https://doi.org/10.20965/jaciii.2019.p0856

Paper:

Scalable Blockchain Protocol Based on Proof of Stake and Sharding

Yuefei Gao, Shin Kawai, and Hajime Nobuhara

Department of Intelligent Interaction Technologies, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577 Japan
E-mail: kou@cmu.iit.tsukuba.ac.jp

[Received October 18, 2018; accepted March 22, 2019]

Blockchain – a distributed and public database of transactions – has become a platform for decentralized applications. Despite its increasing popularity, blockchain technology faces a scalability problem: the throughput does not scale with the increasing network size. Thus, in this paper, we propose a scalable blockchain protocol to solve the scalability problem. The proposed method was designed based on a proof of stake (PoS) consensus protocol and a sharding protocol. Instead of transactions being processed by the whole network, the sharding protocol is employed to divide unconfirmed transactions into transaction shards and to divide the network into network shards. The network shards process the transaction shards in parallel to produce middle blocks. Middle blocks are then combined into a final BLOCK in a timestamp recorded on the blockchain. Experiments were performed in a simulation network consisting of 100 Amazon EC2 instances. The latency of the proposed method was approximately 27 s and the maximum throughput achieved was 36 transactions per second for a network containing 100 nodes. The results of the experiments indicate that the throughput of the proposed protocol increases with the network size. This confirms the scalability of the proposed protocol.

Keywords: blockchain, scalability, consensus protocol, proof of stake, sharding

1. Introduction

Blockchain – the technology behind Bitcoin since 2008 – has become the core infrastructure for novel decentralized applications in recent years. Blockchain is a distributed and immutable public database that stores confirmed transactions in chronological order with high security [1]. Blockchain technology has transitioned from generation 1.0 to 3.0. Blockchain 1.0 – the first generation of the blockchain – was mainly related to cryptocurrency. Blockchain was first used as the database for cryptocurrencies such as Bitcoin. Blockchain 2.0 – the second generation of the blockchain – was related to smart contracts. In this generation, blockchain saw wider applications. Here, "contracts" refer to financial and legal contracts, e.g., bonded contracts, identification, and copyright [2]. Blockchain 3.0 – the third generation of the blockchain – is related to services beyond finance. Blockchain 3.0 provides infrastructure for applications in fields such as politics, health, and art [3]. The generation transition of blockchain technology is due to its decentralization, high security, and immutabilities. Although these advantages widen the application areas of blockchain, blockchain faces a major barrier: low scalability; i.e., the transaction processing rate does not increase with the network size.

Scalability is a primary limitation of blockchain technology [4–6]. In the case of the Bitcoin blockchain, the rate of transaction processing is 7 transactions per second (tps) [7]. In contrast, the average processing rate of payment systems such as Paypal is 115 tps [8]. In VISA's case, the processing rate can reach a peak of 56,000 tps [9]. The processing rate limitation of blockchain is influenced by the block size and block interval. Increasing the block size can improve the transaction throughput, but large blocks can incur a longer propagation time. Reducing the block interval decreases the latency, which is the time taken for a transaction to be confirmed; however, it increases the block duplication rate [4, 10]. The major objective of this study was to develop a scalable blockchain protocol.

Figure 1 presents a comparison of two major consensus protocols with our proposed protocol. **Fig. 1 (left)** shows the flow of the proof of work (PoW) protocol and **Fig. 1 (middle)** shows the flow of the proof of stake (PoS) algorithm. In a peer-to-peer network, the nodes agree on a new block, which contains unconfirmed transactions from the transaction pool, by running a PoW or PoS protocol. **Fig. 1 (right)** presents an overview of the proposed method. The main idea of the protocol is to divide unconfirmed transactions in the network into transaction shards and to divide the peer-to-peer network into multiple network shards. The transaction shards can be processed in parallel with the network shards using the PoS consensus protocol, becoming middle blocks. Finally, the middle blocks are included in a final BLOCK to be recorded on the blockchain.

