12]. The fixed growth speed and block size lead to poor throughput of only 7 transactions per second (Tx/s). Worse still, it will bring about more forks in the blockchain if we simply increase the block size and speed up the generation of blocks, which is likely to result in double-spending [4, 10].

To address the security problem of PoW mechanism under high-speed generating of blocks, Ethereum [2] adopts GHOST (Greedy Heaviest-Observed Sub-Tree) protocol which uses a new policy for selecting the main chain in the block tree to relieve the conflict between security and performance. However, the performance of the performance of GHOST-PoW has not been sufficiently tested. Eyal *et al.* propose the Bitcoin-NG [6] protocol to increase the throughput of blockchains via a primary node appending micro-blocks to the blockchain without proof of work. However, the election of the primary node is based on PoW mechanism which may lead to forks, and the eventual consistency cannot be ensured through this way. Traditional BFT consensus protocols have good performance in throughput, but they require the identities of nodes to be fixed. Furthermore, the communication complexity of BFT will increase dramatically with the increase of participants, so it only works on networks with fewer nodes. Tendermint [1] with low communication overhead is a variant of BFT protocols. It offers better node scalability and security than BFTs.

In recent years, hybrid PoW/BFT consensus protocols become the promising solution for high performance blockchains. SCP [14] is a computationally scalable Byzantine consensus protocol for blockchains. It utilizes a PoW-based identity management mechanism to prevent Sybil attacks [5] and divide nodes into different committees. Moreover, committees generating blocks in parallel through the BFT protocol enables the network throughput to scale approximately linearly with computing power. However, the node scalability of the BFT protocol is poor, and communication complexity increases

dramatically with the increase of nodes within committees, which will lead to long consensus delay. Besides, a final committee is designated to combine the blocks of sub-committees into an ordered blockchain data structure, which may cause the security problem. There exist inherent contradictions between communication complexity and security of the final committee. It is difficult to ensure the security of the protocol while keeping its efficiency. This paper presents an improved SCP protocol (ISCP). We design a decentralized multi-partition consensus model without the final committee to address the security problem in SCP and reduce the communication complexity of the protocol. We further propose a more efficient intra-committee consensus mechanism that simplifies the consensus process and reduces the consensus delay.

The remainder of this paper is organized as follows. In Section 2, we overview some novel blockchain consensus mechanisms that scale PoW and BFT protocols. Section 3 analyses security and efficiency problem of SCP. Section 4 introduces our improved consensus protocol in detail. In Section 5 and Section 6, we analyze the security and efficiency of ISCP thoroughly. Experimental results are presented in Section 7. The contributions of this paper are concluded in Section 8.

2 Related Work

PoW blockchains are not suitable for modern cryptocurrency platforms due to their poor performance. Therefore, many approaches have been proposed to solve the problem. The GHOST protocol used in Ethereum theoretically supports higher throughput than Bitcoin. It adopts a new policy that weights the subtrees rooted in blocks rather than the longest chain rooted in given blocks called the longest chain rule in Bitcoin. The new policy relieves the conflict between performance and security, so it supports higher throughput. However, the performance of GHOST has not been verified yet because the current throughput of Ethereum is only about 0.2 Tx/s on average. Eyal *et al.* proposed Bitcoin-NG that increases network throughput and reduces consensus delay. In Bitcoin-NG, a primary node elected by means of PoW appends multiple micro-blocks that consist of transactions to the blockchain without PoW mining. However, forks will appear during the election of the primary node inevitably and the eventual consistency cannot be guaranteed, which may lead to security problems.

Traditional BFT protocols, which support high throughput, are only applicable in networks with few nodes because they are bandwidth-limited. Classical BFT protocols would run in $O(n^2)$ or $O(n^3)$ communication complexity. With increase of nodes, the consensus delay will eventually become unacceptable. In addition, they cannot tolerate the fluidity of participants. As a result, Byzantine agreement protocols cannot be directly used in blockchain consensus. As a variant of BFT, the Tendermint protocol has higher security, better flexibility than traditional BFTs because participants are forced to lock their coins in a bond deposit during the consensus process and a block is added to the blockchain only if it has been signed by more than 2/3 validators. HoneyBadger [11] is a randomized BFT protocol which supports more nodes than classical BFT protocols and ensures good practical performance. Liu *et al.* [3] argue that the attack model assumed by the BFT systems rarely appears in reality and propose the XFT protocol which reduces the communication complexity and tolerates up to n/2 byzantine nodes simultaneously.

