2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

The Blockchain for Domain based Static Sharding

Hyunkyung Yoo, Jongchoul Yim and Sunme Kim
Network Research Division
Electronics and Telecommunications Research Institute (ETRI)
Daejeon, Republic of Korea
hkyoo@etri.re.kr, hektor@etri.re.kr, kimsunme@etri.re.kr

***Abstract*— Blockchain is the shared distributed ledger to record the history of transactions. In blockchain network, nodes validate the transactions and reach the consensus on the ordered transactions. The more transactions are happened, the more processing powers of nodes are needed. Shard is introduced to blockchain for processing the multiple transactions in parallel. In this paper, we propose the blockchain for a domain based static sharding. We split a blockchain into multiple shards based on the domain. In each shard, nodes validate the multiple transactions concurrently and keep the shard's ledger separately. With this, it can be processing of low-latency transaction. Also, by changing the composition of committee members that validate blocks dynamically, we can enable the blockchain to be more trust.**

***Keywords*— blockchain, ledger, sharding, consensus, domain**

I. INTRODUCTION

Blockchain technologies are gaining massive momentum in the last few years, largely due to the success of bitcoin [1]. Blockchain is the shared distributed ledger to record the history of transactions. Nodes in blockchain network validate transactions, generate blocks, and store them to the ledger. By establishing consensus between the distributed nodes, they can keep the same blockchain.

In IoT(Internet of Things) environment, each transaction must be immediately processed, and massive transactions occur simultaneously [2-3]. The bigger the blockchain, the fewer devices will have the capacity to store and audit the full blockchain, leading to the network becoming more centralized [4]. So, each node becomes the lack of storage, and the delay of transactions consensus becomes to happen.

Sharding in distributed database means partitioning a database table so that the data can be evenly distributed between all nodes in the cluster [5-6]. Shard is introduced to the blockchain for processing the multiple transactions in parallel. It is to partition in network into smaller shards, each of which processes a disjoint set of transactions [7-8].

In order to fulfill the distributed processing of transactions, and reduce the consensus delay, we adopt sharding in our blockchain framework. In each shard, nodes reach to agree the ordered transactions concurrently and maintain the each shard's own ledger.

In our proposal, we split a blockchain into multiple shards based on domain. That is, if a transaction is happened in Korea shard, it is proper to run consensus of transaction in that shard, because it achieves low-latency processing and enhances throughput. If the transaction which is needed to interwork between the specific shards, it is reasonable to process in special shard.

In this paper, we propose the blockchain framework for a domain based static sharding. And we present the domain based static sharding mechanism for the scalability enhancement.

There is a committee in each shard. The membership of committee is decided by proof-of-work and changed dynamically. The committee validates the transaction by PBFT. So, we can enable the blockchain to be more trust and efficiency.

This paper is organized as follows. Section II presents the architecture of blockchain framework for the parallel processing of transactions. In section III, we present the concept and transaction processing of domain based static sharding. And we explain the benefits of sharding. Finally, section IV summarizes and concludes the paper.

II. ARCHITECTURE OF BLOCKCHAIN FRAMEWORK

We consider the permissioned blockchain framework for a domain based static sharding. Our framework supports smart contracts for deployment and query of the transactions such as in Ethereum [9] and Hyperledger Fabric [10].

The functions of blockchain framework consists of validation, block agreement, p2p protocol, smart contract, membership manager. Our blockchain framework provides the Open APIs for clients to request transactions and response result.

Fig.1 shows the architecture of our blockchain framework. Validation function validates the transactions and the blocks. It has the transaction validation module and the block validation module. Transaction validation module checks if a transaction is valid and it has the correct format. Block validation module checks if the received block is valid and it has the correct format and the sequence.

Block agreement function generates the blocks by consensus among distributed nodes and updates the ledger. It has the block consensus module and the block generation module. Block consensus module provides the block agreement with the base of the transaction validation among nodes.

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00252

Block generation module performs to generate block and to update the ledger with new block.

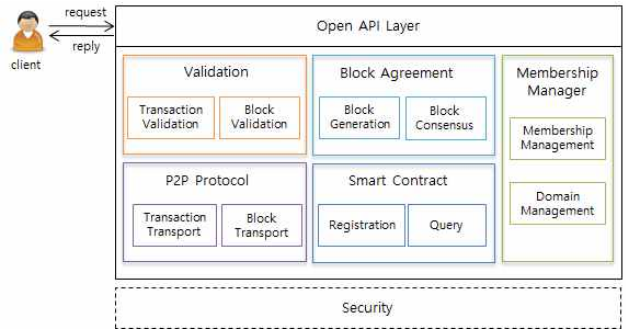

Figure 1. The Architecture of Blockchain Framework

P2p protocol function transmits the transactions and the blocks based on p2p topology. It has the transaction transport module and the block transport module. Transaction transport module broadcasts the transaction for submission, agreement, and commit. Block transport module broadcasts the agreed block, that is, the ordered transactions.