To evaluate the proposed method, we performed several

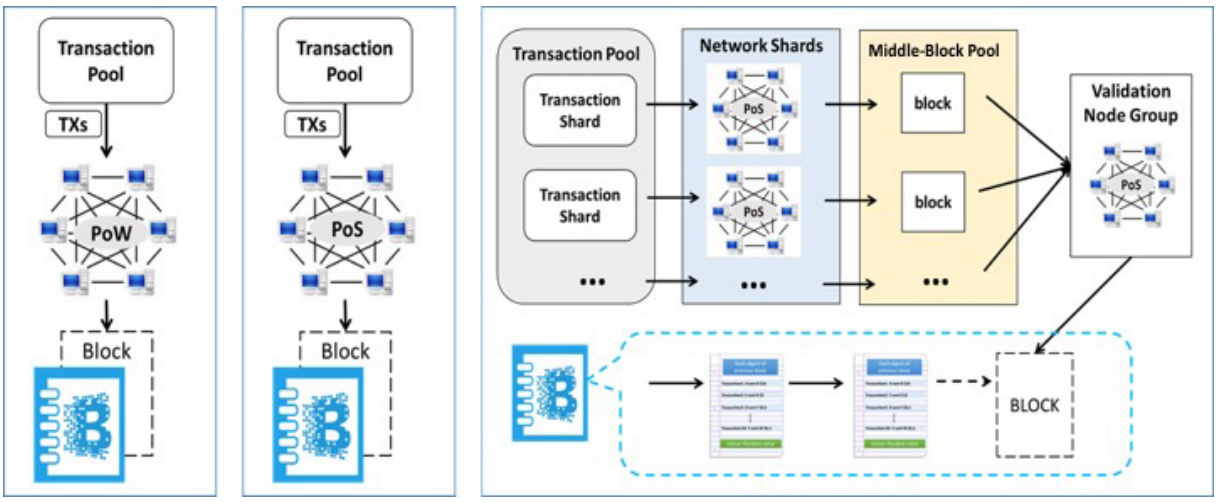

Fig. 1. Overview of existing consensus protocols: PoW (left), PoS (middle), and the proposed protocol (right).

Table 1. Comparison of the two major consensus protocols and the proposed protocol.

	Consensus Algorithm	Scalability	Latency	Throughput
Bitcoin's Protocol [10, 11]	PoW	×	600 s	3–7 tps
Peercoin's Protocol [12, 13]	Hybrid PoW/PoS	×	600 s	3 tps
Proposed Protocol	PoS+Sharding	✓	27 s	36 tps

simulation experiments. **Table 1** presents a comparison of the proposed protocol with the two major existing protocols. For the protocols of Bitcoin's and Peercoin's [13], the latency is 600 s while the throughputs are less than 10 tps each. For the proposed protocol, the latency was reduced to 27 s and the throughput reached 36 tps in a simulation network comprising 100 Amazon Web Services EC2 instances. Experiments showed that the proposed method is scalable, as the throughput scales with the size of the simulation network.

The contributions of this research are as follows:

- the proposed protocol ensures scalability without sacrificing security;
- the proposed protocol is a sharding-based PoS protocol that reduces the computations and increases the performance of the blockchain;
- the performance is compared among the existing PoW and PoS protocols and the proposed sharding-based PoS protocol in an ideal simulation network consisting of 100 nodes to confirm the scalability of the proposed protocol.

Thus, a blockchain protocol based on PoS and sharding is proposed. Although PoS and sharding have been discussed in recent years, a PoS and sharding-based blockchain has not yet been fully implemented. The state sharding strategy has been widely discussed; however, in this study, network sharding and transaction sharding were considered rather than state sharding.

This paper is divided into six sections. The first section is the introduction, which presents a brief overview of the scalability problem of blockchain. The second section introduces related works and discusses three related concepts. The third section presents a detailed discussion of the proposed protocol. The results of our simulation experiments are presented in the fourth section. In the fifth section, the scalability and complexity of the existing and the proposed protocols are evaluated according to the experimental results. Conclusions are drawn in the final section.

2. Previous Works and Related Concepts

This section introduces previous works related to the scalability problem of blockchain and explains three major concepts related to this research: PoW, PoS, and sharding. PoW and PoS are two types of consensus mechanisms used in blockchain applications. Sharding is a database technology that is employed in this proposed method.

2.1. Previous Works

In recent years, several protocols have been proposed for solving the scalability problem of blockchain. In 2016, Bitcoin-NG (Eyal et al.) and ELASTICO (Luu et al.)

were proposed. Bitcoin-NG is based on Byzantine fault tolerance (BFT), and ELASTICO is based on PoW [4, 6], PBFT (a type of BFT), and sharding. Kokoris-Kogias et al. introduced Omniledger, which is a hybrid protocol based on BFT, PoW, and sharding [14]. RapidChain, which employs BFT and sharding, was proposed by Zamani et al. in 2018 [15]. These studies solve the scalability problem of blockchain to some extent. Overall, the studies highlight the need for increasing the scalability of blockchain. However, the aforementioned methods are based on BFT or PoW, which may have limitations, such as high communication complexity or extensive calculations.