The lightning network proposed by Poon *et al.* [16] increases the transaction throughput through a dedicated fast channel. Through the scalable micro-payment channel network, parties can make high-frequency and bidirectional micro-payment with extremely low delay. However, the security of the lightning network is difficult to guarantee and it essentially belongs to offline blockchain technology. Micro-payment channels [17] increase the throughput of blockchains, but it is also offline blockchain technology and its security is difficult to guarantee.

Hybrid consensus refers to a new kind of consensus mechanism which combines PoW and BFT. SCP is a hybrid consensus protocol using PoW for identity management and BFT for consensus. Generating blocks in parallel enables the throughput of blockchains to scale approximately linearly with the number of participants in SCP.

3 SCP and Its Two-layer Blockchain

SCP utilizes proof-of-work to randomly place nodes into different committees, and these committees propose sub-blocks in parallel, thus improving the throughput of the blockchain network. The problem here is how to combine the outputs of committees into an ordered data structure which will be added to the blockchain. In SCP, a final committee is designated to combine these sub-blocks like the centralized institution. Furthermore, SCP has proved the following lemmas:

Lemma 1. *In every epoch with good randomness, for each committee, at least* $c/2+1$ *committee members will be honest with probability at least* $1 - e^{-27c/160}$. *Moreover, the probability of generating* $c/2 + 1$ *malicious identities by the end of the epoch is also exponentially small.*

Lemma 2. *In every epoch with good randomness, the honest members agree on a unique value with at least* $c/2+1$ *signatures, with probability at least* $1 - e^{-27c/160}$.

Lemma 3. *In every epoch with good randomness, honest members of the final committee will broadcast a combined value (from values from other committees) which has at least* $c/2 + 1$ *signatures, with probability at least* $1 - e^{-27c/160}$.

Where c is the size of each committee, 2^s is the number of committees. Lemma 3 ensures security and correctness of the final committee as long as c is large enough. However, the parameter c has significant influence on efficiency of SCP because the total number of message transmissions is $O(nc + c^3)$ in each epoch where n is the total number of nodes in an epoch. As shown in Figure 1, assuming n is 10,000 (10,510 nodes in Bitcoin and 15,147 nodes in Ethereum until May 2018), the number of message transmissions will reach 390,000 when the size of committee is 35. Moreover, according to Lemma 3, probability that the final committee behaves correctly will decrease dramatically when the size of final committee is below 50, which is shown in Figure 2. Therefore, the correctness of final committee cannot be ensured with overwhelming probability while keeping the efficiency of the protocol.

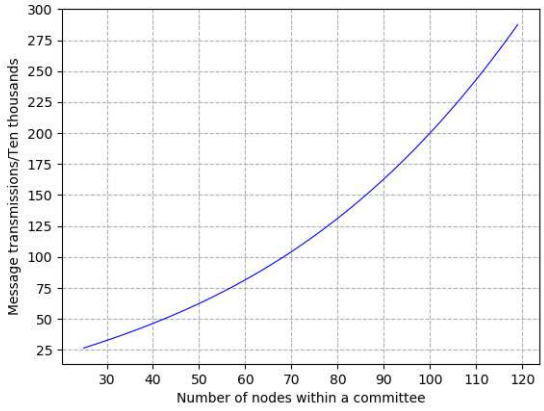

Figure 1: Message transmissions grow polynomial with the size of committee

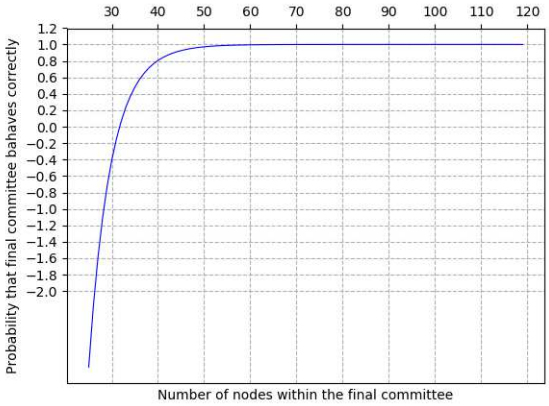

Figure 2: Probability that the final committee behaves correctly decreases dramatically when the size of final committee is below 50

4 ISCP

To address the security and efficiency problems of SCP, we propose ISCP, an improved blockchain protocol. We design a decentralized multi-partition consensus model which consists of only single layer without the final committee. We further propose an inter-committee consensus protocol to ensure all the honest nodes reach an agreement securely and efficiently on the final block which is then added to the blockchain. To further improve the efficiency of ISCP, we also adopt a novel intra-committee consensus algorithm which only requires linear communication complexity.