Smart contract function provides to deploy and to query of the transactions. And if the events which are described in the logic of smart contract happen, the related transactions are executed. It has the registration module and the query module. Registration module deploys and registers the smart contract logic. Query module provides to query the smart contract and to update the status of smart contract.

Membership manager function provides to manage the membership and the domain information. It has the membership management module and the domain management module. Membership management module registers the role of node like a local or a global node. It separates the node's processing authority of transactions according to the shards. Domain management module manages the domain based shard and the accounts information of domain.

When a client requests a transaction, the transaction is propagated to the nodes in network. After the selected nodes validate transactions, they propagate them to the network. And they reply the result of transaction processing to the client.

III. DOMAIN BASED STATIC SHARDING

The proposed concept of a domain based static sharding is shown in Fig.2. There are a lot of nodes in blockchain network. Shards are classified regionally such as a local or a global shard. Each node belongs to a specific shard, and it can be a local or a global node. If a transaction can process in one shard, it is a local transaction. If a transaction must process among two shards, it is a global transaction. Local nodes validate the local transactions happened in own local shard. Global nodes of global shard validate the global transactions.

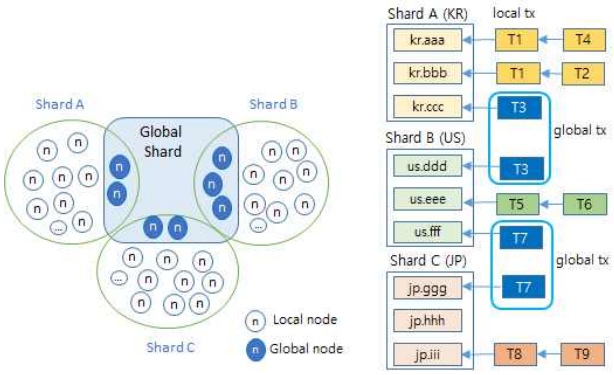

Figure 2. Concept of Domain based Sharding

Each shard manages each domain, such as shard A of KR domain, shard B of US domain, and shard C of JP domain. Accounts of shard are identified with the domain information, and have the prefix of domain such as kr.aaa.

In Fig.2, local transactions are T1, T2, T4, T5, T6, T8, T9 and global transactions are T3 and T7. T3 is the transaction from shard A to shard B, and T7 is the transaction from shard B to shard C.

There is a committee in each shard. The membership of committee is decided by proof-of-work and changed dynamically. The committee members validate the transactions by PBFT (Practical Byzantine Fault Tolerance).

To get a committee membership, nodes of a shard generate blocks through proof-of-work (PoW block) competitively. Among them, the top N nodes are selected to members of a committee, with their votes.

If a committee is composed, the first leader of committee generates the epoch flag block that means the start of the new committee's epoch. By referencing the epoch flag block, other nodes can generate PoW block to qualify the committee members.

Until composing the next committee with next epoch flag block, the current committee members validate transactions based on PBFT. After validation, they generate blocks and save a local ledger. If the composition of committee with next epoch flag block, new epoch starts and new committee validates transactions.

Local and global transactions have a different block generation process. After local transactions are validated in the local committee of a local shard, they are saved in local ledger.

Fig.3 shows the processing of local transactions. We explain the processing procedures of local transaction in case of the local transactions of shard A. Committee A-1 is composed of the selected nodes by proof-of-work among local nodes of shard A. For example, committee A-1 can be composed of n1, n2, n5, n6.

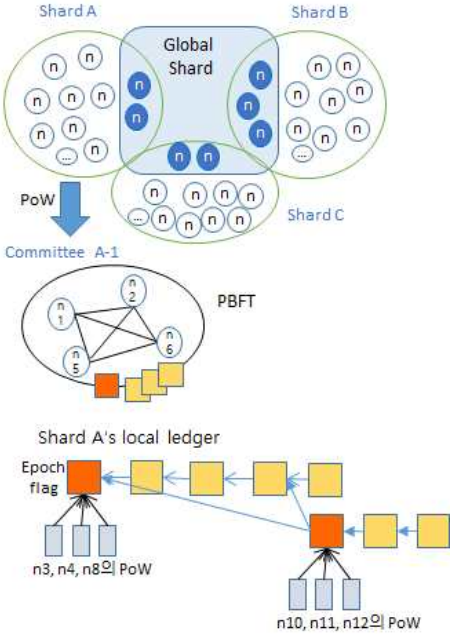

Figure 3. Processing of Local Transactions

The new epoch is started. Committee A-1 members validate transactions and reach to consensus by PBFT. And then, they generate blocks and save it to their local ledger. The nodes (n3, n4, n8), not committee members, generate the PoW blocks and are ready to be the next committee members.