2.2. PoW [11, 16–18]

PoW is a consensus protocol used to maintain the security of cryptocurrencies. In the case of Bitcoin, miners (computational nodes) continuously solve mathematical puzzles as a competition. The fastest miner records a new block in the blockchain and receives Bitcoins as a reward. The result of the competition is determined by the central processing unit (CPU) power of the nodes. Theoretically, a higher CPU power of a node yield a higher probability of recording a block successfully and receiving the reward. Eq. (1) describes the mathematical puzzle, which is a hash puzzle. Miners calculate the hash using three parameters: the hash of the previous block, new unconfirmed transactions, and a nonce (a random number). The target is a hash with a length of 256 bits, starting with the expected number of leading zeros [19]. To solve this puzzle, miners perform repeated calculations with different nonce values until Eq. (1) is satisfied. In the blockchain of Bitcoin, the one-way hash algorithm SHA-256 is used. If an attacker intends to launch attacks on the blockchain, the attacker must perform as many computations as the rest of the nodes in the Bitcoin network. Therefore, the attack would not succeed unless the attacker has more than half of the total CPU power of the entire Bitcoin network. This is known as a 51% attack.

$$\textit{Hash (Index, Previous Block Hash, Transactions, Timestamp, Nonce)} \leq \textit{Target} \quad \ldots \ldots \quad (1)$$

The PoW protocol provides security to cryptocurrencies; however, the cost of hash calculations for producing new blocks cannot be ignored. Six hundred trillion SHA-256 hash computations are performed by the Bitcoin network per second, which results in estimated electricity and hardware costs of over 1 million dollar per day [12, 20].

2.3. PoS [12, 17]

PoS is one of the consensus protocols designed to replace PoW for reducing the energy consumption required by future cryptocurrencies. Peercoin was the first cryptocurrency implemented PoS. In PoS, as shown in Eq. (2), the generation of a new block is proportional to the concept of coin age rather than the CPU power. Coin age is the stake status and is defined as the amount of coins multiplied by the holding period. A larger the coin age yields a higher probability for the node to record a block successfully. This is because when the coin age is large, the value of the right side of Eq. (2) ($\textit{Coin age} * \textit{Target}$) is large. Therefore, there are more possible values for the hash function on the left side of Eq. (2). In contrast to PoW, PoS do not need a large amount of hash calculations. Therefore, PoS significantly more cost-effective than PoW.

$$\textit{Hash (Index, Previous Block Hash, Transactions, Timestamp)} \leq \textit{Coin age} * \textit{Target} \quad \ldots \ldots \quad (2)$$

Because the high security provided by PoW depends on a large number of calculations, concerns may be raised regarding the security of PoS. However, the probability of a 51% attack is lower in the case of PoS, for two reasons. First, performing a 51% attack in a PoS network is extremely expensive, requiring up to $50 million [21]. Second, the attack may turn out to be ineffective. Thus, this gives low incentive to attack the network.

Table 2 presents the main differences between the two consensus protocols. The PoW protocol is based on the CPU power needed to perform a large amount of computations. This consumes considerable electricity, resulting in high costs. PoS is based on the coin age rather than the computation speed; therefore, the cost is low. With regard to the 51% attack concern, PoS is less likely to be attacked than PoW.

Table 2. Comparison of PoW and PoS protocols.

	Proof of Work (PoW)	Proof of Stake (PoS)
Determiner	CPU power	Coin age
Cost	High	Low
51% attack	Potential possible	Almost impossible

2.4. Sharding [22, 23]

Sharding is a traditional technology used to partition a database. Sharding involves separating large databases into small data shards that can be managed quickly and easily. Inspired by the database sharding concept, sharding has been proposed as a technique capable of solving the scalability problem of blockchain. In the current blockchains, nodes are distributed around the world. Each node must process all transactions and store the states of all the transactions in history. This provides high security but limits scalability. In the case of Bitcoin, the processing rate is in the range of only 3–7 tps. In 2016, Luu et al. presented the ELASTICO protocol for open blockchains. This protocol employs sharding: transaction shards are simultaneously processed in small mining networks [6]. PoW is applied for creating node identities, and the consensus protocol is on a standard Byzantine agreement protocol and sharding.