4.1 Decentralized Multi-partition Consensus Model

As shown in Figure 3, we propose a decentralized multi-partition consensus model. The operation of ISCP is divided into epochs. In each epoch, ISCP splits network participants into several sub-committees to generate sub-blocks in parallel. Similar with SCP, nodes in ISCP are random assigned into different committees according to the PoW computation result. The last r bits of the PoW result is used to specify which committee a node belongs to, *i.e.*, each committee is identified by its r-bit committee id. Unlike SCP, a final committee is not required to integrate sub-blocks in our decentralized multi-partition consensus model. The committees in our protocol are responsible for not only generating sub-blocks but also combining all the correct sub-blocks into a final consensus block.

The consensus process of each epoch is divided into two steps. In the first step, committees run our intra-committee consensus protocol to process separate sets of transactions and generate sub-blocks in parallel. Once a sub-block is verified and signed by at least $c/2 + 1$ members of a committee, the sub-block will be broadcast to all the sub-committees instead of sending to the final committee. In the second step, each committee runs an inter-committee consensus protocol to reach an agreement on the final consensus block that includes all the correct sub-blocks. The final consensus block will be added to the blockchain. At the same time, a random string is revealed to each node to start a new epoch.

4.2 Intra-committee Consensus

4.2.1 Algorithm

The PoW consensus algorithm is designed for Bitcoin networks with a large number of nodes and high mobility, but it is criticized for its heavy computational resource consumption and unstable consensus period. SCP adopts the BFT protocol for consensus in committees. The BFTs protocol, which are bandwith-limited, are not suitable for intra-committee consensus in our system because the number of nodes may exceed 200 (*e.g.* 300) in a single committee. In this paper, we propose a novel intra-

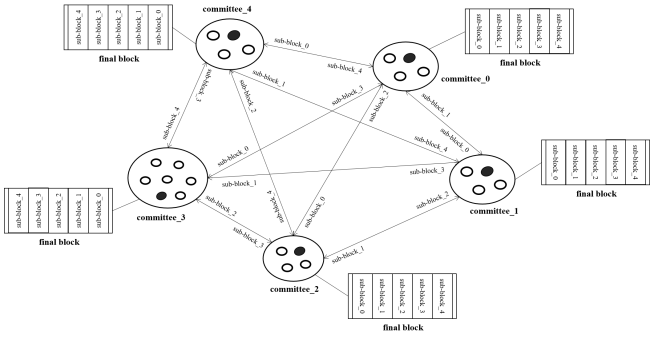

Figure 3: Decentralized multi-partition consensus model

committee consensus algorithm to achieve better consensus performance. We adopt a "retry-on-failure" mechanism to achieve consensus and reduce communication complexity in each committee. The interaction process of the novel intra-committee consensus protocol includes five operation steps. These steps are described as follows:

In Step 1, the leader node in each committee broadcasts the *prepare* message $\langle PreBlockHash, BlockHash, Block_{pre}, random, Signature, CommitteeId\rangle$ within the committee. Where $Block_{pre}$ contains all the correct transactions received by the committee, *random* is a random value chosen by the leader node as the seed for generating *epochRandomness* which will be used in next epoch, and $CommitteeId$ is the identity of the committee;

In Step 2, nodes in the committee verify the correctness of data in $Block_{pre}$, and send a *prevote* message $\langle BlockHash, IP, PK, nonce, Signature\rangle$ to the leader node if the verification succeeds. If a node detects errors in $Block_{pre}$, the node will ask other nodes in the committee to re-elect the leader node.

In Step 3, when the leader node in a partition collects $c/2 + 1$ of the *prevote* messages the for the $Block_{pre}$, it broadcasts a *submit* message $\langle Block, BlockHash, Timestamp, random, PK, Signatures, CommitteeId\rangle$ to the nodes in the committee.

In Step 4, the nodes in the committee verify whether the *Signatures* in the *submit* message contains at least $c/2 + 1$ valid signatures or not. If the verification fails, another leader node will be randomly elected to restart the consensus process. In this paper, a new concept called computation power distance between nodes is introduced to help to randomly elect a new leader node when the current leader node is compromised. We will introduce it later in Section 4.2.2. If the verification succeed, nodes in the committee will broadcast the *submit* message to other committees.