The leader of committee is determined by round-robin. The first leader of committee generates the epoch flag block, and others generate transaction blocks. The generated blocks are propagated to other nodes in own shard. And they are saved in a local ledger.

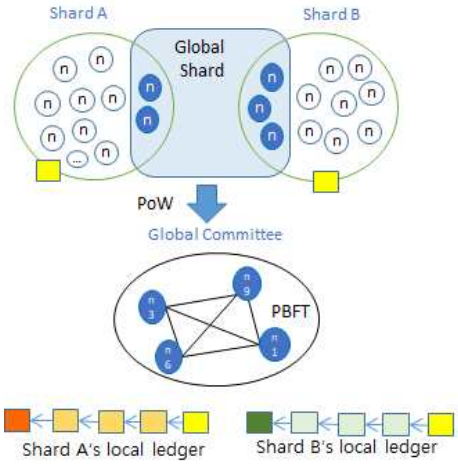

Figure 4. Processing of Global Transactions

After global transactions are validated in the global committee of a global shard, they are saved in relevant local ledgers. The processing of global transaction is shown in Fig.4.

We explain the processing of global transaction in case of the global transaction from shard A to shard B. Global committee is composed of the selected nodes by proof-of-work among global nodes of global shard. For example, global committee can be composed of n1, n3, n6, n9.

The new epoch of global shard is started. Global committee members validate transactions and reach to consensus by PBFT. And then, they transmit the validation result to nodes of shard A and B. Nodes of shard A and B generate blocks and save in each shard's local ledger.

In multiple shards, multiple committee validate and generate blocks concurrently. It reduce the processing time of blocks and the composition time of committee. Also, by changing the composition of committee members that validate blocks dynamically, we can enable the blockchain to be more trust.

IV. CONCLUSION

This paper proposes the blockchain framework architecture and mechanism for a domain based static sharding. We partition a blockchain into multiple shards based on the domain. In each shard, nodes validate the multiple transactions concurrently and keep the shard's ledger separately. With this, it can be processing of low-latency transaction

The membership of committee in each shard is decided by proof-of-work and changed dynamically. The committee validates the transaction by PBFT. So, we can enable the blockchain to be more trust and efficiency.

For further study, we have a plan to develop our blockchain framework and evaluate the experiments on an emulated network. Also we will apply our blockchain framework to IoT domain.

ACKNOWLEDGMENT

This work was supported by the ICT R&D program of MSICT/IITP. [2017-0-00045, Hyper-connected Intelligent Infrastructure Technology Development]

REFERENCES

[1] Dinh, Tien Tuan Anh, et al. "BLOCKBENCH: A Framework for Analyzing Private Blockchains," Proceedings of the 2017 ACM International Conference on Management of Data. ACM, 2017.

[2] Min, Xinping, et al. "A Permissioned Blockchain Framework for Supporting Instant Transaction and Dynamic Block Size," IEEE Trustcom/BigDataSE/ISPA, 2016.

[3] Dorri, Ali, et al. "Blockchain for IoT security and privacy: The case study of a smart home," IEEE Pervasive Computing and Communications Workshops (PerCom Workshops), 2017.

[4] Bano, Shehar, Mustafa, Al-Bassam, and George, Danezis. "The Road to Scalable Blockchain Designs," USENIX; login: magazine, 2017.

[5] Shard, https://en.wikipedia.org/wiki/Shard_(database_architecture)

[6] McConaghy, Trent, et al. "BigchainDB: A Scalable Blockchain Database (DRAFT)," 2016.

[7] Luu, Loi, et al. "A secure sharding protocol for open blockchains," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016.

[8] Gencer, Adem Efe, Robbert van Renesse, and Emin Gün Sirer, "Short Paper: Service-Oriented Sharding for Blockchains," Financial Cryptography and Data Security, 2017.

[9] Ethereum, https://github.com/ethereum/wiki/wiki

[10] Hyperledger Fabric 1.0, http://hyperledger-fabric.readthedocs.io/en/release/fabric_model.html