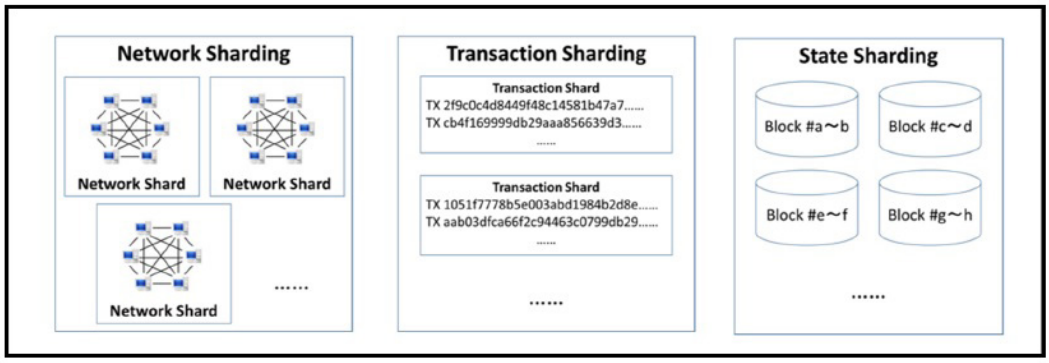

Fig. 2. Three sharding strategies.

Sharding can help blockchains become horizontally scalable, which implies that the transaction processing rate increases with the network size. There are three main sharding strategies: network sharding (**Fig. 2 (left)**), transaction sharding (**Fig. 2 (middle)**), and state sharding (**Fig. 2 (right)**). Network sharding involves dividing the blockchain network into small network shards so that transactions can be processed in parallel. In transaction sharding, unconfirmed transactions are separated from the confirmed ones and grouped into small transaction shards. State sharding is the process of storing different states (i.e., account information) in different shards to reduce the storage burden of each node. Although state sharding offers a great benefit, the need for communications across shards results in high complexity. Additionally, because the state data are stored separately, the integrity of the data can be compromised if a shard is attacked. Therefore, state sharding was not considered in the present study.

3. Proposed Protocol

In this section, we present the proposed sharding protocol for scalable blockchains. We first provide a detailed explanation of the various sharding strategies and then introduce the proposed protocol.

In network sharding, a mechanism is needed to determine how to separate nodes into shards. From the viewpoint of security, centralization caused by the monopoly of certain nodes should be prevented. The mechanisms that we use are 1) random formation and 2) reshuffling.

1) Random formation: In this mechanism, the network is divided into small randomly generated network shards.
2) Reshuffling: After a certain period, the nodes are shuffled, and new network shards are formed. Random formation and reshuffling prevent attacks by malicious nodes.

For transaction sharding, a mechanism is needed to determine how to create transaction shards for avoiding double spending. Double spending can occur if a malicious user creates two transactions from the same input to generate two different outputs in different transaction shards

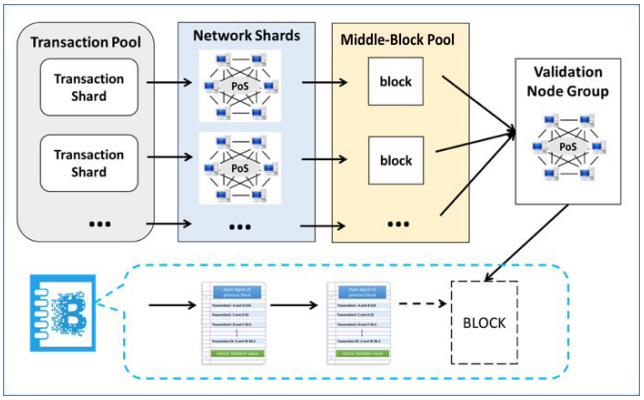

Fig. 3. Overview of the proposed method.

to be processed. There are two main ways to avoid this problem. First, each shard should communicate with every other shard [22]. However, this increases the complexity. Second, transaction shards should be determined by the address of the sender [22]. Here, the transactions with the same sender address are placed into the same shard so that double-spending transactions can be detected easily. Thus, the complexity does not increase because cross-shard messages are not needed.

3.1. Overview of Proposed Method (Fig. 3)

The proposed method employs a combination of the sharding protocol and the PoS consensus mechanism. Consider a network containing cn nodes that is divided into c groups (i.e., network shards). Each group contains n nodes. Among the c groups, $c-1$ groups are regular groups, and one group works as a validation node group. The blocks are of two types: middle and final blocks. To distinguish the two, "block" (lowercase) represents the middle blocks, and "BLOCK" (uppercase) represents the final blocks. Middle blocks are generated by the $c-1$ groups, and then processed and combined by the validation group to produce the final BLOCKs, which are recorded on the blockchain.