4.2.2 Computation Distance

When the leader node in a committee is compromised, honest nodes will randomly select a new leader node which has the minimal computation distance with the previous leader node to continue the consensus process. In ISCP, we introduce a new concept called computation distance to ensure the randomness of selection. The computation power distance is defined as follows:

$$Dist(node_1, node_2) = Hash_{node_1}\ XOR\ Hash_{node_2}. \quad (1)$$

Where $Dist(node_1, node_2)$ is the computation power distance between $node_1$ and $node_2$, $Hash_{node_1}$ and $Hash_{node_2}$ are the suitable fixed-length hash strings calculated by $node_1$ and $node_2$ in the previous PoW phase. XOR stands for the logical operation whose output is true only when inputs differ. We can easily prove the randomness of the selection because the hash string is randomly generated.

4.2.3 Normal-case Operation

When the leader node in a partition is a non-malicious node, the timing diagram of the protocol is illustrated as Figure 4.

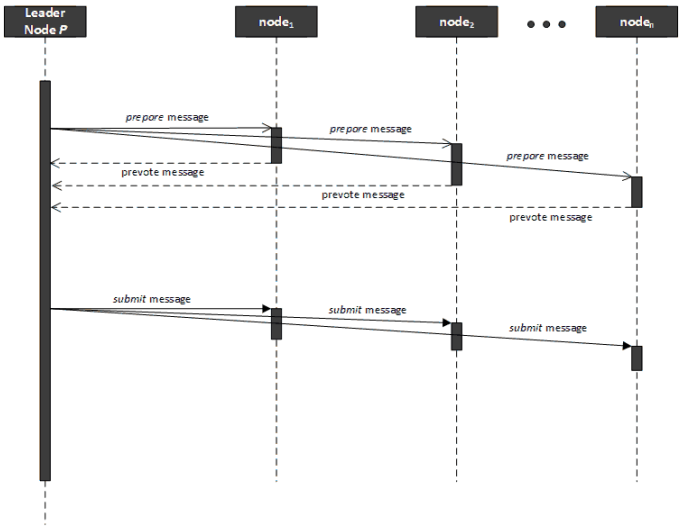

Figure 4: Normal case timing diagram of intra-committee consensus

After joining a committee, leader nodes and ordinary nodes start to collect transactions submitted by users in the blockchain network. In our system, all nodes in a committee can receive transactions and all the received transactions will be broadcasted within the partition. This process ensures that the committee can continue to process transactions even if the leader node is compromised. When the received data reaches a certain number of bytes (such as 1 MB) or the waiting time expires (for example 5 minutes), the committee generates and broadcasts a sub-block following the protocol described above.

4.3 Inter-committee Consensus

After broadcasting sub-blocks, all committees have to achieve consensus on the final block through running the

inter-committee consensus protocol. Our goal is to ensure security with overwhelming probability and achieve $O(n)$ communication complexity which is independent of the size of a committee. An inter-committee consensus protocol is introduced to integrate sub-blocks into a final consensus block. The protocol consists of the following steps:

In Step 1, after receiving a *submit* message with sub-block, an honest node checks whether the sub-block contains at least $c/2+1$ correct signatures or not. If the verification fails, the honest node will discard this message and stop to propagate to other nodes. If the verification succeeds, the honest node will save the sub-block and sends the *submit* message to its neighboring nodes.

In Step 2, once a node has received sub-blocks from all the committee, it begins to take the ordered set union of all transactions in sub-blocks into a final consensus block where sub-blocks are arranged by the order of committee id. If there exist conflicts between sub-blocks, the transaction in the sub-block behind will be deleted from the final consensus block.

In Step 3, a node which has generated the final block becomes a leader node in its committee and the committee run the intra-committee consensus protocol to reach an agreement on the final consensus block.

In Step 4, each committee broadcast its $confirm$ messages $\langle PreBlockHash,\ BlockHash,\ Timestamp,\ CommitteeId,\ epochRandomness,\ nodesList\rangle$ to other committees, where $BlockHash$ is the cryptographic digest of the final consensus block and $epochRandomness$ calculated from seeds in all the sub-blocks is the random value for next epoch of consensus process. $nodeList$ is a list of 20 to 30 members in a committee from where other nodes can download the final consensus block.

In Step 5, once a node has received at least $c/2+1$ valid $confirm$ message with the same $PreBlockHash$ and $BlockHash$, it adds the final consensus block to the blockchain locally and begins the next epoch.

5 Security Analysis

In ISCP, we consider the same threat model and security assumptions as SCP. Malicious nodes may behave arbitrarily and the portion of byzantine adversaries is no more than 1/3. In addition, honest nodes in the network topology are connected and the communication channel is synchronous.