The middle blocks and final BLOCKs are produced in epochs. Each epoch comprises the following four steps:

Step 1: Form node groups
Each node works in a group. A leader node is selected when a node group is formed. The leader node collects the identities of the other nodes in the group and creates an identity list. This list is broadcast to the other group leaders. This process reduces the node communication complexity from $O(n^2)$ to $O(cn)$.

Step 2: Create middle blocks
Internal group consensus is conducted in each node group to produce middle blocks. The PoS consensus protocol is applied to determine the coin age for each node. Nodes with a larger coin age (i.e., coin amount multiplied by holding period) have a higher probability of generating a new middle block.

Step 3: Generate final BLOCK
The middle blocks created by the general node groups are collected, verified, and combined by the final validation group via a PoS consensus protocol. The final BLOCK is produced and broadcast to the whole blockchain network.

Step 4: Reshuffle nodes
This step is executed every t epochs. All of the nodes are reshuffled to form new node groups.

3.2. Forming Node Groups

Initially, all the nodes in the network form node groups. Assume that there are n nodes in a group. All the nodes in the same node group know the identities of each other. Each node broadcasts its identity to all the other nodes in a simple manner. However, this yields a high message complexity of $O(n^2)$. In this study, we used a new strategy to reduce the complexity. The strategy is presented in Section 5. We assume that there are c node groups in total, and one group is randomly selected as the final validation group.

3.3. Create Middle Blocks

When the node-group formation is completed, transaction shards are assigned to the $c-1$ regular node groups. When transactions are separated into shards, those with the same sender address are placed in the same shard. This mechanism for transaction sharding has two advantages: 1) it prevents double spending and 2) it helps avoid cross-shard communication. Transaction shards are produced by regular node groups to create middle blocks, which are processed by the final validation group. A middle block contains an index, the previous BLOCK hash, a transaction shard and a timestamp.

3.4. Generating Final BLOCKs

The middle blocks produced by regular node groups are collected by the final validation node group to generate the final BLOCK. A PoS consensus protocol is employed to select a node whose final BLOCK is recorded in the blockchain. The final BLOCK contains an index of the previous BLOCK hash, middle transactions and a timestamp.

3.5. Reshuffling Nodes

Every t epochs, nodes are reshuffled and new groups are formed. Node reshuffling helps to prevent malicious nodes from taking control of a network shard, reducing the risk of centralization. Thus, reshuffling secures the network.

To evaluate the scalability of the proposed method, we conducted simulation experiments under different conditions.

4. Implementation and Evaluation

We implemented the proposed protocol and conducted experiments to evaluate its scalability in comparison with that of the existing protocols for the same simulation network. The two objectives of the experiments were as follows: 1) to confirm that the performance of the proposed protocol is in agreement with the theory and 2) to compare the proposed protocol with the existing PoW and PoS consensus protocols.

4.1. Experiment Setup

We implemented the proposed protocol using Node.js and conducted simulation experiments on Amazon EC2 to measure the performance. The size of the simulation network ranged from 20 to 100 t2.micro Amazon EC2 instances. Each instance performed as a single node with 1 vCPU and 1.0 GB of memory. Therefore, when the number of nodes increased, the computation power of the network increased. The number of nodes n in a group varied among 5, 10, and 20, and the network size cn varied from 20 to 100. We conducted a total of 15 experiments with different settings to measure the scalability of the proposed protocol. To compare the proposed protocol with the PoW and PoS protocols, we also conducted experiments using the two existing protocols in the same simulation network with up to 100 nodes.

4.2. Evaluation Experiment

The experiments comprised two parts. The first part involved evaluating the performance of the proposed protocol with regard to the throughput and latency under different conditions and comparing it with that of the two existing protocols. We fixed the size of a node group as $n = 5$, 10, and 20 and performed five experiments for each node group size for network sizes of $cn = 20$, 40, 60, 80, and 100. Thus, a total of 15 experiments were performed. The second part involved evaluating the proposed protocol with regard to not only the throughput and latency but also the network shard size, data size, and detailed latency.

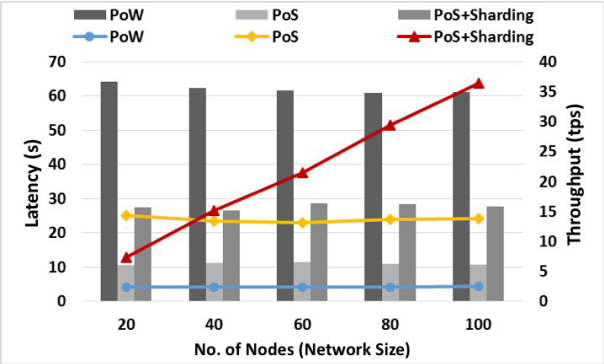

Fig. 4. Throughput and latency comparison among PoW, PoS, and the proposed method (shard size = 5).

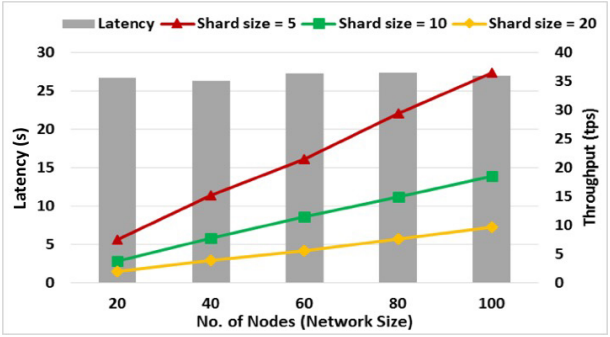

Fig. 5. Impact of shard size on throughput and latency.

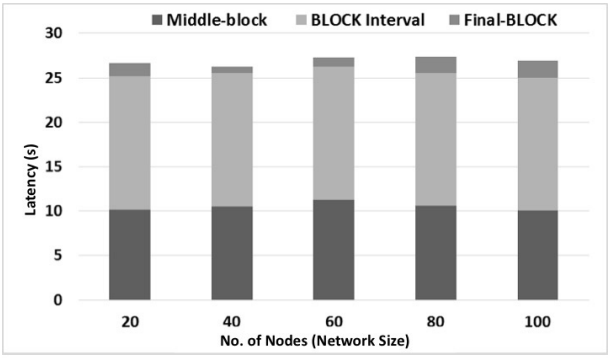

Fig. 6. Latency of the final blocks in the proposed protocol.

First, for each node group size, we performed a simulation network of 20 nodes and increased the network size four times, up to 100 nodes. We measured the latency for 10 BLOCKs and calculated the throughput. Second, we conducted experiments using the same simulation network for a total number of nodes ranging from 20 to 100 for the PoW and PoS protocols. We measured the latency for 10 PoW blocks and 10 PoS blocks.

Figure 4 compares the throughput and latency for the existing and proposed protocols. As indicated by the data bars, the PoW protocol had the highest latency of approximately 60 s, while the PoS protocol had the lowest latency of 12 s. The latency of the proposed protocol was around 27 s. The lines show the throughputs of the three protocols. As shown in **Fig. 4**, with the increase of the network size, neither PoW nor PoS exhibited an increase in the transaction processing rate. However, the proposed method processed more transactions as the network size increased. When the network size was 100 nodes, the throughput of the proposed protocol reached 36 tps, which was higher than the throughputs of PoW (3 tps) and PoS (14 tps). Thus, the throughput of the proposed protocol scaled with the network size, confirming that that the proposed protocol is scalable.

Figure 5 shows the effects of the network shard sizes on the throughput and latency. For the network shards (i.e., node groups), we used three different sizes: 5, 10, and 20 nodes. When the total network size (number of nodes) increased from 20 to 100, the number of network shards varied. **Fig. 6** shows that the latency remained at approximately 27 s for different shard sizes, while the throughput increased with the shard size. The throughput increased faster when the shard size was 5 than in the other two conditions. **Fig. 6** presents detailed information regarding the BLOCK latency for the proposed protocol. As shown, the latency of the proposed protocol comprised three main parts: 1) the latency to create middle blocks, 2) the interval time between two BLOCKs, and 3) the consensus on a final BLOCK. The sum of these three time periods determines the BLOCK latency for the proposed protocol, which was approximately 27 s.

5. Discussion

In the section, the results presented in Section 4 are discussed, and the complexity and security of the proposed protocol are analyzed.

5.1. Results and Discussion

Figure 4 shows a comparison of the latency and throughput of the three protocols. Although the latency of the proposed method was more than double that of PoS, the proposed protocol had a higher throughput than PoS because it employed sharding techniques, allowing it to perform parallel transaction processing. In the proposed method, the simulation network and the unconfirmed transactions are sharded, and transaction shards are processed in parallel in different network shards. Therefore, the final throughput is higher than those of PoW and PoS. The distribution of the transaction processing increased the throughput and scalability of the proposed protocol. According to the results shown in **Fig. 5**, the largest throughput increase was achieved when the shard size was 5, because when the network size is fixed, a smaller shard size yields a larger number of network shards. With more network shards, more middle blocks are produced. When the block size is fixed, the size of the final BLOCK is determined by the number of network shards. Therefore, the final BLOCK contains more transactions when the network shards are small, and this contributes to higher throughput.