5.1 Intra-committee Consensus Security

As mentioned above, nodes are randomly assigned into different committees, the number of compromised nodes is at most 1/3 at a high probability. We utilize a "retry on failur" method to elect an honest leader node to propose a correct sub-block and reach consensus within a committee. We also adopt a new concept called computation distance to ensure randomness of the election. In each time of election, The probability that the elected leader node behaves arbitrarily is no more than 1/3. In the first x times of elections, the probability P that all the leader nodes are compromised satisfies the following constraint:

$$P = 3^{-x}. \tag{2}$$

As shown in Figure 5, with the increase of election times, the probability that all the previous leader nodes are malicious decreases dramatically and honest nodes in a committee will reach an agreement once an honest leader node turns up. A Malicious leader node may broadcast a sub-block without enough signatures, but honest nodes will refuse to accept it and stop to propagate to other nodes.

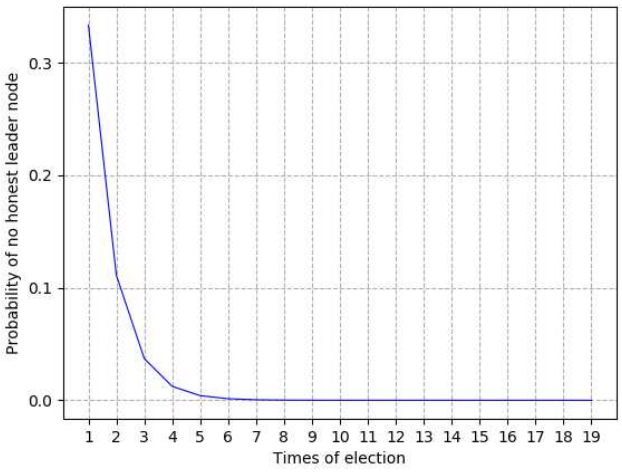

Figure 5: Probability that elected leader nodes are all malicious decreases quickly with times of election

5.2 Inter-committee Consensus Security

During propagation phase of sub-blocks, a malicious adversary may forge sub-blocks to confuse other honest nodes in the network because honest nodes do not know the identities of the nodes in other committees. These counterfeit sub-blocks consist of correct transaction data but are different from the origin ones, *i.e.*, part of transactions are deleted. We will prove that it is extremely hard for malicious adversaries to launch such an attack.

As mentioned above, $Provotes$ in the *submit* message must contain at least $c/2+1$ valid *prevote* messages. The *prevote* message contains IP, PK, $Signature$ and *nonce* of a specific node. Honest nodes can check the validation of an identity by comparing the difficulty and hash string which is calculated from IP, PK and *nonce* in *prevote* message. A malicious adversary wants to forge a sub-block, he must create enough identities to provide enough valid signatures. Moreover, these identities must belong to the same committee which is identified by an

r-bit committee id. The malicious adversary has to search for valid nonce that makes the calculated hash string have (50+r) same bits with the original. If T is the expected time for all the users, collectively, to find one proof-of-work, then the adversary has to take a time T_{byz} to find a satisfied nonce value. The T_{byz} satisfies the following constraint:

$$T_{byz} = 2^s \cdot T. \tag{3}$$

Assuming T10 minutes, 2^s is 32, T_{byz} will be 5.3 hours which are far more than the time an epoch takes. Therefore, It is nearly impossible for the adversary to launch such an attack.

6 Efficiency Analysis

6.1 Intra-committee Consensus Efficiency

In SCP, a committee runs classical consensus protocol such as PBFT to propose sub-block. The number of nodes is c in a committee, the operation of PBFT protocol in normal case is shown in Figure 6. Four phases are needed in each consensus epoch, including *pre-prepare* phase, *prepare* phase, *commit* phase and *reply* phase. During the last *reply* phase, nodes in a committee submit a sub-block to the upper layer (the final committee). The messages required for a consensus process is the sum of messages in four phases, $Msg_{sum} = 2c^2 - c$, and the time complexity is $O(c^2)$.

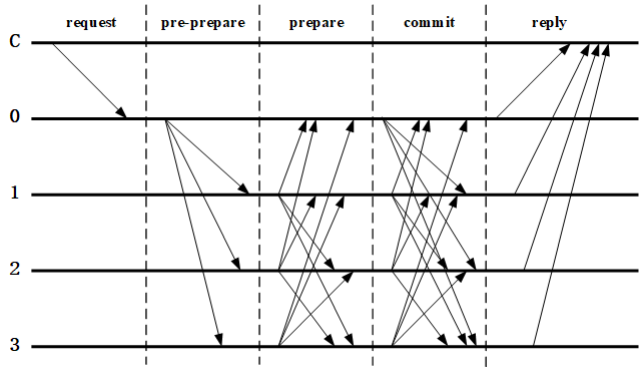

Figure 6: Normal case operation of the PBFT protocol

The operation of our intra-committee consensus protocol in normal case is showed in Figure 7. Four phases are required to complete a consensus, including *prepare* phase in which the leader node broadcasts $Block_{pre}$ to other nodes in the partition for verification, *prevote* phase during which vote messages towards $Block_{pre}$ from other nodes will be sent to the leader node, *submit* phase in which the leader node broadcast a block with enough signatures to the other nodes in the committee, *broadcast* phase in which the committee members broadcast the *submit* message to other committees. The number of messages required for a consensus is $Msg_{sum} = N + c + c + c - 3 = 3c - 3 + N$, and the time complexity is $O(c)$.