5.2. Complexity Analysis

Assume that there are cn nodes in the network. For each node to know the identities of the other nodes, each node broadcasts its identity to the other nodes and receives the identity information of the other nodes. This results in $O(n^2)$ message complexity. In the proposed protocol, c node groups are formed, and each group contains n nodes. To reduce the message complexity, we propose a mechanism for reducing both the intar-shard communication and cross-shard communication.

In a given node group, if each node broadcasts its identity to the other nodes, the message complexity is $O(n^2)$. To reduce the intra-shard communication, a leader node is randomly selected and the identities of all the $n-1$ other nodes are sent to the leader node. Therefore, the nodes do not need to broadcast their identity to the group or to the whole network. The identity information is shared among the leader nodes. A non-leader node can ask the leader node of its group for the identities of other nodes.

Cross-shard communication is needed when transaction shards are assigned to the c node groups. Without communication between the node groups, two transactions with the same input address may be processed in two different node groups. Cross-shard communication helps prevent the double-spending problem; however, it results in high message complexity. Our strategy for avoiding this type of cross-shard communication is to place transactions with the same input addresses in the same transaction shard. Therefore, the double-spending problem can be avoided without relying on cross-shard communication.

Cross-shard communication cannot be avoided altogether, because leader nodes must share identity information regarding all the nodes. In the proposed protocol, the message complexity for intra-group and cross-shard communication is reduced from $O(n^2)$ to $O(cn)$.

5.3. Security Analysis

The 51% attack is one of the security concerns for blockchain. In the case of the PoW consensus protocol, node(s) that control more than half of the total CPU power can launch malicious attacks successfully [7]. A 51% attack is related to cost and incentive. PoS is considered more secure in this regard. D. Larimer confirmed that it is far more expensive to perform a 51% attack in a PoS-based network than in a PoW-based network [24]. While 51% attacks in PoW-based networks require considerable costs and a large amount of hard-ware, in PoS-based networks, 51% attacks require not only high costs (control of over 50% stakes) but also a certain amount of coins and a holding period. Because of the PoS mechanism, even if a 51% attack succeeds, the attacker would benefit minimally. Thus, there is little incentive for malicious nodes to attack the network.

The proposed method employs three mechanisms to achieve high security: 1) randomness – the formation of node groups and the selection of leader nodes are random, 2) reshuffling – every t epochs, all the nodes are reshuffled to form new groups, and 3) coin-age limitation – the coin age is limited and reset to zero once coins are spent. These three mechanisms help to prevent malicious nodes from taking control of the network shards and reduce the incentive for launching a 51% attack.

6. Conclusions

We proposed a protocol for solving the scalability problem of blockchain. The proposed protocol combines PoS with sharding techniques to increase the throughput. We conducted experiments to evaluate the proposed method in comparison with the two major existing protocols: PoW and PoS. Although the latency of the proposed method (27 s) was higher than that of PoS (12 s), the proposed method achieved better throughput (36 tps) when the network size increased to 100 nodes (shard size = 5). The experiments confirmed that the throughput of the proposed protocol increases with the network size.

References:

[1] M. Swan, "What is the Blockchain?," Blockchain: Blueprint for a New Economy, O'Reilly Media, Inc., pp. x-xi, 2015.

[2] M. Swan, "Blockchain 2.0: Contracts," Blockchain: Blueprint for a New Economy, O'Reilly Media, Inc., pp. 9-10, 2015.

[3] M. Swan, "Blockchain 3.0: Justice Applications Beyond Currency, Economics, and Markets," Blockchain: Blueprint for a New Economy, O'Reilly Media, Inc., pp. 29-69, 2015.

[4] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-NG: A Scalable Blockchain Protocol," Proc. of the 13th USENIX Symp. on Networked Systems Design and Implementation (NSDI '16), pp. 45-59, 2016.

[5] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller et al., "On scaling decentralized blockchains," Int. Conf. on Financial Cryptography and Data Security, pp. 106-125, 2016.

[6] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security, pp. 17-30, 2016.

[7] Scalability, Bitcoin wiki, https://en.bitcoin.it/wiki/Scalability [accessed July 24, 2018]

[8] Welcome to the PayPal Information Center, https://web.archive.org/web/20141226073503/https://www.paypal-media.com/about/ [accessed July 24, 2018]

[9] VISA Inc. at a Glance, https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf [accessed July 24, 2018]

[10] Block intervals, Bitcoin wiki, https://en.bitcoin.it/wiki/Block_intervals [accessed July 24, 2018]

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009.