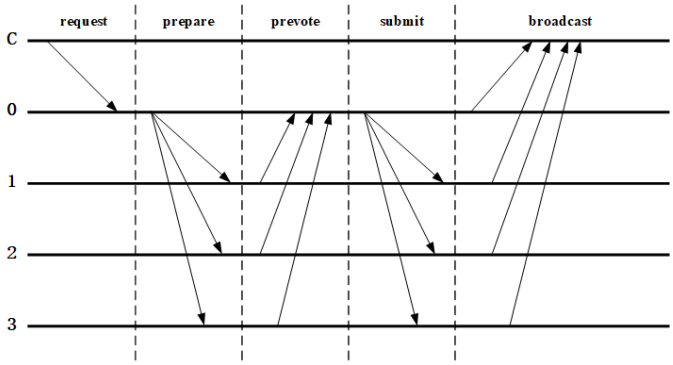

Figure 7: Normal case operation of the intra-committee consensus protocol

PBFT can achieve the state synchronization among honest nodes even if a few nodes are compromised. However, consistency among nodes within a committee is achieved through a voting process in our system. Compared with BFTs, we cancel the mutual communication among the nodes in our intra-committee consensus protocol. Even if the leader node is compromised, other nodes can detect the compromise in time and continue to complete the consensus. From the analysis above, it can be concluded that the intra-committee consensus protocol in ISCP greatly reduces the computation complexity of the protocol compared with the BFTs protocol.

6.2 Inter-committee Consensus Efficiency

In SCP, the number of messages transmitted in the final consensus phase and broadcast phase is $Msg_{sum} = N + c^3$. The final committee has to run the PBFT protocol whenever a sub-block is proposed by a node in committees, which causes very high communication complexity. In contrast, each committee only broadcasts a sub-block to the network in ISCP, which makes the communication complexity independent of the size of committee. The total number of messages transmitted during the inter-committee consensus phase is $(3 + 2^s) \cdot N$ which consists of an intra-committee consensus process and a broadcast of the final block.

7 Experimental Evaluation

Experiments are conducted to test and compare the consensus delay and throughput of ISCP and SCP in the intra-committee.

7.1 Experiment Setup

In the experiments, Docker, an advanced container virtualization technology, is used to simulate network nodes with version of Docker Community Edition 17.09.0-ce-win33 (13620). The codes are based on Python 3. The communication between nodes is based on the UDP protocol. An official Docker image with python version 3.5.4-jessie is the running environment of the codes. The host's memory is 8GB and its operating system is Windows 10 Professional Version 14393.1770. 2048MB memory is allocated to Docker for use.

7.2 Consensus Delay Test

Consensus delay experiments test the time required for SCP and ISCP to complete a consensus in one committee. SCP uses the PBFT protocol to reach a consensus, while ISCP uses the intra-committee consensus protocol reach a consensus. By continuously increasing the number of nodes in the committee, we obtain a delay trend for consensus in a partition, as showed in Figure 8. Each data is the average of 20 test results under the same conditions.

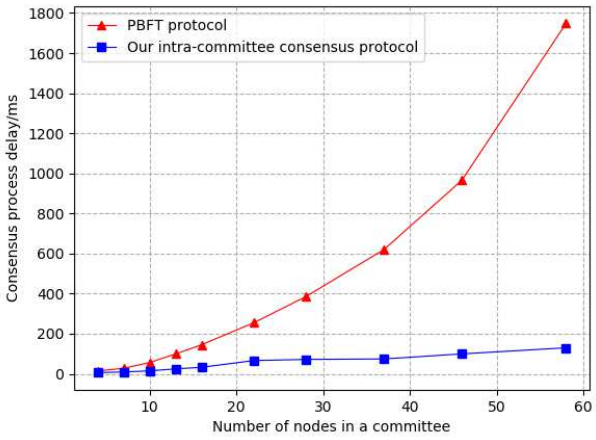

Figure 8: Consensus delay evaluation

In a committee, the consensus delay of SCP approximately grows quadratically with the increase of number of nodes, but the consensus delay of ISCP increases linearly at the same conditions. The reason is that there is many unnecessary communication between nodes in the PBFT protocol used by SCP when the leader node is honest with high probability. Massive message transmission between nodes greatly increases the consensus delay, especially in the internet environment where there may be non-negligible delay during message delivery. The intra-committee consensus protocol in our single-layer blockchain is used for electing a consensus sub-block by voting within a committee. As a result, messages exchanged between nodes are greatly reduced. Experimental results show that our intra-committee consensus protocol can greatly reduce the time it takes to reach consensus in a committee, which enables the committee process network requests more quickly and provide better services.

7.3 Throughput Test

This test compares the processing performance of SCP and ISCP in a committee. SCP uses the PBFT protocol to reach a consensus, while ISCP uses the intra-committee consensus protocol to reach a consensus. A certain number of requests (200 in the test) are send to the committee with sending rate increased constantly. When the sending rate is increased to a certain extent, requests cannot be fully processed in the committee and some messages are lost. We think this is a failure. In the experiment, the failure rate of processing requests of the two protocols is compared at different request sending rate.

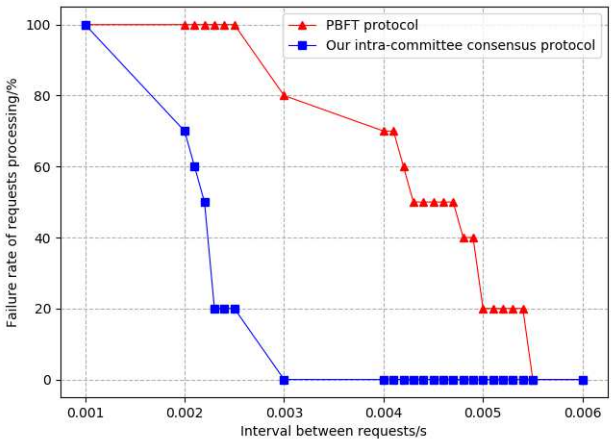

Figure 9: Throughput evaluation

As shown in Figure 9, the failure of the PBFT protocol occurs when requests are processed at a sending rate of 185 requests per second. However, the failure of ISCP occurs at 333 requests per second. At a high level, when new requests arrive at the partition, a queue of requests with limited length is allocated to cache the requests in each member of the partition. If the partition cannot process and remove the requests from the queue in time, the newly arrived requests will be discarded. That is, request processing begins to fail. When a committee in ISCP runs the intra-committee consensus protocol, the delay of consensus process is low and requests are processed quickly. Because the delay of the consensus process of PBFT in SCP are longer than the intra-committee consensus protocol in ISCP, slow request processing rate leads to low throughput. The experimental results show that the committee in ISCP can handle requests with higher sending rate when using our intra-committee consensus protocol. The throughput of the ISCP in committees is higher than that of the SCP committees.

8 Conclusion

BFT used in SCP can result in higher latency and communication complexity in the intra-committee consensus process. In addition, existence of the centralized final committee also leads to the increase of communication complexity and the security of final committee is hard to guarantee, which threatens the system security. This paper introduced ISCP, an improved blockchain consensus protocol to address these problems. We design a decentralized multi-partition consensus model without the final committee and an inter-committee consensus protocol to enable honest nodes to reach an agreement on the final consensus block with high efficiency. We further propose an intra-committee consensus protocol for committee consensus which is more efficient than the BFTs protocol in SCP. The consensus mechanism of ISCP enhanced the performance and security of blockchains. Experimental results showed that the consensus delay of the committees in ISCP is much lower than that of the committees in SCP, especially as the number of nodes increases. The intra-committee consensus protocol of ISCP supports higher processing rates of transactions than PBFT under the same conditions.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61472139 and a research Grant made to East China University of Science and Technology by Shanghai Education Commission. The authors are also grateful to the anonymous referees for their insightful and valuable comments and suggestions.

References

[1] S. Bano, A. Sonnino, M. Al-Bassam, *et al.* "Consensus in the age of blockchains," *Cryptography and Security*, 2017. (https://arxiv.org/abs/1711.03936)

[2] M. Bartoletti, S. Carta, T. Cimoli, R. Saia, "Dissecting ponzi schemes on ethereum: Identification, analysis, and impact," *Cryptography and Security*, 2017. (https://arxiv.org/abs/1703.03779)

[3] T. Crain, V. Gramoli, M. Larrea, and M. Raynal, *DBFT: Efficient Byzantine Consensus with a Weak Coordinator and its Application to Consortium Blockchains*, Technical Report 1702.03068, 2017. (https://arxiv.org/abs/1702.03068v3)

[4] A. Dmitrienko, D. Noack, M. Yung, "Secure wallet-assisted offline bitcoin payments with double-spender revocation," *ACM*, pp.520-531, 2017.