[12] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf [accessed July 24, 2018]

[13] S. Goswami, "Scalability analysis of blockchains through blockchain simulation," Thesis, Master of Science in Computer Science, University of Nevada, Las Vegas, 2017.

[14] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding," IACR Cryptology ePrint Archive 2017, 406, 2017.

[15] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling Blockchain via Full Sharding," Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security, pp. 931-948, 2018.

[16] Proof of work, Bitcoin wiki, https://en.bitcoin.it/wiki/Proof_of_work [accessed July 24, 2018]

[17] BitFury Group, "Proof of Stake versus Proof of Work," http://bitfury.com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf [accessed July 24, 2018]

[18] V. Buterin, "What Proof of Stake Is And Why It Matters," https://bitcoinmagazine.com/articles/what-proof-of-stake-is-and-why-it-matters-1377531463/ [accessed July 24, 2018]

[19] K. Vaidya, "Decoding the enigma of Bitcoin Mining – Part I: Mechanism," https://medium.com/all-things-ledger/decoding-the-enigma-of-bitcoin-mining-f8b2697bc4e2/ [accessed July 24, 2018]
[20] Proof of Stake FAQ, https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ/ [accessed July 24, 2018]
[21] V. Buterin, "A Proof of Stake Design Philosophy," https://medium.com/@VitalikButerin/a-proof-of-stake-design-philosophy-506585978d51/ [accessed July 24, 2018]
[22] Y. Jia, "Op Ed: The Many Faces of Sharding for Blockchain Scalability," https://bitcoinmagazine.com/articles/op-ed-many-faces-sharding-blockchain-scalability/ [accessed July 24, 2018]
[23] D. Rhodes, "The state of sharding: How can this technology make blockchain more scalable?," https://jaxenter.com/sharding-in-blockchain-145534.html [accessed August 10, 2018]
[24] D. Larimer, "Transactions as Proof-of-Stake," https://bravenewcoin.com/assets/Uploads/TransactionsAsProofOfStake10.pdf [accessed August 10, 2018]

Name:
Yuefei Gao

Affiliation:
Department of Intelligent Interaction Technologies, University of Tsukuba

Address:
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan

Brief Biographical History:
2009-2013 University of Science and Technology Beijing
2014- Department of Intelligent Interaction Technologies, University of Tsukuba

Main Works:
• "A Decentralized Trusted Timestamping Based on Blockchains," IEEJ J. of Industry Applications, Vol.6, No.4, pp. 252-257, 2017.

Name:
Shin Kawai

Affiliation:
Department of Intelligent Interaction Technologies, University of Tsukuba

Address:
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan

Brief Biographical History:
2017 Received the Ph.D. degree from Department of Intelligent Interaction Technologies, University of Tsukuba
2018- Affiliated with University of Tsukuba

Main Works:
• "On the Choice of a Proper Initial Condition for Derivative Controllers," Proc. of the 44th Annual Conf. of the IEEE Industrial Electronics Society (IECON 2018), pp. 2391-2396, 2018.
•"General Mapping Discrete-Time Models of a Descriptor System with an Arbitrary Initial Condition," Automatica, Vol.87, pp. 428-431, 2018.
•"Exact Temporal/Spatial Discretization of a Parabolic Partial-Differential-Equation," Proc. of The 5th Int. Conf. on Control, Mechatronics and Automation (ICCMA 2017), pp. 131-136, 2018.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Hajime Nobuhara

Affiliation:
Department of Intelligent Interaction Technologies, University of Tsukuba

Address:
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan

Brief Biographical History:
2002 Received the Ph.D. degree from Tokyo Institute of Technology
2002-2006 Affiliated with Tokyo Institute of Technology
2006- Affiliated with University of Tsukuba

Main Works:
• "The transformation method between tree and lattice for file management system," Evolving Systems, Vol.4, No.3, pp. 183-193, 2013.
• "Efficient Construction of Wireless Personal Area Networks for Disaster Control," Proc. of 5th Int. Symp. on Computational Intelligence and Industrial Applications (ISCIIA 2012), 2012.
• "Approximate Solution of Fuzzy Relational Equation and its Solvability Degree for Image Compression and Reconstruction," The 4th Asian Fuzzy Systems Symp., pp. 858-863, 2000.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
• The Institute of Electronics, Information and Communication Engineers (IEICE)