[5] X. Feng, C. Y. Li, D. X. Chen, *et al.* "A method for defensing against multi-source Sybil attacks in VANET," *Peer-to-Peer Networking and Applications*, vol. 10, no. 2, pp. 305-314, 2017.

[6] A. E. Gencer, S. Basu, I. Eyal, *et al.* "Decentralization in bitcoin and ethereum networks," *Cryptography and Security*, 2018. (https://arxiv.org/abs/1801.03998)

[7] A. Judmayer, N. Stifter, K. Krombholz, *et al.* "Blocks and chains: Introduction to bitcoin, cryptocurrencies, and their consensus mechanisms," *Synthesis Lectures on Information Security Privacy & Trust*, vol. 9, no. 1, pp. 1-123, 2017.

[8] T. T. Kuo, L. Ohnomachado, "ModelChain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks," *Computers and Society*, 2018. (https://arxiv.org/abs/1802.01746)

[9] J. Li, T. Wolf, "A one-way proof-of-work protocol to protect controllers in software-defined networks," *Symposium on Architectures for Networking & Communications Systems*, pp. 123-124, 2016.

[10] I. C. Lin, T. C. Liao, "A survey of blockchain security issues and challenges," *International Journal of Network Security*, vol. 19, no. 5, pp. 653-659, 2017.

[11] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Cryptology ePrint Archive*, 2016. (https://eprint.iacr.org/2016/199.pdf)

[12] M. Nofer, P. Gomber, O. Hinz, D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 183-187, 2017.

[13] R. Pass, E. Shi, "Fruitchains: A fair blockchain," *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pp.315-324, 2017.

[14] R. Pass, E. Shi, "Hybrid consensus: Efficient consensus in the permissionless model," *LIPIcs-Leibniz International Proceedings in Informatics*, 2017. (https://eprint.iacr.org/2016/917.pdf)

[15] A. Pazaitis, P. D. Filippi, V. Kostakis, "Blockchain and value systems in the sharing economy: The illustrative case of backfeed," *Social Science Electronic Publishing*, vol. 125, pp. 105-115, 2017.

[16] J. Poon, T. Dryja, "The bitcoin lightning network," *Draft*, 2015. (http://lightning.network/lightning-network.pdf)

[17] E. Rohrer, J. F. Laß, F. Tschorsch, "Towards a concurrent and distributed route selection for payment channel networks," *Networking and Internet Architecture*, 2017. (https://arxiv.org/abs/1708.02419)

Biography

Zhong-Cheng Li had received the B.Eng degree in computer science and technology from East China University of Science and Technology, Shanghai, China. He is currently pursuing the M.Sc. degree in computer science and technology from East China University of Science and Technology, Shanghai, China. His current research interests include blockchain technology and distributed network security.

Jian-Hua Huang had received the B.S. and M.S. degrees from East China University of Science and

Technology, Shanghai,China, and the Ph.D. degree in control theory and control engineering from East China University of Science and Technology, Shanghai,China. He has served as Associate Professor of Computer Science and Engineering at East China University of Science and Technology since 1998. His current research interests include computer networks, wireless sensor networks, information security, data mining, cloud computing, optimization and modeling. Dr. Huang has been a member of various network committees including Specialist Group of Shanghai Education and Research Network and Network Specialized Committee of Shanghai Higher Education Association. He is also the director of Development Center of Shanghai Education Network IPv6 Laboratory.

Da-Qi Gao had received the PhD degree in Industrial Automation from Zhejiang University, China, in 1996. Currently, he is a Full Professor in the Department of Computer Science at East China University of Science and Technology. He has authored or coauthored more than 100 papers. His research interests include Machine Learning, Pattern Recognition, Neural Networks and Artificial Olfactory.

Ya-Hui Jiang had received the B.Eng degree in information security from Jiangsu University, Jiangsu, China. She is currently pursuing the M.Sc. degree in computer science and technology from East China University of Science and Technology, Shanghai, China. Her current research interests include blockchain technology and secure multiparty computation.

Li Fan had received the B.Eng degree in information security from Qingdao University, Shandong, China. She is currently pursuing the M.Sc. degree in computer science and technology from East China University of Science and Technology, Shanghai, China. Her current research interests include blockchain technology and distributed network security.